aws

Guida per l'utente

AWS Identity and Access Management



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Identity and Access Management: Guida per l’utente

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in qualsiasi modo che possa causare confusione tra i clienti o in qualsiasi modo che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Che cos'è IAM? 1
Perché utilizzare IAM? 3
Accesso condiviso al tuo Account AWS 3
Autorizzazioni granulari 3
Accesso sicuro a AWS risorse per le applicazioni eseguite su Amazon EC2 3
Multi-factor autenticazione (MFA) 4
Federazione delle identità 4
Informazioni d'identità per la sicurezza 4
Conformità PCI DSS 4
Quando si usa IAM? 4
Quando si eseguono diverse funzioni lavorative 5
Quando sei autorizzato ad accedere AWS risorse 5
Quando accedi come utente IAM 6
Quando assumi un ruolo IAM 6
Quando crei policy e autorizzazioni 6
Come viene gestito IAM? 7
Usa il Console di gestione AWS 8
AWS Strumenti da riga di comando 10
Usa la AWS SDKs 12
Usare l'API Query IAM 12
Funzionamento di IAM 12
Componenti di una richiesta 14
Come vengono autenticati i principali 14
Nozioni di base sulle autorizzazioni e sulla policy di autorizzazione 15
..... 16
Confrontare le identità IAM e le credenziali 17
Termini 17
Differenza tra gli utenti IAM e gli utenti nel Centro identità IAM 19
Federare gli utenti da un'origine di identità esistente 20
Diversi metodi per fornire l'accesso agli utenti 21
Supportare l'accesso programmatico degli utenti 25
Come le autorizzazioni e le policy forniscono la gestione degli accessi 26
Policy e account 27
Policy e utenti 27

Policy e gruppi IAM ... 28
Sessioni e ruoli dell'utente federato ... 28
Policy basate su identità e policy basate su risorse. ... 29
Definire le autorizzazioni con l'autorizzazione ABAC ... 30
Confronto di ABAC con il modello RBAC tradizionale ... 30
Nozioni di base ... 33
Configurare il tuo Account AWS ... 34
Visualizzazione del tuo Account AWS ID ... 36
Per visualizzare il tuo Account AWS ID ... 36
Usare un alias per il tuo Account AWS ID ... 37
Creazione di un alias dell'account ... 38
Eliminazione di un alias dell'account ... 39
Pianifica l'accesso al tuo AWS account ... 40
Origini di identità ... 40
Gestione degli accessi ... 41
Casi d'uso per utenti IAM ... 42
Utilizzo dell'autenticazione a più fattori con le identità ... 55
Preparazione per le autorizzazioni con privilegi minimi ... 56
Revisione delle informazioni dell'ultimo accesso per il tuo AWS account ... 57
Generazione di una policy basata sull'attività di accesso ... 63
Utilizzo della ricerca per trovare risorse IAM ... 67
Best practice per la sicurezza e casi d'uso ... 70
Best practice di sicurezza ... 70
Richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzo di credenziali temporanee ... 71
Richiedi ai carichi di lavoro di utilizzare credenziali temporanee con ruoli IAM per accedere AWS ... 72
Richiedere l'autenticazione a più fattori (MFA) ... 73
Aggiornamento delle chiavi di accesso quando necessario per i casi d'uso che richiedono credenziali a lungo termine ... 73
Segui le best practice per proteggere le credenziali di utente root ... 75
Assegna le autorizzazioni con privilegi minimi ... 75
Inizia con AWS politiche gestite e passaggio alle autorizzazioni con privilegi minimi ... 75
Utilizzare IAM Access Analyzer per generare policy con privilegi minimi in base all'attività di accesso ... 75

Esaminare e rimuovere regolarmente utenti, ruoli, autorizzazioni, criteri e credenziali inutilizzati 76
Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso 76
Verifica dell'accesso multi-account e pubblico alle risorse con IAM Access Analyzer 76
Usa IAM Access Analyzer per convalidare le tue policy IAM per ottenere autorizzazioni sicure e funzionali 77
Stabilisci guardrail delle autorizzazioni su più account 77
Utilizzare i limiti delle autorizzazioni per delegare la gestione delle autorizzazioni all'interno di un account 77
Best practice per gli utenti root 78
Proteggi le credenziali di utente root per impedirne l'uso non autorizzato 79
Utilizza una password dell'utente root sicura per proteggere l'accesso 79
Abilita l'autenticazione a più fattori (MFA) per la sicurezza dell'utente root 80
Non creare chiavi di accesso per l'utente root 80
Utilizza l'approvazione di più persone per l'accesso come utente root laddove possibile 81
Usa un indirizzo email di gruppo per le credenziali dell'utente root 81
Limita l'accesso ai meccanismi di recupero dell'account 81
Proteggi i tuoi AWS Organizations credenziali utente root dell'account 82
Monitora l'accesso e l'utilizzo 83
Casi d'uso di business 85
Configurazione iniziale di Example Corp 85
Caso d'uso per IAM con Amazon EC2 86
Caso d'uso per IAM con Amazon S3 88
Esercitazioni 91
Delega l'accesso Account AWS tramite i ruoli 91
Considerazioni 93
Prerequisiti 94
Creare un ruolo dell'account Destination 94
Concedi autorizzazione per l'accesso al ruolo 98
Accesso al test tramite cambio di ruoli 100
Risorse aggiuntive 105
Riepilogo 106
Creazione di una policy gestita dal cliente 106
Prerequisiti 106
Fase 1: creazione della policy 107
Fase 2: collegamento della policy 108

Fase 3: test dell'accesso utente 109
Risorse correlate 109
Riepilogo 109
Uso del controllo degli accessi basato su attributi (ABAC) 109
Panoramica del tutorial 110
Prerequisiti 112
Fase 1: creazione degli utenti di test 113
Fase 2: creazione della policy ABAC 115
Fase 3: creazione di ruoli 119
Fase 4: verifica della creazione di segreti 120
Fase 5: verifica della visualizzazione dei segreti 123
Fase 6: verifica della scalabilità 125
Fase 7: verifica dell'aggiornamento e dell'eliminazione dei segreti 127
Riepilogo 129
Risorse correlate 129
Utilizzo dei tag di sessione SAML per ABAC 130
Consentire agli utenti di gestire le loro credenziali e le impostazioni MFA 134
Prerequisiti 135
Fase 1: creazione di una policy per applicare l'accesso MFA 136
Fase 2: Collegamento delle policy al gruppo di utenti di test 137
Fase 3: test dell'accesso dell'utente 138
Risorse correlate 140
Crea IdP SAML con CloudFormation 140
Prerequisiti 140
Creazione dell'IdP 141
Verifica dell'IdP 144
Delete resources (Elimina risorse) 144
Dettagli del modello 145
Modello 145
Crea un ruolo federato SAML con CloudFormation 149
Prerequisiti 150
Creazione del ruolo federato 150
Prova del ruolo federato 152
Delete resources (Elimina risorse) 153
Dettagli del modello 153
Modello 154

Crea IdP SAML e ruolo federato con CloudFormation 160
Prerequisiti 161
Creazione delle risorse 161
Test della federazione 164
Delete resources (Elimina risorse) 165
Dettagli del modello 165
Modello 166
Identità 175
Utente root IAM 176
Utenti IAM 176
Gruppi di utenti IAM 176
Ruoli IAM 176
Utente root dell'account AWS 176
Gestire centralmente l'accesso root per gli account membri 177
Centralizzare l'accesso root 178
Eseguire un'attività con privilegi 182
MFA per l'utente root 189
Modifica della password 198
Reimpostare una password dell'utente root persa o dimenticata 200
Creare chiavi di accesso per l'utente root 202
Eliminare le chiavi di accesso per l'utente root 204
Attività che richiedono credenziali dell'utente root 206
Risorse aggiuntive 208
Utenti 209
In che modo AWS identifica un utente IAM 209
Utenti IAM e credenziali 210
Utenti e autorizzazioni IAM 212
Utenti IAM e account 212
Utenti IAM come account di servizio 212
In che modo gli utenti IAM accedono a AWS 213
Creazione di un utente 218
Visualizzare gli utenti IAM 220
Rinominare un utente 222
Rimuovere un utente 224
Controllare l'accesso dell'utente alla console 228
Modificare le autorizzazioni utente 230

Password utente ... 239
Gestione delle chiavi di accesso ... 260
Multi-factor authentication ... 290
Credenziali specifiche del servizio ... 371
Ricerca di credenziali inutilizzate ... 382
Generare report di credenziali ... 386
Credenziali IAM per CodeCommit ... 393
Gestire i certificati server ... 397
Gruppi di utenti ... 402
Creare gruppi IAM ... 404
Visualizzare i gruppi IAM ... 406
Modificare gli utenti nei gruppi IAM ... 407
Collegare una policy a un gruppo di utenti ... 409
Rinominare un gruppo di utenti ... 410
Eliminazione di un gruppo IAM ... 411
Roles ... 413
Quando creare un utente IAM invece di un ruolo ... 415
Termini e concetti dei ruoli ... 416
Risorse aggiuntive ... 419
Problema del "confused deputy" ... 420
Scenari comuni ... 431
Creazione di ruoli ... 445
Gestione del ruolo ... 513
Metodi per assumere un ruolo ... 551
Fornitori di identità e federazione in AWS ... 713
Federazione con il Centro identità IAM ... 714
Federazione con IAM ... 715
Federazione con pool di identità Amazon Cognito ... 716
Risorse aggiuntive ... 716
Scenari comuni ... 716
Federazione OIDC ... 722
Federazione SAML 2.0 ... 748
Federando AWS Identità verso servizi esterni ... 790
Casi di utilizzo comune ... 791
Come funziona ... 792
Guida introduttiva alla federazione delle identità in uscita ... 792

Comprendere le dichiarazioni relative ai token ... 799
Controllo degli accessi con policy IAM ... 805
Credenziali di sicurezza temporanee ... 807
AWS STS and AWS regioni ... 807
Scenari comuni per le credenziali temporanee ... 808
Applicazioni di esempio che usano credenziali temporanee ... 810
Risorse aggiuntive per le credenziali temporanee ... 811
Confronta le AWS STS credenziali ... 811
Token di connessione al servizio ... 816
Richiedere credenziali di sicurezza temporanee ... 817
Usa credenziali temporanee con AWS risorse ... 828
Autorizzazioni per le credenziali di sicurezza temporanee ... 833
Manage (Gestione) AWS STS in un Regione AWS ... 872
AWS STS Regioni ed endpoint ... 877
Abilitare l'accesso personalizzato alla console del gestore di identità ... 882
Tag per risorse IAM ... 898
Scegli un AWS convenzione di denominazione dei tag ... 899
Regole per l'etichettatura in IAM e AWS STS ... 900
Aggiungere tag agli utenti IAM ... 903
Aggiungere tag ai ruoli IAM ... 906
Aggiungere tag alle policy gestite dal cliente ... 910
Aggiungere tag ai provider di identità OIDC ... 913
Aggiungere tag ai provider di identità SAML per IAM ... 916
Tagging dei profili dell'istanza ... 919
Aggiungere tag ai certificati server ... 921
Aggiungere tag ai dispositivi MFA virtuali ... 924
Etichetta le istanze del gestore degli accessi agli account ... 926
Passare i tag di sessione ... 929
Gestione degli accessi ... 946
Accesso alle risorse di gestione ... 948
Policy e autorizzazioni ... 948
Tipi di policy ... 949
Policy e utente root ... 956
Panoramica delle policy JSON ... 956
Grant least privilege ... 961
Policy gestite e policy inline ... 963

Perimetri di dati 974
Limiti delle autorizzazioni 981
Identità e risorse 995
Controllo dell'accesso tramite le policy 998
Controllo dell'accesso a utenti e ruoli IAM mediante i tag 1011
Controlla l'accesso alle AWS risorse utilizzando i tag 1014
Accesso alle risorse multi-account 1019
Inoltro delle sessioni di accesso 1026
Policy di esempio 1029
Gestire le policy IAM 1114
Risorse aggiuntive 1114
Creare policy IAM 1115
Convalida di policy 1125
Test delle policy 1126
Aggiunta o rimozione di autorizzazioni per le identità 1141
Controllo delle versioni delle policy IAM 1153
Modificare le policy IAM 1157
Eliminare le policy IAM 1168
Perfezionamento delle autorizzazioni mediante le informazioni di accesso 1175
Riepiloghi delle policy 1766
Riepilogo della policy (elenco di servizi) 1767
Livelli di accesso nei riepiloghi delle policy 1778
Riepilogo del servizio (elenco di operazioni) 1781
Riepilogo delle operazioni (elenco di risorse) 1787
Riepiloghi di policy di esempio 1792
Autorizzazioni necessarie per accedere alle risorse IAM 1802
Autorizzazioni per amministrare le identità IAM 1802
Autorizzazioni per lavorare in Console di gestione AWS 1804
Concedi le autorizzazioni a tutti AWS account 1805
Autorizzazioni per consentire a un servizio di accedere a un altro servizio 1805
Operazioni necessarie 1806
Esempi di policy per IAM 1807
Delega temporanea 1811
Panoramica di 1811
Come funziona la delega temporanea 1811
Avviare una richiesta di delega temporanea 1814

Esamina le richieste di delega temporanea ... 1817
Revoca l'accesso temporaneo alla delega ... 1819
Gestione delle autorizzazioni per le richieste di delega ... 1820
Notifications ... 1821
CloudTrail ... 1822
Delegazione temporanea per i partner ... 1823
Panoramica di ... 1823
Come funziona ... 1811
Qualifica del partner ... 1824
Procedura di onboarding ... 1824
Questionario per i partner ... 1825
Comprendere la tua integrazione ... 1827
Comprendere le autorizzazioni ... 1830
Linee guida di valutazione delle politiche ... 1833
Modelli di policy ... 1836
Sviluppa la tua integrazione ... 1847
Gestore dell'accesso all'account ... 1856
Cos'è l'account access manager? ... 1856
Perché usare l'account access manager? ... 1856
Funzionalità chiave ... 1858
Nozioni di base ... 1859
Prerequisiti e considerazioni ... 1860
Abilita l'account access manager ... 1860
Aggiorna firewall e gateway ... 1861
Elimina il gestore degli accessi agli account ... 1863
Gestisci l'accesso a Account AWS ... 1865
Amministrazione delegata ... 1865
Prepara i tuoi ruoli IAM ... 1868
Assegna e rimuovi l'accesso ... 1870
Visualizza o cerca le assegnazioni ... 1875
Sessioni e controllo degli accessi ... 1877
Accesso alla forza lavoro ... 1878
Portali utente ... 1878
Guida introduttiva al portale di accesso all'account ... 1879
Accesso utente a Account AWS ... 1880
Sicurezza ... 1885

Gestione delle identità e degli accessi per l'account access manager 1885
Protezione dei dati in Account Access Manager 1893
Registrazione e monitoraggio in Account Access Manager 1894
Quote 1894
Quota di assegnazione di gruppo 1894
Limiti di accelerazione 1894
Esempi di codice 1896
IAM 1903
Nozioni di base 1909
Scenari 2706
AWS STS 3200
Nozioni di base 3201
Scenari 3233
Sicurezza 3487
AWS credenziali di sicurezza 3488
Considerazioni relative alla sicurezza 3489
Accesso programmatico 3491
AWS linee guida per l'audit di sicurezza 3495
Quando è necessario eseguire un controllo di sicurezza 3495
Linee guida per l'audit 3496
Controlla le credenziali del tuo account AWS 3496
Verifica degli utenti IAM 3496
Verifica dei gruppi IAM 3497
Verifica dei ruoli IAM 3497
Verifica dei provider IAM per SAML e OpenID Connect (OIDC) 3498
Verifica le app per dispositivi mobili 3498
Suggerimenti per la verifica delle policy IAM 3499
Protezione dei dati 3500
Crittografia dei dati in IAM e AWS STS 3501
Gestione delle chiavi in IAM e AWS STS 3502
Privacy del traffico di rete in IAM e AWS STS 3502
Registrazione di log e monitoraggio 3502
Registra gli eventi con CloudTrail 3503
Tenere traccia delle attività con privilegi in CloudTrail 3531
Convalida della conformità 3535
Resilienza 3535

Best practice per la resilienza di IAM ... 3538
Sicurezza dell'infrastruttura ... 3538
Analisi della configurazione e delle vulnerabilità ... 3539
AWS politiche gestite ... 3539
IAMReadOnlyAccess ... 3540
IAMUserChangePassword ... 3540
IAMAccessAnalyzerFullAccess ... 3540
IAMAccessAnalyzerReadOnlyAccess ... 3542
AccessAnalyzerServiceRolePolicy ... 3543
IAMAuditRootUserCredentials ... 3544
IAMCreateRootUserPassword ... 3546
IAMDeleteRootUserCredentials ... 3547
S3UnlockBucketPolicy ... 3549
SQSUnlockQueuePolicy ... 3550
AccountAccessManagerServiceRolePolicy ... 3551
Aggiornamenti delle policy ... 3551
Funzioni di sicurezza al di fuori di IAM ... 3559
Sistema di analisi degli accessi IAM ... 3561
Identificazione delle risorse condivise con un'entità esterna ... 3561
Identificazione dell'accesso interno alle risorse aziendali critiche ... 3564
Identificazione dell'accesso inutilizzato concesso a utenti e ruoli IAM ... 3565
Convalida delle politiche rispetto a AWS best practice ... 3565
Convalida delle policy rispetto agli standard di sicurezza specificati ... 3566
Generazione di policy ... 3566
Prezzi per Sistema di analisi degli accessi IAM ... 3566
Esiti ... 3567
Come funzionano i risultati ... 3569
Nozioni di base su IAM Access Analyzer ... 3573
Dashboard dei risultati ... 3592
Esaminare i risultati ... 3597
Filtrare i risultati ... 3600
Archivia risultati ... 3607
Risolvere i risultati ... 3608
Esiti di errore ... 3611
Tipi di risorse supportati ... 3616
Amministratore delegato ... 3625

Regole di archiviazione ... 3627
Monitoraggio con EventBridge ... 3630
Integrazioni CSPM di Security Hub ... 3642
Registrazione con CloudTrail ... 3650
Chiavi filtro ... 3653
Uso di ruoli collegati ai servizi ... 3670
Anteprima dell'accesso ... 3673
Visualizzazione in anteprima dell'accesso nella console Amazon S3 ... 3673
Visualizzazione in anteprima dell'accesso con IAM Access Analyzer APIs ... 3675
Controlli per la convalida delle policy ... 3678
Come funzionano i controlli delle policy personalizzati ... 3679
Esempi di policy di riferimento per verificare la presenza di nuovi accessi ... 3680
Ispezione dei controlli delle policy personalizzate non riusciti ... 3681
Convalidare con i controlli di base delle policy ... 3681
Riferimento ai controlli delle policy ... 3684
Convalida con controlli delle policy personalizzate ... 3828
Generazione di policy per Sistema di analisi degli accessi AWS IAM ... 3829
Come funziona la generazione di policy ... 3830
Informazioni sul servizio e sul livello di azione ... 3831
Da sapere ... 3831
Autorizzazioni richieste ... 3832
Genera una policy basata sull' CloudTrail attività (console) ... 3836
Genera una policy utilizzando AWS CloudTrail dati in un altro account ... 3839
Genera una policy basata sull' CloudTrail attività (AWS CLI) ... 3843
Genera una politica basata sull' CloudTrail attività (AWS API) ... 3843
Servizi di generazione di policy per Sistema di analisi degli accessi IAM ... 3844
Quote Sistema di analisi degli accessi AWS IAM ... 4403
Risoluzione dei problemi relativi a IAM ... 4406
Non riesco ad accedere al mio AWS account ... 4406
Chiavi di accesso smarrite ... 4406
Variabili della policy non funzionanti ... 4407
Le modifiche che apporto non sono sempre immediatamente visibili ... 4408
Non sono autorizzato a eseguire: iam: DeleteVirtualMFADevice ... 4409
Come posso creare utenti IAM in modo sicuro? ... 4409
Risorse aggiuntive ... 4410
Messaggi di errore di accesso rifiutato ... 4411

Ricevo la dicitura «accesso negato» quando faccio una richiesta a un AWS service 4412
Messaggio di accesso rifiutato quando si effettua una richiesta con credenziali di sicurezza temporanee 4413
Esempi di accesso negato 4415
Accesso ai messaggi di errore negati con ID di autorizzazione (Anteprima) 4422
Panoramica di 4422
Concetti e definizioni 4423
Risolvi i problemi utilizzando AWS Console di gestione 4424
Risolvi i problemi relativi all'utilizzo AWS API o AWS CLI 4425
Quando non ricevi un ID di autorizzazione 4426
Ulteriori considerazioni 4426
Servizi supportati 4427
Problemi relativi agli utenti root 4427
Policy IAM 4428
Risoluzione dei problemi tramite l'editor visivo 4430
Risoluzione dei problemi tramite i riepiloghi delle policy 4435
Risoluzione dei problemi di gestione delle policy 4445
Risoluzione dei problemi relativi ai documenti di policy JSON 4446
Passkey e chiavi di sicurezza FIDO 4453
Non riesco ad abilitare la chiave di sicurezza FIDO 4453
Non riesco a effettuare l'accesso utilizzando la chiave di sicurezza FIDO 4454
Ho perso o danneggiato la mia chiave di sicurezza FIDO 4454
Altri problemi. 4455
Ruoli IAM 4455
Non è possibile assumere un ruolo 4455
È apparso un nuovo ruolo nel mio AWS account 4458
Non riesco a modificare o eliminare un ruolo nel mio Account AWS 4458
Non sono autorizzato a svolgere: iam: PassRole 4459
Perché non posso assumere un ruolo con una sessione di 12 ore? (AWS CLI, AWS API) . 4459
Viene visualizzato un errore quando provo a passare da un ruolo a un altro nella console IAM 4460
Il mio ruolo ha un policy che mi consente di eseguire un'operazione, ma ricevo "Accesso negato" 4460
Il servizio non ha creato la versione delle policy predefinite del ruolo 4461
Non esiste un caso d'uso per un ruolo di servizio nella console 4462
IAM e Amazon EC2 4463

Durante l'avvio di un'istanza, il ruolo non viene visualizzato nell'elenco Ruolo IAM nella console Amazon EC2. ... 4464
Le credenziali per l'istanza si riferiscono al ruolo errato ... 4465
Quando tento di chiamare `AddRoleToInstanceProfile`, viene visualizzato un errore `AccessDenied` ... 4465
Amazon EC2: quando provo ad avviare un'istanza con un ruolo, ricevo un errore `AccessDenied` ... 4465
Non è possibile accedere alle credenziali di sicurezza temporanee nell'istanza EC2 ... 4466
Cosa significano gli errori riportati nel documento `info` nella sottostruttura IAM? ... 4467
IAM e Amazon S3 ... 4468
Come posso concedere l'accesso anonimo a un bucket Amazon S3? ... 4468
Ho effettuato l'accesso come Account AWS utente root. Perché non riesco ad accedere a un bucket Amazon S3 con il mio account? ... 4468
Federazione SAML 2.0 ... 4469
Risposta SAML non valida ... 4470
RoleSessionName è obbligatorio ... 4470
Non autorizzato per AssumeRoleWithSAML ... 4471
Caratteri non validi RoleSessionName ... 4472
Caratteri identità di origine non validi ... 4472
Risposta di firma non valida ... 4472
Chiave privata non valida ... 4472
Impossibile rimuovere la chiave privata ... 4473
Impossibile rimuovere la chiave privata perché l'ID della chiave non corrisponde a una chiave privata. ... 4473
Impossibile assumere il ruolo. ... 4473
Impossibile analizzare i metadati ... 4473
Impossibile aggiornare il provider di identità ... 4474
Impossibile impostare la modalità di crittografia delle asserzioni su Richiesta perché non è stata fornita alcuna chiave privata. ... 4474
Errore: impossibile aggiungere e rimuovere chiavi private nella stessa richiesta ... 4475
Il provider specificato non esiste ... 4475
DurationSeconds supera MaxSessionDuration ... 4475
È stato raggiunto il limite di 2 per la chiave privata. ... 4475
La risposta non contiene il pubblico richiesto ... 4476
In che modo IAM interagisce con altri AWS servizi ... 4477
Crea risorse IAM con CloudFormation ... 4477

IAM e CloudFormation modelli ... 4478
Scopri di più su CloudFormation ... 4478
Usare AWS CloudShell con IAM ... 4478
Ottieni le autorizzazioni IAM per AWS CloudShell ... 4479
Interagire con IAM ... 4479
Lavorare con gli AWS SDK ... 4481
Documentazione di riferimento ... 4483
Identificazione AWS risorse con Amazon Resource Names (ARN) ... 4483
Formato ARN ... 4483
Ricerca del formato dell'ARN per una risorsa ... 4485
Percorsi negli ARN ... 4485
Identificatori IAM ... 4487
Nomi descrittivi e percorsi ... 4487
ARN IAM ... 4488
Identificatori univoci ... 4496
Approfondimento dei prefissi ID univoci ... 4497
Ottenere l'identificatore univoco ... 4498
IAM e AWS STS quote ... 4499
Requisiti del nome IAM ... 4499
IAM Quote oggetto ... 4500
Quote Sistema di analisi degli accessi IAM ... 4502
Quote di IAM Roles Anywhere ... 4502
Quote di richiesta STS ... 4502
Limiti di caratteri di IAM e STS ... 4503
Supporto per endpoint dual-stack ... 4508
Supporto per endpoint dual-stack con IAM ... 4508
AWS STS supporto per endpoint dual-stack ... 4509
Endpoint VPC di interfaccia ... 4510
Disponibilità dell'endpoint VPC ... 4511
Creare un endpoint VPC per IAM ... 4512
Crea un endpoint VPC per AWS STS ... 4513
Crea un endpoint VPC per AWS STS Discovery OIDC ... 4514
Controlla l'accesso a AWS STS con politiche per gli endpoint VPC ... 4517
Usa le chiavi di condizione VPC per controllare l'accesso federato ... 4525
Servizi supportati da IAM ... 4531
Servizi supportati da IAM ... 4533

Ulteriori informazioni 4563
AWS Signature versione 4 4568
In che modo AWS Sigv4 funziona 4569
In che modo AWS SigV4a funziona 4569
Quando firmare le richieste 4570
Perché le richieste vengono firmate 4570
Altre risorse 4571
Elementi di una richiesta SigV4 4572
Metodi di autenticazione 4575
Creazione di una richiesta firmata 4582
Richiesta di esempi di firma 4596
Risoluzione dei problemi relativi a SigV4 4598
Riferimento alla policy 4603
Documentazione di riferimento dell'elemento JSON 4604
Logica di valutazione delle policy 4718
Sintassi della policy 4745
AWS politiche gestite per le funzioni lavorative 4754
Chiavi della condizione globale 4772
Chiavi di condizione IAM 4858
Operazioni, risorse e chiavi di condizione 4912
Resources 4913
Identità 4913
Credenziali (password, chiavi di accesso e dispositivi MFA) 4913
Autorizzazioni e policy 4914
Federazione e delega 4914
IAM e altro AWS prodotti 4915
Uso di IAM con Amazon EC2 4915
Uso di IAM con Amazon S3 4915
Utilizzo di IAM con Amazon RDS 4915
Uso di IAM con Amazon DynamoDB 4916
Best practice generali relative alla sicurezza 4916
Risorse generali 4916
Chiamata di richieste di query HTTP 4918
Endpoints 4919
HTTPS obbligatorio 4919
Firma delle richieste API IAM 4919

Cronologia dei documenti 4921
..... mmmmcmliii

Che cos'è IAM?

Follow us on Twitter

AWS Identity and Access Management (IAM) è un servizio web che ti aiuta a controllare in modo sicuro l'accesso alle AWS risorse. Con IAM, puoi gestire le autorizzazioni che controllano a quali AWS risorse gli utenti possono accedere. Utilizza IAM per controllare chi è autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) per l'utilizzo di risorse. IAM offre l'infrastruttura necessaria per gestire l'autenticazione e l'autorizzazione per il tuo Account AWS.

Per istruzioni sulla configurazione Account AWS e sulla creazione delle identità, consulta. Nozioni di base su IAM

Identità

Quando ne crei una Account AWS, inizi con un'identità di accesso chiamata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali come utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'utente di IAM.

Usa IAM per configurare altre identità oltre al tuo utente root, come amministratori, analisti e sviluppatori, e concedere loro l'accesso alle risorse di cui hanno bisogno per portare a termine con successo le loro attività.

Gestione degli accessi

Dopo aver configurato un utente in IAM, questo utilizzerà le proprie credenziali di accesso per autenticarsi con AWS. L'autenticazione viene fornita abbinando le credenziali di accesso a un principale (un utente IAM, un principale AWS STS federato, un ruolo IAM o un'applicazione) considerato affidabile da. Account AWS Successivamente, viene fatta una richiesta per concedere al principale accesso alle risorse. L'accesso è concesso in risposta a una richiesta di autorizzazione se all'utente è stata concessa l'autorizzazione alla risorsa. Ad esempio, quando accedi per la prima volta alla console e ti trovi nella home page, non stai accedendo a un servizio specifico. Quando selezioni un servizio, la richiesta di autorizzazione viene inviata a quel servizio e verifica se la tua identità è nell'elenco degli utenti autorizzati, quali policy vengono applicate per controllare il livello di accesso concesso e qualsiasi altra policy che potrebbe essere in vigore. Le richieste di autorizzazione possono essere effettuate dai principali all'interno dell'utente Account AWS o da un altro Account AWS di cui ci si fida.

Una volta autorizzato, il principale può intervenire o eseguire operazioni sulle risorse del tuo Account AWS. Ad esempio, il principale potrebbe avviare una nuova Amazon Elastic Compute Cloud istanza, modificare l'appartenenza al gruppo IAM o eliminare i Amazon Simple Storage Service bucket.

> **Tip**
>
> AWS Skill Builder fornisce un'introduzione video di 10 minuti a IAM:
> Introduzione al sito AWS Identity and Access Management web di AWS Skill Builder.

Disponibilità del servizio

IAM, come molti altri AWS servizi, alla fine è coerente. IAM raggiunge un'alta disponibilità replicando i dati su più server nei data center di Amazon di tutto il mondo. Se una richiesta per modificare alcuni dati ha successo, la modifica viene completata e memorizzata in maniera sicura. Tuttavia, le modifiche devono essere replicate su IAM e questo può richiedere tempo. Tali modifiche includono la creazione o l'aggiornamento di utenti, gruppi, ruoli, o policy. Si consiglia di non includere tali modifiche IAM nei percorsi critici e ad alta disponibilità del codice dell'applicazione. Al contrario, apporta modifiche IAM in un'inizializzazione separata o in una routine di configurazione che si esegue meno frequentemente. Inoltre, assicurarsi di verificare che le modifiche siano state propagate prima che i flussi di lavoro di produzione dipendano da esse. Per ulteriori informazioni, consulta Le modifiche che apporto non sono sempre immediatamente visibili.

Informazioni sui costi del servizio

AWS Identity and Access Management (IAM) Centro identità AWS IAM e Servizio di token di sicurezza AWS (AWS STS) sono funzionalità del tuo AWS account offerte senza costi aggiuntivi. Ti viene addebitato solo quando accedi ad altri AWS servizi utilizzando i tuoi utenti IAM o le credenziali di sicurezza AWS STS temporanee.

L'analisi degli accessi esterni del Sistema di analisi degli accessi IAM è disponibile senza costi aggiuntivi. Tuttavia, ti verranno addebitati dei costi per l'analisi degli accessi inutilizzati e i controlli delle policy dei clienti. Per un elenco completo delle tariffe e dei prezzi specifici per Sistema di analisi degli accessi IAM, consulta la pagina dedicata.

Per informazioni sui prezzi di altri AWS prodotti, consulta la pagina dei prezzi di Amazon Web Services.

Integrazione con altri AWS servizi

IAM è integrato con molti AWS servizi. Per un elenco dei AWS servizi che funzionano con IAM e le funzionalità di supporto di IAM, consultaAWS servizi supportati da IAM.

Perché utilizzare IAM?

AWS Identity and Access Management è un potente strumento per gestire in modo sicuro l'accesso alle tue AWS risorse. Uno dei principali vantaggi dell'utilizzo di IAM è la possibilità di concedere l'accesso condiviso al tuo AWS account. Inoltre, IAM ti consente di assegnare autorizzazioni granulari, permettendo di controllare esattamente quali azioni i diversi utenti possono eseguire su risorse specifiche. Questo livello di controllo degli accessi è fondamentale per mantenere la sicurezza del tuo AWS ambiente. IAM fornisce anche diverse altre funzionalità di sicurezza. È possibile aggiungere l'autenticazione a più fattori (MFA) per un ulteriore livello di protezione e sfruttare la federazione delle identità per integrare senza problemi gli utenti della rete aziendale o altri provider di identità. IAM si integra inoltre con AWS CloudTrail, fornendo informazioni dettagliate sulla registrazione e sull'identità a supporto dei requisiti di controllo e conformità. Sfruttando queste funzionalità, puoi contribuire a garantire che l'accesso alle tue AWS risorse critiche sia strettamente controllato e sicuro.

Accesso condiviso al tuo Account AWS

Puoi concedere ad altri utenti le autorizzazioni per amministrare e utilizzare le risorse nel tuo account AWS senza la necessità di condividere la password o la chiave di accesso.

Autorizzazioni granulari

Puoi concedere autorizzazioni diverse a diverse persone per diverse risorse. Ad esempio, potresti consentire ad alcuni utenti l'accesso completo ad Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Redshift e altri servizi. AWS Per gli altri utenti, puoi consentire l'accesso in sola lettura solo ad alcuni bucket S3, oppure l'autorizzazione ad amministrare solo alcune istanze Amazon EC2 oppure l'autorizzazione ad accedere alle informazioni di fatturazione, ma nient'altro.

Accesso sicuro a AWS risorse per le applicazioni eseguite su Amazon EC2

Puoi utilizzare le funzionalità di IAM per fornire in maniera sicura le credenziali per le applicazioni che funzionano su istanze EC2. Queste credenziali forniscono all'applicazione le autorizzazioni per accedere ad altre risorse. AWS Alcuni esempi includono i bucket S3 e le tabelle DynamoDB.

Multi-factor autenticazione (MFA)

Puoi aggiungere l'autenticazione a due fattori per il tuo account e per i singoli utenti per maggiore sicurezza. Con MFA tu o i tuoi utenti dovete fornire non solo una password o la chiave di accesso che funzioni con il tuo account, ma anche un codice da un dispositivo appositamente configurato. Se utilizzi già una chiave di sicurezza FIDO con altri servizi e questa ha una configurazione AWS supportata, puoi utilizzarla WebAuthn per la sicurezza MFA. Per ulteriori informazioni, consulta Configurazioni supportate per l'utilizzo di passkey e chiavi di sicurezza

Federazione delle identità

Puoi consentire agli utenti che utilizzano già le password altrove, ad esempio nella tua rete aziendale o con un provider di identità Internet, di ottenere l'accesso temporaneo al tuo Account AWS. A questi utenti vengono concesse credenziali temporanee conformi alle raccomandazioni delle best practice di IAM. L'utilizzo della federazione delle identità migliora la sicurezza del tuo account AWS .

Informazioni d'identità per la sicurezza

Se utilizzi AWS CloudTrail riceverai i record del log che includono le informazioni su chi effettua le richieste per le risorse nel tuo account. Queste informazioni sono basate sulle identità IAM.

Conformità PCI DSS

IAM supporta l'elaborazione, l'archiviazione e la trasmissione di dati di carte di credito da parte di un esercente o di un provider di servizi, oltre a essere conforme allo standard Payment Card Industry Data Security Standard (PCI DSS). Per ulteriori informazioni su PCI DSS, incluso come richiedere una copia del PCI Compliance Package, vedi AWS PCI DSS Level 1.

Quando si usa IAM?

AWS Identity and Access Management è un servizio di infrastruttura di base che fornisce le basi per il controllo degli accessi basato sulle identità interne. AWS Usi IAM ogni volta che accedi al tuo AWS account. Il modo in cui utilizzi IAM dipenderà dalle responsabilità e dalle funzioni dei processi specifiche all'interno della tua organizzazione. Gli utenti dei AWS servizi utilizzano IAM per accedere alle AWS risorse necessarie per il loro lavoro quotidiano, con gli amministratori che concedono le autorizzazioni appropriate. Gli amministratori IAM, d'altra parte, sono responsabili della gestione delle identità IAM e della stesura di policy per controllare l'accesso alle risorse. Indipendentemente

dal tuo ruolo, interagisci con IAM ogni volta che autentichi e autorizzi l'accesso alle risorse. AWS Ciò potrebbe comportare l'accesso come utente IAM, l'assunzione di un ruolo IAM o l'uso della federazione delle identità per un accesso senza interruzioni. Comprendere le varie funzionalità e casi d'uso di IAM è fondamentale per gestire efficacemente l'accesso sicuro all'ambiente. AWS Quando si tratta di creare policy e autorizzazioni, IAM offre un approccio flessibile e granulare. È possibile definire policy di attendibilità per controllare quali principali possono assumere un ruolo, oltre a policy basate sull'identità che specificano le azioni e le risorse a cui un utente o un ruolo può accedere. Configurando queste policy IAM, puoi contribuire a garantire che gli utenti e le applicazioni dispongano del livello di autorizzazioni appropriato per eseguire le attività richieste.

Quando si eseguono diverse funzioni lavorative

AWS Identity and Access Management è un servizio di infrastruttura di base che fornisce le basi per il controllo degli accessi basato sulle identità interne AWS. IAM viene utilizzato ogni volta che accedi al tuo account AWS .

Le modalità di utilizzo di IAM cambiano in base alle operazioni eseguite in AWS.

- Utente del servizio: se utilizzi un AWS servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità utilizzate per il lavoro, potrebbero essere necessarie altre autorizzazioni. La comprensione della gestione dell'accesso consente di richiedere le autorizzazioni corrette all'amministratore.
- Amministratore del servizio: se sei responsabile di una AWS risorsa presso la tua azienda, probabilmente hai pieno accesso a IAM. Il tuo compito è determinare le funzionalità e le risorse IAM a cui gli utenti del servizio devono accedere. Devi quindi inviare le richieste all'amministratore IAM per modificare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM.
- Amministratore IAM: se sei un amministratore IAM, puoi gestire le identità IAM e scrivere policy per gestire l'accesso ad IAM.

Quando sei autorizzato ad accedere AWS risorse

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali da una fonte di identità come Centro identità AWS IAM (IAM Identity Center), autenticazione Single Sign-on o credenziali. Google/

Facebook Per ulteriori informazioni sull'accesso, consulta Come accedere all' Account AWS nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta AWS Signature Version 4 per le richieste API nella Guida per l'utente di IAM.

Quando accedi come utente IAM

Un utente IAM è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella IAM User Guide.

Un gruppo IAM specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente di IAM.

Quando assumi un ruolo IAM

Un ruolo IAM è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo passando da un utente a un ruolo IAM (console) o chiamando un'operazione AWS API AWS CLI or. Per ulteriori informazioni, consulta Metodi per assumere un ruolo nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Quando crei policy e autorizzazioni

Concedi le autorizzazioni a un utente creando una policy che è un documento che elenca le operazioni che un utente può eseguire e le risorse che tali operazioni possono influenzare. Qualsiasi operazione o risorsa che non è esplicitamente consentita viene negata come impostazione predefinita. Le policy possono essere create e collegate ai principali (utenti, gruppi di utenti, ruoli assunti da utenti e risorse).

Puoi utilizzare queste policy con un ruolo IAM:

- Policy di attendibilità: definisce quali principali possono assumere il ruolo e a quali condizioni. Una policy di attendibilità è un tipo specifico di policy basata sulle risorse per i ruoli IAM. Un ruolo può avere una sola policy di attendibilità.
- Identity-based policy (in linea e gestite): queste policy definiscono le autorizzazioni che l'utente del ruolo è in grado di eseguire (o gli viene negato di eseguire) e su quali risorse.

Utilizza gli Esempi di policy basate su identità IAM per definire le autorizzazioni per le identità IAM. Una volta trovata la policy desiderata, seleziona view the policy (visualizza la policy) per consultare il JSON della policy. Puoi utilizzare il documento di policy JSON come modello per le tue policy.

> **Note**
>
> Se utilizzi il Centro identità IAM per gestire i tuoi utenti, assegni set di autorizzazioni nel Centro identità IAM invece di collegare una policy di autorizzazioni a un principale. Quando si assegna un set di autorizzazioni a un gruppo o utente in Centro identità AWS IAM, IAM Identity Center crea i ruoli IAM corrispondenti in ciascun account e allega le politiche specificate nel set di autorizzazioni a tali ruoli. Il Centro identità IAM gestisce il ruolo e consente agli utenti autorizzati che hai definito di assumerlo. Se modifichi il set di autorizzazioni, il Centro identità IAM garantisce che le policy e i ruoli IAM corrispondenti vengano aggiornati di conseguenza.
> Puoi anche utilizzare l'account access manager , una funzionalità IAM che ti consente di assegnare i ruoli IAM esistenti che crei e gestisci agli utenti e ai gruppi di IAM Identity Center, dandoti accesso al set completo di funzionalità dei ruoli IAM. Puoi utilizzarlo insieme ai set di autorizzazioni o da solo.
> Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di Centro identità AWS IAM .

Come viene gestito IAM?

La gestione AWS Identity and Access Management all'interno di un AWS ambiente implica l'utilizzo di una varietà di strumenti e interfacce. Il metodo più comune è l' Console di gestione AWS interfaccia basata sul Web che consente di eseguire un'ampia gamma di attività amministrative IAM, dalla creazione di utenti e ruoli alla configurazione delle autorizzazioni.

Per gli utenti che si sentono più a proprio agio con le interfacce a riga di comando, AWS fornisce due set di strumenti da riga di comando: il e il. AWS Command Line Interface AWS Tools for Windows

PowerShell Questi consentono di emettere comandi relativi a IAM direttamente dal terminale, spesso in modo più efficiente rispetto alla navigazione nella console. Inoltre, AWS CloudShell consente di eseguire comandi CLI o SDK direttamente dal browser Web, utilizzando le autorizzazioni associate all'accesso alla console.

Oltre alla console e alla riga di comando, AWS offre Software Development Kit (SDKs) per vari linguaggi di programmazione, che consentono di integrare le funzionalità di gestione IAM direttamente nelle applicazioni. In alternativa, puoi accedere a IAM a livello di codice usando l'API Query IAM che ti consente di inviare richieste HTTPS direttamente al servizio. L'utilizzo di questi diversi approcci di gestione offre la flessibilità necessaria per incorporare IAM nei flussi di lavoro e nei processi esistenti.

Usa il Console di gestione AWS

La console di AWS gestione è un'applicazione Web che comprende e fa riferimento a un'ampia raccolta di console di servizio per la gestione AWS delle risorse. Quando effettui l'accesso per la prima volta, visualizzi la home page della console. La home page fornisce l'accesso a ciascuna console di servizio e offre un'unica posizione per accedere alle informazioni per l'esecuzione delle attività AWS correlate. I servizi e le applicazioni disponibili dopo l'accesso alla console dipendono dalle AWS risorse a cui si è autorizzati ad accedere. È possibile ottenere le autorizzazioni per le risorse assumendo un ruolo, facendo parte di un gruppo al quale sono state concesse le autorizzazioni oppure ricevendo un'autorizzazione esplicita. Per un account AWS autonomo, l'accesso alle risorse viene configurato dall'utente root o dall'amministratore IAM. Per AWS Organizations, l'accesso alle risorse viene configurato dall'account di gestione o dall'amministratore delegato.

Se prevedi che persone utilizzino la console di AWS gestione per gestire AWS le risorse, ti consigliamo di configurare gli utenti con credenziali temporanee come best practice di sicurezza. Gli utenti IAM che hanno assunto un ruolo, i principali federati e gli utenti in Centro identità IAM dispongono di credenziali temporanee, mentre l'utente IAM e l'utente root dispongono di credenziali a lungo termine. Le credenziali dell'utente root forniscono l'accesso completo a Account AWS, mentre gli altri utenti dispongono di credenziali che forniscono l'accesso alle risorse concesse loro dalle politiche IAM.

L'esperienza di accesso è diversa per i diversi tipi di utenti. Console di gestione AWS

- Gli utenti IAM e l'utente root accedono dall'URL di AWS accesso principale (). https://signin.aws.amazon.com Una volta effettuato l'accesso, hanno accesso alle risorse dell'account per il quale hanno ricevuto l'autorizzazione.

Per accedere come utente root è necessario disporre dell'indirizzo e-mail e della password dell'utente root.

Per accedere come utente IAM devi disporre del Account AWS numero o dell'alias, del nome utente IAM e della password utente IAM.

Ti consigliamo di limitare gli utenti IAM del tuo account a situazioni specifiche che richiedono credenziali a lungo termine, ad esempio per l'accesso di emergenza, e di utilizzare l'utente root solo per le attività che richiedono le credenziali dell'utente root.

Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. La volta successiva che l'utente accede a qualsiasi pagina di Console di gestione AWS, la console utilizza il cookie per reindirizzare l'utente alla pagina di accesso dell'account.

Per evitare che le tue credenziali vengano riutilizzate dopo il tuo accesso, esci dalla console al termine della sessione.

- Gli utenti di IAM Identity Center accedono utilizzando un portale di AWS accesso specifico, unico per la loro organizzazione. Una volta effettuato l'accesso, possono scegliere a quale account o applicazione accedere. Se scelgono di accedere a un account, scelgono quale set di autorizzazioni utilizzare per la sessione di gestione.
- I principali federati OIDC e SAML gestiti in un provider di identità esterno collegato a un Account AWS eseguono l'accesso tramite un portale di accesso aziendale personalizzato. Le risorse AWS disponibili per gli utenti dipendono dalle policy selezionate dall'organizzazione.

> **Note**
>
> Per fornire un ulteriore livello di sicurezza, l'utente root, gli utenti IAM e gli utenti di IAM Identity Center possono far verificare l'autenticazione a più fattori (MFA) prima AWS di concedere l'accesso alle risorse. AWS Quando l'MFA è abilitata, devi avere accesso anche al dispositivo MFA per accedere.

Per ulteriori informazioni su come diversi utenti accedono alla console di gestione, consulta Accedere alla console di AWS gestione nella Guida per l'utente di accesso.AWS

AWS Strumenti da riga di comando

È possibile utilizzare gli strumenti della riga di AWS comando per impartire comandi dalla riga di comando del sistema per eseguire IAM e AWS attività. L'utilizzo della riga di comando può essere più veloce e semplice rispetto all'uso della console. Gli strumenti da riga di comando sono utili anche se desideri creare script che eseguano AWS attività.

AWS fornisce due set di strumenti da riga di comando: the AWS Command Line Interface(AWS CLI) e the AWS Tools for Windows PowerShell. Per informazioni sull'installazione e l'utilizzo di AWS CLI, consulta la Guida AWS Command Line Interface per l'utente. Per informazioni sull'installazione e l'utilizzo degli strumenti per Windows PowerShell, consulta la Guida per AWS Strumenti per PowerShell l'utente.

Dopo aver effettuato l'accesso alla console, puoi utilizzarla AWS CloudShell dal tuo browser per eseguire i comandi CLI o SDK. Le autorizzazioni per l'accesso alle AWS risorse si basano sulle credenziali utilizzate per accedere alla console. A seconda della tua esperienza, potresti ritenere che la CLI sia un metodo più efficiente per gestire il tuo Account AWS. Per ulteriori informazioni, consulta Utilizzo AWS CloudShell con cui lavorare AWS Identity and Access Management

AWS Interfaccia a riga di comando (CLI) e kit di sviluppo software () SDKs

Gli utenti di IAM Identity Center e IAM utilizzano metodi diversi per autenticare le proprie credenziali quando si autenticano tramite la CLI o le interfacce applicative () nell'area associata. APIs SDKs

Le credenziali e le impostazioni di configurazione si trovano in più posizioni, come le variabili di sistema o di ambiente utente, i file di AWS configurazione locali o sono dichiarate esplicitamente sulla riga di comando come parametro. Alcune posizioni hanno la precedenza su altre.

Sia Centro identità IAM sia IAM forniscono chiavi di accesso che possono essere utilizzate con la CLI o l'SDK. Le chiavi di accesso Centro identità IAM sono credenziali temporanee che possono essere aggiornate automaticamente e sono consigliate rispetto alle chiavi di accesso a lungo termine associate agli utenti IAM.

Puoi gestire l' Account AWS utilizzo della CLI o dell'SDK AWS CloudShell dal tuo browser. Se utilizzi CloudShell per eseguire comandi CLI o SDK, devi prima accedere alla console. Le autorizzazioni per l'accesso alle AWS risorse si basano sulle credenziali utilizzate per accedere alla console. A seconda della tua esperienza, potresti ritenere che la CLI sia un metodo più efficiente per gestire il tuo Account AWS.

Per lo sviluppo di applicazioni, puoi scaricare la CLI o l'SDK sul tuo computer e accedere dal prompt dei comandi o da una finestra Docker. In questo scenario, configuri l'autenticazione e le credenziali di accesso come parte dello script della CLI o dell'applicazione SDK. È possibile configurare l'accesso a livello di programmazione alle risorse in diversi modi, a seconda dell'ambiente e dell'accesso a disposizione.

- Le opzioni consigliate per l'autenticazione del codice locale con il AWS servizio sono IAM Identity Center e IAM Roles Anywhere
- Le opzioni consigliate per l'autenticazione del codice in esecuzione all'interno di un ambiente AWS consistono nell'utilizzare i ruoli IAM o le credenziali Centro identità IAM.

Quando accedi utilizzando il portale di AWS accesso, puoi ottenere credenziali a breve termine dalla pagina iniziale in cui scegli il tuo set di autorizzazioni. Queste credenziali hanno una durata definita e non si aggiornano automaticamente. Se desideri utilizzare queste credenziali, dopo aver effettuato l'accesso al AWS portale, scegli il set di autorizzazioni Account AWS e quindi scegli. Seleziona Accesso da riga di comando o accesso programmatico per visualizzare le opzioni che puoi utilizzare per accedere alle AWS risorse a livello di codice o dalla CLI. Per ulteriori informazioni su questi metodi, consulta la pagina Ottenimento e aggiornamento di credenziali temporanee nella Guida per l'utente di Centro identità IAM. Queste credenziali vengono spesso utilizzate durante lo sviluppo di applicazioni per testare rapidamente il codice.

Ti consigliamo di utilizzare le credenziali IAM Identity Center che si aggiornano automaticamente durante l'automazione dell'accesso alle risorse. AWS Se hai configurato utenti e set di autorizzazioni in Centro identità IAM, utilizza il comando `aws configure sso` per impiegare una procedura guidata da linea di comando che ti aiuterà a identificare le credenziali a tua disposizione e a memorizzarle in un profilo. Per ulteriori informazioni sulla configurazione del profilo, consulta la pagina Configurazione del profilo con la procedura guidata `aws configure sso` della Guida per l'utente dell'interfaccia della linea di comando AWS per la versione 2.

> **Note**
>
> Molte applicazioni di esempio utilizzano chiavi di accesso a lungo termine associate agli utenti IAM o all'utente root. È consigliabile utilizzare le credenziali a lungo termine solo all'interno di un ambiente di sperimentazione (sandbox) come parte di un'esercitazione. Esamina le alternative alle chiavi di accesso a lungo termine e pianifica la transizione del codice per utilizzare credenziali alternative, come le credenziali Centro identità IAM o i ruoli IAM, il prima possibile. Dopo la transizione del codice, elimina le chiavi di accesso.

Per ulteriori informazioni sulla configurazione della CLI, consulta Installare o aggiornare la versione più recente della AWS CLI nella Guida per l'utente dell'interfaccia a riga di comando per AWS la versione 2 e Credenziali di autenticazione e accesso nella Guida per l'utente dell'interfaccia a AWS riga di comando

Per ulteriori informazioni sulla configurazione dell'SDK, consulta l'autenticazione di IAM Identity Center nella and Tools Reference Guide AWS SDKs e IAM Roles Anywhere nella and Tools Reference Guide.AWS SDKs

Usa la AWS SDKs

AWS fornisce SDKs (kit di sviluppo software) costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione (Java, Python, Ruby, .NET, iOS, Android, ecc.). SDKs Forniscono un modo conveniente per creare un accesso programmatico a IAM e. AWS Ad esempio, SDKs si occupano di attività come la firma crittografica delle richieste, la gestione degli errori e il ritentativo automatico delle richieste. Per informazioni su AWS SDKs, incluso come scaricarli e installarli, consulta la pagina Tools for Amazon Web Services.

Usare l'API Query IAM

Puoi accedere a IAM e in modo AWS programmatico utilizzando l'API IAM Query, che consente di inviare richieste HTTPS direttamente al servizio. Quando utilizzi l'API Query, devi includere il codice per firmare in modo digitale le richieste utilizzando le tue credenziali. Per ulteriori informazioni, consulta Chiamata all'API IAM utilizzando le richieste di query HTTP e Documentazione di riferimento dell'API IAM.

Funzionamento di IAM

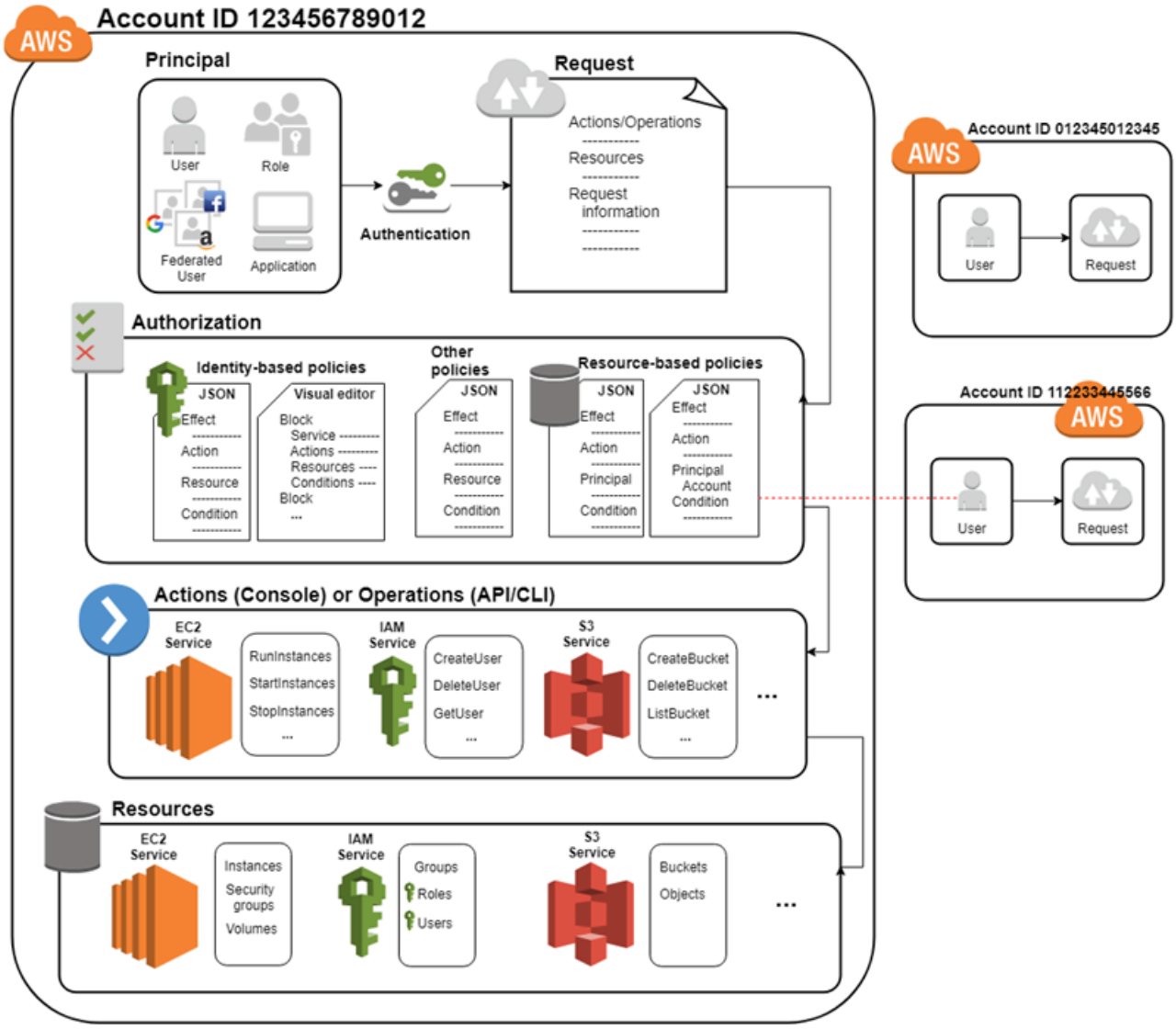
AWS Identity and Access Management fornisce l'infrastruttura necessaria per controllare l'autenticazione e l'autorizzazione per il tuo Account AWS.

Innanzitutto, per autenticarsi con AWS, un utente umano o un'applicazione utilizza le proprie credenziali di accesso. IAM abbina le credenziali di accesso a un principale (un utente IAM, un utente principale AWS STS federato, un ruolo o un'applicazione IAM) considerato attendibile Account AWS e autentica il principale per l'accesso. AWS

Successivamente, IAM effettua una richiesta per concedere al principale l'accesso alle risorse. IAM concede o nega l'accesso in risposta a una richiesta di autorizzazione. Ad esempio, quando accedi

per la prima volta alla console e ti trovi nella home page, non stai accedendo a un servizio specifico. Quando selezioni un servizio, invii una richiesta di autorizzazione a IAM per tale servizio. IAM verifica che la tua identità sia nell'elenco degli utenti autorizzati, determina quali policy controllano il livello di accesso concesso e valuta qualsiasi altra policy che potrebbe essere in vigore. Gli amministratori interni Account AWS o esterni di cui ti fidi possono effettuare richieste Account AWS di autorizzazione.

Una volta autorizzato, il principale può eseguire azioni o operazioni sulle risorse del tuo Account AWS. Ad esempio, il principale potrebbe lanciare una nuova Amazon Elastic Compute Cloud istanza, modificare l'appartenenza al gruppo IAM o eliminare i Amazon Simple Storage Service bucket. Il diagramma seguente illustra questo processo nell'infrastruttura IAM:



Componenti di una richiesta

Quando un principale tenta di utilizzare l' Console di gestione AWS AWS API o il AWS CLI, quel principale invia una richiesta a AWS. La richiesta include le informazioni seguenti:

- Azioni o operazioni: le azioni o le operazioni che il principale desidera eseguire, ad esempio un'azione nel Console di gestione AWS, o un'operazione nell' AWS API AWS CLI o.
- Risorse: l'oggetto AWS risorsa su cui il principale richiede di eseguire un'azione o un'operazione.
- Principale – Persona o applicazione che utilizza un'entità (utente o ruolo) per inviare la richiesta. Le informazioni sul principale includono le policy di autorizzazione.
- Dati di ambiente: le informazioni sull'indirizzo IP, l'agente utente, lo stato abilitato per SSL e il timestamp.
- Dati delle risorse: i dati relativi alla risorsa richiesta, ad esempio un nome di tabella DynamoDB o un tag su un'istanza Amazon EC2.

AWS raccoglie le informazioni sulla richiesta in un contesto di richiesta, che IAM valuta per autorizzare la richiesta.

Come vengono autenticati i principali

Un principale accede AWS utilizzando le proprie credenziali che IAM autentica per consentire al principale di inviare una richiesta a. AWS Alcuni servizi, come Amazon S3 e Amazon AWS STS, consentono richieste specifiche da parte di utenti anonimi. Tuttavia, si tratta di un'eccezione alla regola. Ogni tipo di utente viene sottoposto all'autenticazione.

- Utente root: le credenziali di accesso utilizzate per l'autenticazione sono l'indirizzo email utilizzato per creare Account AWS e la password specificata in quel momento.
- Principal federato: il tuo provider di identità ti autentica e trasmette le tue credenziali a AWS, a cui non devi accedere direttamente. AWS Sia il Centro identità IAM che IAM supportano la federazione delle identità.
- Utenti registrati Elenco AWS IAM Identity Center (non federati): gli utenti creati direttamente nella directory predefinita di IAM Identity Center accedono utilizzando il portale di AWS accesso e forniscono nome utente e password.
- Utente IAM: accedi fornendo il tuo ID account o alias, il nome utente e la password. Per autenticare i carichi di lavoro dall'API AWS CLI, potresti utilizzare credenziali temporanee assumendo un ruolo o potresti utilizzare credenziali a lungo termine fornendo la chiave di accesso e la chiave segreta.

Per ulteriori informazioni sulle entità IAM, consulta Utenti IAM e Ruoli IAM.

AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) con tutti gli utenti per aumentare la sicurezza del proprio account. Per ulteriori informazioni su MFA, consulta AWS Multi-factor autenticazione in IAM.

Nozioni di base sulle autorizzazioni e sulla policy di autorizzazione

L'autorizzazione si riferisce al principale che dispone delle autorizzazioni necessarie per completare la richiesta. Durante l'autorizzazione, IAM identifica le policy che si applicano alla richiesta utilizzando i valori dal contesto della richiesta. Quindi, utilizza le policy per determinare se accettare o rifiutare la richiesta. IAM memorizza la maggior parte delle policy di autorizzazione sotto forma di documenti JSON che specificano le autorizzazioni per le entità principali.

Vi sono diversi tipi di policy che possono influire su una richiesta di autorizzazione. Per fornire agli utenti le autorizzazioni per accedere alle AWS risorse del tuo account, puoi utilizzare politiche basate sull'identità. Resource-based le politiche possono concedere l'accesso tra più account. ??? Se devi effettuare una richiesta in un account differente, una policy nell'altro account deve consentirti di accedere alla risorsa e l'entità IAM che utilizzi per effettuare la richiesta deve avere una policy basata su un'identità che consenta la richiesta.

IAM controlla ogni policy applicabile al contesto della richiesta. La valutazione della policy IAM utilizza una negazione esplicita, il che significa che se una singola policy di autorizzazione include un'operazione negata, IAM nega l'intera richiesta e interrompe la valutazione. Poiché le richieste vengono rifiutate per impostazione predefinita, le policy di autorizzazione applicabili devono consentire ogni parte della richiesta perché IAM autorizzi la richiesta. La logica di valutazione per una richiesta all'interno di un singolo account segue queste regole di base:

- Come impostazione predefinita, tutte le richieste vengono negate. (In generale, le richieste effettuate utilizzando le credenziali Utente root dell'account AWS per risorse nell'account sono sempre consentite).
- Un'autorizzazione esplicita in una policy di autorizzazione qualsiasi (basata su identità o basata su risorse) sostituisce questa impostazione predefinita.
- L'esistenza di una politica di controllo del AWS Organizations servizio (SCP) o di una politica di controllo delle risorse (RCP), del limite delle autorizzazioni IAM o di una politica di sessione ha la precedenza sull'autorizzazione. Se esiste uno o più di questi tipi di policy, devono tutti consentire la richiesta. In caso contrario, viene rifiutata implicitamente. Per ulteriori informazioni su SCP e

RCP, consulta Policy di autorizzazione in AWS Organizations nella Guida per l'utente di AWS Organizations .

- Un rifiuto esplicito in una policy sostituisce qualsiasi permesso in qualsiasi policy.

Per ulteriori informazioni, consulta Logica di valutazione delle policy.

Dopo che IAM ha autenticato e autorizzato il principale, IAM approva le azioni o le operazioni contenute nella richiesta valutando la policy di autorizzazione applicabile al principale. Ogni AWS servizio definisce le azioni (operazioni) che supporta e include le operazioni che è possibile eseguire su una risorsa, come la visualizzazione, la creazione, la modifica e l'eliminazione di tale risorsa. La policy di autorizzazione che si applica al principale deve includere le azioni necessarie per eseguire un'operazione. Per ulteriori informazioni su come IAM valuta le policy di autorizzazione, consultathe section called "Logica di valutazione delle policy".

Il servizio definisce un insieme di azioni che un principale può eseguire su ogni risorsa. Quando crei policy di autorizzazione, assicurati di includere le azioni che desideri che l'utente sia in grado di eseguire. Ad esempio, IAM supporta circa 40 azioni per una risorsa di utente, incluse le seguenti azioni di base:

- `CreateUser`
- `DeleteUser`
- `GetUser`
- `UpdateUser`

Inoltre, è possibile specificare condizioni nella policy di autorizzazione che consentano l'accesso alle risorse quando la richiesta soddisfa le condizioni specificate. Ad esempio, potresti volere che una istruzione di policy diventi effettiva solo dopo una data specifica o che consenta l'accesso solo quando nella richiesta API è presente un valore specifico. Per specificare le condizioni, è possibile utilizzare l'elemento `Condition` di un'istruzione di policy.

Dopo che IAM ha approvato le operazioni nella richiesta, il principale può lavorare con le risorse correlate all'interno dell'account. Una risorsa è un oggetto esistente all'interno di un servizio. Esempi sono un'istanza Amazon EC2, un utente IAM e un bucket Amazon S3. Se il principale crea una richiesta per eseguire un'azione su una risorsa che non è inclusa nella policy di autorizzazione, il servizio nega la richiesta. Ad esempio, se disponi dell'autorizzazione per eliminare un ruolo IAM ma richiedi di eliminare un gruppo IAM, la richiesta ha esito negativo se non disponi dell'autorizzazione

per eliminare i gruppi IAM. Per saperne di più sulle azioni, le risorse e le chiavi di condizione supportate dai diversi AWS servizi, vedi Azioni, risorse e chiavi di condizione per AWS i servizi.

Confrontare le identità IAM e le credenziali

Le identità gestite AWS Identity and Access Management sono utenti IAM, ruoli IAM e gruppi IAM. Queste identità si aggiungono all'utente root AWS creato insieme al tuo. Account AWS

È vivamente consigliato di non utilizzare l'utente root per le attività quotidiane, anche quelle amministrative. Invece, fornisci ulteriori utenti e concedi loro le autorizzazioni necessarie per eseguire le attività necessarie. Puoi aggiungere utenti aggiungendo persone alla tua directory del Centro identità IAM, federando un provider di identità esterno con il Centro identità IAM o IAM o creando utenti IAM con privilegi minimi.

Per una maggiore sicurezza, ti consigliamo di centralizzare l'accesso root per aiutarti a proteggere centralmente le credenziali dell'utente root dell'utilizzo gestito. Account AWS AWS OrganizationsGestire centralmente l'accesso root per gli account membriconsente di rimuovere e impedire a livello centralizzato il ripristino a lungo termine delle credenziali degli utenti root, impedendo l'accesso root involontario su larga scala. Dopo aver abilitato l'accesso root centralizzato, è possibile ipotizzare una sessione con privilegi per eseguire azioni sugli account membri.

Dopo aver configurato gli utenti, puoi concedere l'accesso ai tuoi Account AWS a persone specifiche e fornire loro le autorizzazioni per accedere alle risorse.

Come best practice, AWS è consigliabile richiedere agli utenti umani di assumere un ruolo IAM per accedere AWS in modo che utilizzino credenziali temporanee. Se gestisci le identità nella directory del Centro identità IAM o utilizzi la federazione con un provider di identità, stai seguendo le best practice.

Termini

Questi termini sono comunemente usati quando si lavora con le identità IAM:

Risorsa IAM

Il servizio IAM archivia queste risorse. Puoi aggiungerle, modificarle e rimuoverle dalla console.

- Utente IAM
- Gruppo IAM

- Ruolo IAM
- Policy di autorizzazione
- Identity-provider oggetto

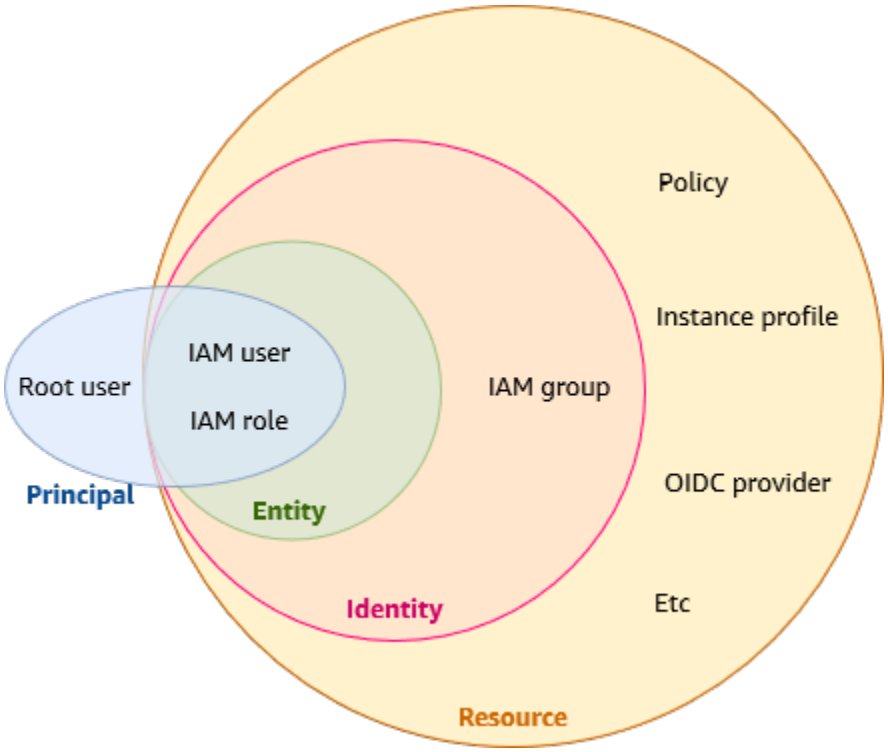
Entità IAM

risorse IAM AWS utilizzate per l'autenticazione. Specifica l'entità come principale in una policy basata sulle risorse.

- Utente IAM
- Ruolo IAM

Identità IAM

La risorsa IAM autorizzata nelle policy per eseguire azioni e accedere alle risorse. Le identità includono utenti IAM, ruoli IAM e gruppi IAM.



Principali

Un Utente root dell'account AWS utente IAM o un ruolo IAM che può effettuare una richiesta per un'azione o un'operazione su una AWS risorsa. I principali includono utenti umani, carichi di lavoro, principali federati e ruoli assunti. Dopo l'autenticazione, IAM concede al principale le credenziali permanenti o temporanee a cui effettuare le richieste AWS, a seconda del tipo di principale.

Gli utenti umani sono noti anche come identità umane, sono le persone, gli amministratori, gli sviluppatori, gli operatori e i consumatori delle tue applicazioni.

I carichi di lavoro sono una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione, un processo, strumenti operativi e altri componenti.

I federated principal sono utenti la cui identità e credenziali sono gestite da un altro provider di identità, come Active Directory, Okta o Microsoft Entra.

I ruoli IAM sono un'identità IAM che puoi creare nell'account che ha le autorizzazioni specifiche che determinano ciò che l'identità può e non può fare. Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque.

IAM concede agli utenti IAM e all'utente root credenziali a lungo termine e credenziali temporanee ai ruoli IAM. utenti in Centro identità AWS IAM, gli amministratori federati OIDC e SAML assumono i ruoli IAM al momento dell'accesso, il che concede loro credenziali temporanee. AWS Come best practice, consigliamo di richiedere agli utenti umani e ai carichi di lavoro di accedere alle risorse utilizzando credenziali temporanee. AWS

Differenza tra gli utenti IAM e gli utenti nel Centro identità IAM

Gli utenti IAM non sono account separati, ma singoli utenti all'interno del tuo account. Ogni utente ha la propria password per accedere a. Console di gestione AWS Puoi anche assegnare a ciascun utente una diversa chiave di accesso, per consentirgli di apportare richieste programmatiche e utilizzare le risorse del tuo account.

Gli utenti IAM e le relative chiavi di accesso dispongono di credenziali a lungo termine per AWS le tue risorse. L'uso principale per gli utenti IAM è fornire ai carichi di lavoro che non possono utilizzare i ruoli IAM la possibilità di effettuare richieste programmatiche ai AWS servizi utilizzando l'API o la CLI.

> **Note**
>
> Per gli scenari in cui sono necessari utenti IAM con accesso a livello di programmazione e credenziali a lungo termine, si consiglia di aggiornare le chiavi di accesso all'occorrenza. Per ulteriori informazioni, consulta Aggiornare le chiavi di accesso.

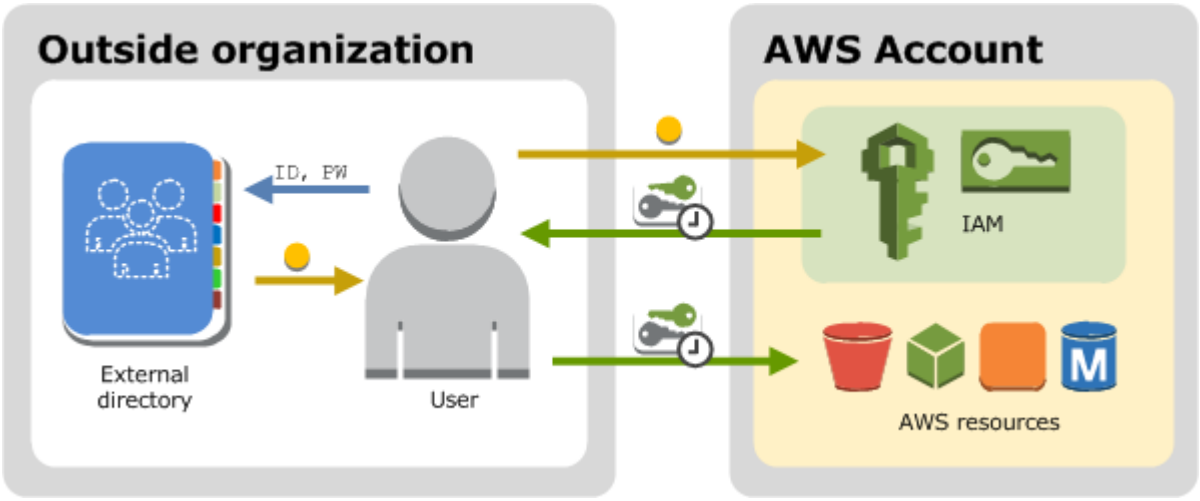
Le identità della forza lavoro (persone) hanno esigenze di autorizzazione diverse a seconda del ruolo utenti in Centro identità AWS IAM che svolgono e possono lavorare in vari modi all'interno

dell'organizzazione. Account AWS Se hai casi d'uso che richiedono chiavi di accesso, puoi supportarli con. utenti in Centro identità AWS IAM Le persone che effettuano l'accesso tramite il portale di AWS accesso possono ottenere chiavi di accesso con credenziali a breve termine alle tue AWS risorse. Per una gestione centralizzata degli accessi, consigliamo di utilizzare Centro identità AWS IAM (IAM Identity Center) per gestire l'accesso ai tuoi account e le autorizzazioni all'interno di questi account. IAM Identity Center viene configurato automaticamente con una directory Identity Center come fonte di identità predefinita in cui puoi aggiungere persone e gruppi e assegnare il loro livello di accesso alle tue AWS risorse. Per ulteriori informazioni, consulta Che cos'è Centro identità AWS IAM? nella Guida per l'utente di Centro identità AWS IAM .

La principale differenza tra questi due tipi di utenti è che gli utenti in IAM Identity Center assumono automaticamente un ruolo IAM quando effettuano l'accesso AWS prima di accedere alla console di gestione o alle AWS risorse. I ruoli IAM concedono credenziali temporanee ogni volta che l'utente accede. AWS Affinché gli utenti IAM possano accedere utilizzando un ruolo IAM, devono disporre dell'autorizzazione per assumere e cambiare ruolo e devono scegliere esplicitamente di passare al ruolo che desiderano assumere dopo aver effettuato l'accesso all'account. AWS

Federare gli utenti da un'origine di identità esistente

Se gli utenti dell'organizzazione sono già autenticati quando accedono alla rete aziendale, non sarà necessario creare utenti IAM o utenti nel Centro identità IAM separati. Invece, puoi federare tali identità utente AWS utilizzando IAM o. Centro identità AWS IAM I principali federati OIDC e SAML assumono un ruolo IAM che fornisce loro le autorizzazioni per accedere a risorse specifiche. Per ulteriori informazioni sui ruoli, consulta Termini e concetti dei ruoli.



La federazione risulta particolarmente utile nei casi seguenti:

- Gli utenti esistono già in una directory aziendale.

Se la directory aziendale è compatibile con Security Assertion Markup Language 2.0 (SAML 2.0), è possibile configurare la directory aziendale in modo da fornire l'accesso Single Sign-On (SSO) agli utenti. Console di gestione AWS Per ulteriori informazioni, consulta Scenari comuni per le credenziali temporanee.

Se la tua directory aziendale non è compatibile con SAML 2.0, puoi creare un'applicazione di identity broker per fornire l'accesso Single Sign-On (SSO) ai tuoi utenti. Console di gestione AWS Per ulteriori informazioni, consulta Abilita l'accesso personalizzato del broker di identità al AWS console.

Se la tua directory aziendale è Microsoft Active Directory, puoi utilizzarla Centro identità AWS IAM per connettere una directory autogestita in Active Directory o una directory in Servizio di directory AWS per stabilire la fiducia tra la tua directory aziendale e la tua. Account AWS

Se utilizzi un provider di identità (IdP) esterno come Okta o Microsoft Entra per gestire gli utenti, puoi utilizzarlo per Centro identità AWS IAM stabilire la fiducia tra il tuo IdP e il tuo. Account AWS Per ulteriori informazioni, consulta Connessione a un provider di identità esterno nella Guida per l'utente di Centro identità AWS IAM .

- I tuoi utenti dispongono di identità Internet.

  Se stai creando un'app mobile o un'applicazione Web che può consentire agli utenti di identificarsi tramite un provider di identità Internet, come Login with Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC), puoi decidere di utilizzare la federazione per accedere ad AWS. Per ulteriori informazioni, consulta Federazione OIDC.

  > **Tip**
  >
  > Per la federazione delle identità con i provider di identità Internet, ti consigliamo di utilizzare Amazon Cognito.

Diversi metodi per fornire l'accesso agli utenti

Ecco i modi in cui puoi fornire l'accesso alle tue risorse. AWS

Tipo di accesso utente	Quando viene utilizzato?	Dove si possono trovare ulteriori informazioni?
Accesso single sign-on per le persone, come gli utenti della forza lavoro, alle risorse AWS tramite il Centro identità IAM	IAM Identity Center fornisce un luogo centrale che riunisce l'amministrazione degli utenti e il loro accesso alle Account AWS applicazioni cloud. È possibile configurare un archivio di identità all'interno del Centro identità IAM oppure configurare la federazione con un gestore dell'identità digitale (IdP) esistente. Le migliori pratiche di sicurezza consigliano di concedere agli utenti umani credenziali limitate alle AWS risorse. Le persone hanno un'esperienza di accesso più semplice e tu mantieni il controllo sul loro accesso alle risorse da un unico sistema. Il Centro identità IAM supporta l'autenticazione a più fattori (MFA) per una maggiore sicurezza degli account.	Per ulteriori informazioni sulla configurazione del Centro identità IAM, consulta Nozioni di base nella Guida per l'utente di Centro identità AWS IAM . Per ulteriori informazioni sull'utilizzo dell'MFA in IAM Identity Center, consulta Multi-factor l'autenticazione nella Guida per l'utente Centro identità AWS IAM
Accesso federato per gli utenti umani, come gli utenti della forza	Supporti IAM IdPs compatibili con OpenID Connect (OIDC) o SAML 2.0 (Security Assertion Markup Language 2.0). Una volta creato un provider di identità IAM, sarà necessari	Per ulteriori informazioni sulla federazione e sui gestori di identità IAM, consulta Fornitori di identità e federazione in AWS.

Tipo di accesso utente	Quando viene utilizzato?	Dove si possono trovare ulteriori informazioni?
lavoro, ai AWS servizi che utilizzano i provider di identità IAM () IdPs	o creare uno o più ruoli IAM che possano essere assegnati dinamicamente a un principale federato.	
Cross-account accesso tra Account AWS	Vuoi condividere l'accesso a determinate AWS risorse con utenti in altre Account AWS. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, alcuni dei servizi AWS ti consentono di collegare una policy direttamente a una risorsa (invece di utilizzare un ruolo come proxy).	Per ulteriori informazioni sui ruoli IAM, consulta Ruoli IAM. Per ulteriori informazioni sui ruoli collegati al servizio, consulta Creare un ruolo collegato ai servizi. Per ulteriori informazioni sui servizi che supportano l'utilizzo di ruoli collegati ai servizi, consulta AWS servizi supportati da IAM. Trova i servizi che hanno Sì nella colonna Service-Linked Ruolo. Per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio, seleziona il link associato a Yes (Sì) nella colonna.

Tipo di accesso utente	Quando viene utilizzato?	Dove si possono trovare ulteriori informazioni?
Long-term le credenzia li per gli utenti IAM designati nel tuo Account AWS	Potresti avere casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM in. AWS Puoi usare IAM per creare questi utenti IAM nel tuo Account AWS e usare IAM per gestire le loro autorizzazioni. Alcuni dei casi d'uso sono i seguenti: • Carichi di lavoro che non possono utilizzare ruoli IAM • Third-party AWS client che richiedono l'accesso programmatico tramite chiavi di accesso • Service-specific credenzia li per Amazon AWS CodeCommit Keyspaces • Centro identità AWS IAM non è disponibile per il tuo account e non hai altri provider di identità Come best practice, negli scenari in cui sono necessari utenti IAM con accesso a livello di programmazione e credenziali a lungo termine, si consiglia di aggiornare le chiavi di accesso all'occor renza. Per ulteriori informazi	Per informazioni sulla configurazione di un utente IAM, consulta Crea un utente IAM nel tuo Account AWS. Per ulteriori informazioni sulle chiavi di accesso per gli utenti IAM, consulta Gestione delle chiavi di accesso per gli utenti IAM. Per ulteriori informazioni sulle credenziali specifiche del servizio per AWS CodeCommi t Amazon Keyspaces, consulta e. Credenzia li IAM per CodeCommit: credenziali Git, chiavi SSH e chiavi di accesso AWS Utilizzare IAM con Amazon Keyspaces (per Apache Cassandra)

Tipo di accesso utente	Quando viene utilizzato?	Dove si possono trovare ulteriori informazioni?
	oni, consulta Aggiornare le chiavi di accesso.	

Supportare l'accesso programmatico degli utenti

Gli utenti hanno bisogno di un accesso programmatico se vogliono interagire con persone esterne a. AWS Console di gestione AWS Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede: AWS

- Se gestisci le identità in IAM Identity Center, le AWS API richiedono un profilo e AWS Command Line Interface richiedono un profilo o una variabile di ambiente.
- Se hai utenti IAM, le AWS API e le chiavi di accesso AWS Command Line Interface richiedono. Quando possibile, creare credenziali temporanee formate da un ID della chiave di accesso, una chiave di accesso segreta e un token di sicurezza che ne indica la scadenza.

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Opzione	Ulteriori informazioni
Identità della forza lavoro (Persone e utenti gestiti nel Centro identità IAM)	Usa le credenziali a breve termine per firmare le richieste programmatiche alle AWS API AWS CLI o (direttamente o utilizzando gli SDK). AWS	Per farlo AWS CLI, segui le istruzioni in Ottenere le credenziali del ruolo IAM per l'accesso alla CLI nella Guida per l'utente. Centro identità AWS IAM Per le AWS API, segui le istruzioni nelle credenziali SSO nella Guida di riferimento per AWS SDK and Tools.

Quale utente necessita dell'accesso programmatico?	Opzione	Ulteriori informazioni
Utenti IAM	Usa le credenziali a breve termine per firmare le richieste programmatiche alle AWS API AWS CLI o (direttamente o utilizzando gli SDK). AWS	Segui le istruzioni in Utilizzo delle credenziali temporanee con le risorse. AWS
Utenti IAM	Usa le credenziali a lungo termine per firmare le richieste programmatiche alle AWS API AWS CLI o (direttamente o utilizzando gli SDK). AWS (Non consigliato)	Segui le istruzioni in Gestione delle chiavi di accesso per gli utenti IAM.
Principali federati	Utilizza un'operazione API AWS STS per creare una nuova sessione con credenziali di sicurezza temporanee che includono una coppia di chiavi di accesso e un token di sessione.	Per spiegazioni delle operazioni dell'API, consulta the section called "Richiedere credenziali di sicurezza temporanee"

Come le autorizzazioni e le policy forniscono la gestione degli accessi

La parte di gestione degli accessi di AWS Identity and Access Management (IAM) ti aiuta a definire cosa può fare un'entità principale in un account. Un'entità principale è una persona o un'applicazione autenticata tramite un'entità IAM (utente o ruolo IAM). La gestione degli accessi viene spesso definita come autorizzazione. Puoi gestire l'accesso AWS creando policy e collegandole a identità o risorse IAM (utenti IAM, gruppi IAM o ruoli IAM). AWS Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale utilizza un'entità IAM (utente IAM o ruolo IAM) per effettuare una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte

delle policy viene archiviata AWS come documenti JSON. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Politiche e autorizzazioni in AWS Identity and Access Management.

Policy e account

Se gestisci un singolo account in AWS, definisci le autorizzazioni all'interno di quell'account utilizzando le politiche. Se gestisci le autorizzazioni su più account, è più difficile gestire le autorizzazioni per i tuoi utenti IAM. Puoi utilizzare i ruoli IAM, le politiche basate sulle risorse o le liste di controllo degli accessi (ACLs) per le autorizzazioni tra account. Tuttavia, se possiedi più account, ti consigliamo invece di utilizzare il AWS Organizations servizio per aiutarti a gestire tali autorizzazioni. Per ulteriori informazioni, consulta Cos'è AWS Organizations? nella Guida AWS Organizations per l'utente.

Policy e utenti

Gli utenti IAM sono identità nell' Account AWS. Quando si crea un utente IAM, l'utente non potrà accedere ad alcun elemento nell'account finché non gli viene concessa l'autorizzazione. È possibile fornire autorizzazioni a un utente IAM creando una policy basata su identità, che è una policy collegata all'utente IAM o a un gruppo IAM a cui appartiene l'utente IAM. L'esempio seguente mostra una policy JSON che consente all'utente IAM di eseguire tutte le azioni di Amazon DynamoDB (`dynamodb:*`) sulla tabella `Books` nell'account `123456789012` all'interno della regione `us-east-2`.

JSON

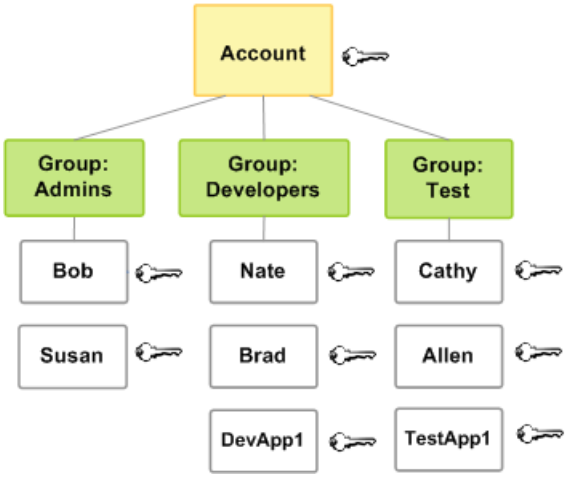
```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "dynamodb:*",
    "Resource": "arn:aws:dynamodb:us-east-2:123456789012:table/Books"
  }
}
```

Dopo il collegamento di questa policy all'utente IAM, l'utente disporrà dell'autorizzazione per eseguire tutte le operazioni nella tabella `Books` dell'istanza DynamoDB. La maggior parte degli utenti IAM dispone di più policy che si combinano per rappresentare il totale delle autorizzazioni concesse.

Le operazioni o le risorse che non sono esplicitamente consentite da una policy vengono rifiutate per impostazione predefinite. Ad esempio, se la policy precedente è la policy singola collegata a un utente, quell'utente può eseguire operazioni DynamoDB nella tabella `Books`, ma non può eseguire operazioni in altre tabelle. Allo stesso modo, all'utente non è consentito eseguire alcuna azione in Amazon EC2, Amazon S3 o in qualsiasi AWS altro servizio perché le autorizzazioni per lavorare con tali servizi non sono incluse nella politica.

Policy e gruppi IAM

Puoi organizzare gli utenti IAM in gruppi IAM e collegare una policy a un gruppo IAM. In quel caso, i singoli utenti IAM hanno ancora le proprie credenziali, ma tutti gli utenti IAM in un gruppo IAM dispongono delle autorizzazioni collegate al gruppo IAM. Utilizza i gruppi IAM per facilitare la gestione delle autorizzazioni.



Gli utenti o i gruppi IAM possono avere più policy a loro collegate, le quali concedono diverse autorizzazioni. In questo caso, la combinazione di policy determina le autorizzazioni effettive del principale. Se il principale non dispone dell'`Allow`autorizzazione esplicita sia per un'azione che per una risorsa, non dispone di tali autorizzazioni.

Sessioni e ruoli dell'utente federato

I presidi federati non hanno identità permanenti nel modo in cui li hanno gli utenti Account AWS IAM. Per assegnare le autorizzazioni ai principali federati, puoi creare un'entità definita come ruolo e stabilire le autorizzazioni per il ruolo. Quando un principale federato SAML o OIDC accede AWS, l'utente viene associato al ruolo e gli vengono concesse le autorizzazioni definite nel ruolo. Per ulteriori informazioni, consulta Creazione di un ruolo per un provider di identità di terze parti.

Policy basate su identità e policy basate su risorse.

Le policy basate su identità sono policy di autorizzazione che si collegano a un'identità IAM, come un utente, un gruppo o un ruolo IAM. Le policy basate su risorse sono policy di autorizzazione che si collegano a una risorsa, come un bucket Amazon S3 o una policy di attendibilità del ruolo IAM.

Le policy basate su identità controllano quali operazioni l'identità può eseguire, su quali risorse e in quali condizioni. Le policy basate su identità possono essere ulteriormente suddivise:

- Politiche gestite: politiche autonome basate sull'identità che puoi associare a più utenti, gruppi e ruoli nel tuo. Account AWS Puoi utilizzare due tipi di policy gestite:
  - AWS politiche gestite: politiche gestite create e gestite da. AWS Se non conosci l'utilizzo delle politiche, ti consigliamo di iniziare utilizzando le politiche AWS gestite.
  - Policy gestite dal cliente: le policy gestite che sono create e gestite nel tuo Account AWS. Le policy gestite dai clienti offrono un controllo più preciso sulle policy rispetto alle policy AWS gestite. Puoi creare, modificare e convalidare una policy IAM nell'editor visivo oppure creando direttamente il documento di policy JSON. Per ulteriori informazioni, consultare Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente e Modificare le policy IAM.
- Policy in linea: le policy che sono create, gestite e direttamente incorporate in un singolo utente, gruppo o ruolo. Nella maggior parte dei casi, non è consigliato l'uso di policy inline.

Le policy basate su risorse controllano quali operazioni uno specifico principale può eseguire, su quale risorsa e in quali condizioni. Le policy basate risorse sono policy inline. Non esistono policy gestite basate su risorse. Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse.

Il servizio IAM supporta solo un tipo di policy basata su risorse detta policy di attendibilità del ruolo, collegata a un ruolo IAM. Poiché un ruolo IAM è sia un'identità che una risorsa che supporta policy basate su risorse, a un ruolo IAM è necessario collegare sia una policy di attendibilità che una policy basata su identità. Le politiche di fiducia definiscono quali entità principali (account, utenti, ruoli e responsabili degli utenti AWS STS federati) possono assumere il ruolo. Per capire in che modo i ruoli IAM si differenziano da altre policy basate su risorse, consulta Accesso alle risorse multi-account in IAM.

Per scoprire quali servizi supportano le policy basate su risorse, consulta la pagina AWS servizi supportati da IAM. Per ulteriori informazioni sulle policy basate su risorse, consulta la pagina Identity-based politiche e politiche basate sulle risorse.

Definire le autorizzazioni basate su attributi con l'autorizzazione ABAC

Attribute-based il controllo degli accessi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. AWS chiama questi attributi tag. Puoi allegare tag alle risorse IAM, incluse le entità IAM (utenti IAM o ruoli IAM) e alle AWS risorse. È possibile creare una singola policy ABAC o un piccolo insieme di policy per i principali IAM. Queste policy ABAC possono essere definite affinché autorizzino le operazioni quando il tag dell'entità corrisponde al tag della risorsa. Il sistema di attributi di ABAC che fornisce sia un contesto utente elevato che un controllo degli accessi granulare. Poiché ABAC è basato sugli attributi, può eseguire autorizzazioni dinamiche per dati o applicazioni che concedono o revocano l'accesso in tempo reale. La strategia ABAC è utile in ambienti soggetti a una rapida scalabilità e in situazioni in cui la gestione delle policy di identità o delle risorse è diventata complessa.

Ad esempio, è possibile creare tre ruoli IAM con la chiave di tag `access-project`. Impostare il valore del tag del primo ruolo IAM su `Heart`, del secondo su `Star` e del terzo su `Lightning`. Puoi quindi utilizzare un'unica policy che consenta l'accesso quando il ruolo IAM e la AWS risorsa hanno il valore del tag`access-project`. Per un tutorial dettagliato che illustra come utilizzare ABAC in AWS, consulta Tutorial IAM: Definizione delle autorizzazioni di accesso AWS risorse basate sui tag. Per ulteriori informazioni sui servizi a supporto di ABAC, consulta AWS servizi supportati da IAM. Questo diagramma illustra che i tag applicati a un principale devono corrispondere ai tag applicati a una risorsa affinché all'utente vengano concesse le autorizzazioni per l'uso della risorsa. I tag devono essere applicati a gruppi IAM, gruppi di risorse, singoli utenti e singole risorse.

Confronto di ABAC con il modello RBAC tradizionale

Il modello di autorizzazione tradizionale utilizzato in IAM è chiamato controllo degli accessi basato su ruoli (RBAC). RBAC definisce le autorizzazioni in base alle mansioni lavorative o al ruolo di una persona, che è diverso da un ruolo IAM. IAM include policy gestite per le funzioni processo che allineano le autorizzazioni a una funzione di processo in un modello RBAC.

In IAM, è possibile implementare RBAC creando diverse policy per diverse mansioni lavorative. Quindi è possibile collegare le policy alle identità (utenti, gruppi o ruoli IAM). Come best practice si suggerisce di concedere le autorizzazioni minime necessarie per la mansione lavorativa. Ciò si traduce in un accesso con privilegi minimi. Ogni policy relativa alla funzione lavorativa elenca le risorse specifiche a cui possono accedere le identità assegnate a tale policy. Lo svantaggio di

utilizzare il modello RBAC tradizionale è che, nel momento in cui gli utenti aggiungono nuove risorse, per consentire l'accesso a esse è necessario aggiornare le policy.

Ad esempio, si supponga di disporre di tre progetti denominati `Heart`, `Star` e `Lightning`, su cui lavorano i dipendenti. Si crea un ruolo IAM per ogni progetto. È quindi possibile collegare le policy a ciascun ruolo IAM per definire le risorse a cui può accedere chiunque sia autorizzato ad assumere il ruolo IAM. Se un dipendente cambia mansione all'interno dell'azienda, è necessario assegnargli un ruolo IAM differente. Le persone o i programmi possono essere assegnati a più di un ruolo IAM. Tuttavia, il progetto `Star` potrebbe richiedere risorse aggiuntive, ad esempio un nuovo container Amazon EC2. In tal caso, è necessario aggiornare la policy collegata al ruolo `Star` IAM per specificare la nuova risorsa del container. In caso contrario, i membri del progetto `Star` non potranno accedere al nuovo container.
Questo diagramma illustra che il controllo degli accessi basato sui ruoli richiede che a ciascuna identità venga assegnata una policy specifica basata sulla funzione lavorativa per accedere a risorse diverse.

Il modello ABAC offre i seguenti vantaggi rispetto al modello RBAC tradizionale:

- Le autorizzazioni ABAC si ridimensionano con l'innovazione. Non è più necessario che un amministratore aggiorni le policy esistenti per consentire l'accesso a nuove risorse. Ad esempio, si supponga di aver progettato la strategia ABAC con il tag `access-project`. Uno sviluppatore utilizza il ruolo IAM con il tag `access-project` = `Heart`. Quando le persone del progetto `Heart` hanno bisogno di risorse Amazon EC2 aggiuntive, lo sviluppatore può creare nuove istanze Amazon EC2 con il tag `access-project` = `Heart`. In questo modo chiunque partecipi al progetto `Heart` può avviare e arrestare tali istanze perché i rispettivi valori di tag corrispondono.
- ABAC richiede meno policy. Poiché non è necessario creare policy diverse per diverse mansioni lavorative, è necessario creare meno policy. Tali policy sono più facili da gestire.
- Utilizzando ABAC, i team possono rispondere in modo dinamico ai cambiamenti e alla crescita. Poiché le autorizzazioni per le nuove risorse vengono concesse automaticamente in base agli attributi, non è necessario assegnare manualmente le policy alle identità. Ad esempio, se la propria azienda supporta già i progetti `Heart` e `Star` utilizzando ABAC, è facile aggiungere un nuovo progetto `Lightning`. Un amministratore IAM crea un nuovo ruolo con il tag `access-project` = `Lightning`. Non è necessario modificare la policy per supportare un nuovo progetto. Chiunque disponga delle autorizzazioni per assumere il ruolo IAM può creare e visualizzare istanze a cui è stato assegnato il tag `access-project` = `Lightning`. Un altro scenario si verifica quando un membro del team passa dal progetto `Heart` al progetto `Lightning`. Per fornire ai membri del

team l'accesso al progetto `Lightning`, l'amministratore IAM li assegna a un ruolo IAM diverso. Non è necessario modificare le policy di autorizzazione.

- Utilizzando la strategia ABAC e possibile definire autorizzazioni con un maggior livello di granularità. Quando si creano le policy, è consigliabile concedere i privilegi minimi. Utilizzando l'approccio RBAC tradizionale, è necessario scrivere una policy che consenta l'accesso a specifiche risorse. Tuttavia, quando si utilizza ABAC, è possibile consentire operazioni su tutte le risorse, ma solo se il tag della risorsa corrisponde al tag del principale.
- Con ABAC è possibile utilizzare gli attributi dei dipendenti memorizzati nella directory aziendale. È possibile configurare il provider di identità SAML o OIDC per passare i tag di sessione a IAM. Quando i dipendenti si uniscono AWS, IAM applica i loro attributi al responsabile risultante. È quindi possibile utilizzare ABAC per consentire o negare le autorizzazioni sulla base di tali attributi.

Per un tutorial dettagliato che dimostra come utilizzare ABAC in AWS, vedi. Tutorial IAM: Definizione delle autorizzazioni di accesso AWS risorse basate sui tag

Nozioni di base su IAM

AWS Identity and Access Management (IAM) ti aiuta a controllare in modo sicuro l'accesso ad Amazon Web Services (AWS) e alle risorse del tuo account. IAM può anche mantenere le credenziali di accesso private. Non è necessario registrarsi specificamente per utilizzare IAM. L'uso di IAM non comporta alcun costo.

Il modo in cui utilizzi IAM dipende dalla tua funzione lavorativa. Consulta Quando si usa IAM? la sezione per una guida specifica per il ruolo.

Utilizza IAM per concedere a identità, ad esempio ruoli e ruoli, l'accesso alle risorse nell'account. Ad esempio, puoi utilizzare IAM con utenti esistenti nella tua directory aziendale che gestisci esternamente AWS oppure puoi creare utenti in AWS uso. Centro identità AWS IAM Le identità federate assumono ruoli IAM definiti per accedere alle risorse di cui hanno bisogno. Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di Centro identità AWS IAM .

> **Note**
>
> IAM è integrato con diversi AWS prodotti. Per un elenco di servizi che supportano IAM, consulta AWS servizi supportati da IAM.

Per informazioni su come iniziare AWS, creare un utente amministrativo e utilizzare più servizi per risolvere un problema AWS Organizations, ad esempio la creazione e il lancio del primo progetto, consulta il Centro risorse per iniziare.

Per esercitazioni pratiche con le attività IAM più comuni, consulta. Tutorial IAM Le attività includono la creazione di ruoli per l'accesso tra più account e l'utilizzo di tag per il controllo degli accessi basato sugli attributi.

Dopo aver configurato l'account, pianifica in che modo persone e carichi di lavoro vi accedono. Per ulteriori informazioni, consulta Pianifica l'accesso al tuo AWS account.

Quindi usalo Preparazione per le autorizzazioni con privilegi minimi per seguire il principio del privilegio minimo.

Configurare il tuo Account AWS

Prima di iniziare a lavorare con IAM, assicurati di aver completato la configurazione iniziale del tuo AWS ambiente.

AWS ti invia un'email di conferma al termine della procedura di registrazione. Puoi visualizzare l'attività corrente del tuo account e gestire l'account in qualsiasi momento accedendo a https://aws.amazon.com/ e scegliendo My Account (Il mio account).

Quando ti sei registrato al servizio, hai creato un Account AWS utilizzando un indirizzo email e una password. Queste sono le credenziali dell'utente AWS root. Come best practice, non si utilizzano le credenziali dell'utente root AWS per accedere alle attività quotidiane. Utilizza le credenziali dell'utente root solo per le attività che richiedono le credenziali dell'utente root. Inoltre, non condividere le credenziali con nessun altro. Invece, aggiungi persone alla directory e consenti loro di accedere al tuo Account AWS.

Per proteggere i tuoi Utente root dell'account AWS

1. Accedi Console di gestione AWS come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Accedere come utente root nella Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella IAM User Guide.

Concedere l'accesso alla console di fatturazione

Per impostazione predefinita, gli utenti e ruoli IAM in un Account AWS non possono accedere alla console Gestione costi e fatturazione. Ciò vale anche se dispongono di policy IAM che concedono l'accesso a determinate funzionalità di fatturazione. Per concedere l'accesso, l'utente root Account AWS deve prima attivare l'accesso IAM.

> **Note**
>
> Come best practice di sicurezza, consigliamo di fornire l'accesso alle risorse tramite la federazione delle identità con Centro identità AWS IAM. Quando abiliti IAM Identity

Center insieme a AWS Organizations, la console di fatturazione e gestione dei costi è abilitata per impostazione predefinita con una fatturazione consolidata per tutti i membri dell'organizzazione. Account AWS Per ulteriori informazioni, consulta Consolidamento della fatturazione per AWS Organizations nella Guida per l'utente di Gestione costi e fatturazione.

1. Accedi Console di gestione AWS con le tue credenziali di utente root (in particolare, l'indirizzo email e la password che hai usato per creare il tuo account). AWS
2. Nella barra di navigazione seleziona il tuo nome account, quindi scegli Account.
3. Scorri la pagina verso il basso fino a trovare la sezione Accesso utente e ruolo IAM alle informazioni di fatturazione, quindi seleziona Modifica.
4. Seleziona la casella di controllo Activate IAM Access (Attiva l'accesso IAM) per attivare l'accesso alle pagine della console di Gestione costi e fatturazione.
5. Scegli Update (Aggiorna).

   La pagina mostra il messaggio che l' user/role accesso IAM alle informazioni di fatturazione è attivato.

   > **Important**
   >
   > L'attivazione dell'accesso IAM da sola non concede agli utenti e ai ruoli IAM le autorizzazioni necessarie per queste pagine della console Gestione costi e fatturazione. È inoltre necessario collegare le policy basate sull'identità richieste ai ruoli IAM per concedere l'accesso alla console di fatturazione. I ruoli forniscono credenziali temporanee che gli utenti possono assumere quando necessario.

6. Usa il Console di gestione AWS per creare un ruolo che un utente può assumere per accedere alla console di fatturazione.
7. Nella pagina Aggiungi autorizzazioni per il ruolo, aggiungi le autorizzazioni all'elenco e visualizza i dettagli sulle risorse di fatturazione disponibili nel tuo Account AWS.

   La politica AWS gestita Billing concede agli utenti il permesso di visualizzare e modificare la console di fatturazione e gestione dei costi. Ciò include la visualizzazione dell'utilizzo dell'account, la modifica dei budget e dei metodi di pagamento. Per altri esempi di policy che puoi collegare ai ruoli IAM per controllare l'accesso alle informazioni di fatturazione del tuo account, consulta Esempi di policy di fatturazione AWS nella Guida per l'utente di Gestione costi e fatturazione.

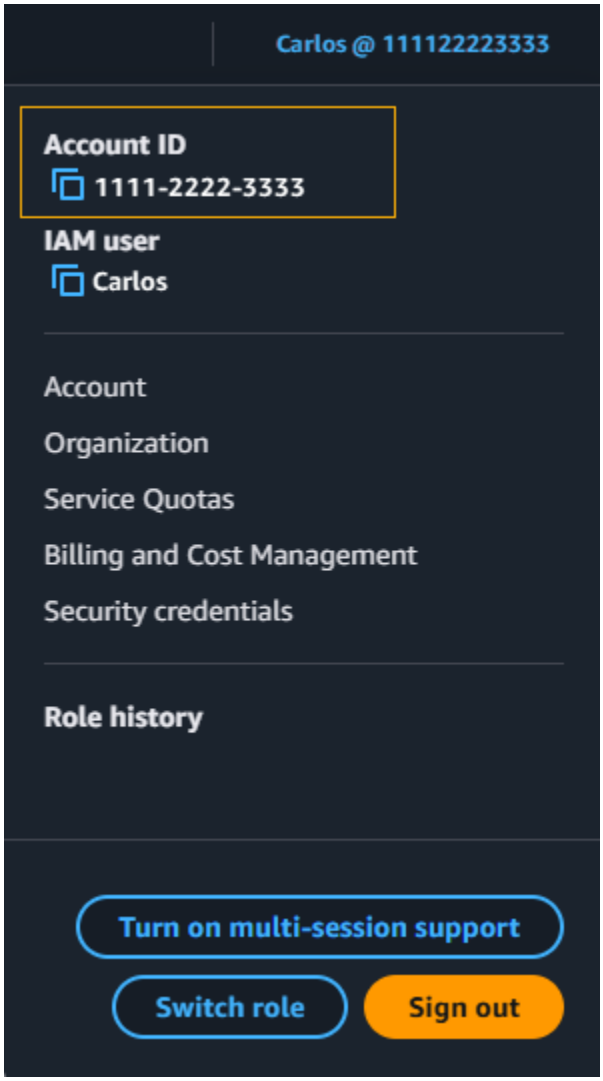
Visualizzazione del tuo Account AWS ID

Se hai effettuato l'accesso alla console, puoi visualizzare l'ID account del tuo Account AWS utilizzando i seguenti metodi.

Per visualizzare il tuo Account AWS ID

Console

L'ID AWS dell'account viene visualizzato quando si accede alla dashboard IAM nella Account AWS sezione. Puoi anche visualizzare l'ID account nella barra di navigazione in alto a destra. Scegli il nome utente e l'ID account sarà visualizzato sopra di esso.



AWS CLI

Utilizza il comando seguente per visualizzare l'ID utente, l'ID account e l'ARN utente:

- aws sts get-caller-identity

API

Utilizza la seguente API per visualizzare l'ID utente, l'ID account e l'ARN utente:

- GetCallerIdentity

Usare un alias per il tuo Account AWS ID

L'ID account è un numero di 12 cifre che identifica in modo univoco l'account. Per impostazione predefinita, gli utenti IAM dell'account accedono utilizzando un URL Web che include l'ID account. Se non dispongono dell'URL, possono fornire l'ID dell'account nella pagina di accesso al momento dell'accesso. AWS

Per impostazione predefinita, l'URL della pagina di accesso ha il formato seguente.

```
https://Your_Account_ID.signin.aws.amazon.com/console/
```

Molte persone trovano che le parole siano più facili da ricordare rispetto ai numeri, quindi la creazione di un alias per l'ID account può aiutare gli utenti IAM ad accedere più facilmente.

Se crei un Account AWS alias per il tuo Account AWS ID, l'URL della pagina di accesso è simile all'esempio seguente.

```
https://Your_Account_Alias.signin.aws.amazon.com/console/
```

Considerazioni prima di creare un alias di un account

- Account AWS Puoi avere un solo alias. Se crei un nuovo alias per il tuo AWS account, il nuovo alias sovrascrive l'alias precedente e l'URL contenente l'alias precedente smette di funzionare.
- L'alias dell' account deve includere solo cifre, lettere minuscole e trattini. Per ulteriori informazioni sulle limitazioni relative alle entità dell'account, consulta. AWS IAM e AWS STS quote
- L'alias dell'account deve essere univoco nei prodotti Amazon Web Services nella partizione di una determinata rete.

  Una partizione è un gruppo di regioni AWS . Ogni AWS account è limitato a una partizione.

Di seguito sono riportate le partizioni supportate:

- `aws`- Regioni AWS
- `aws-cn`: regioni Cina
- `aws-us-gov` – Regioni AWS GovCloud (US)

> **Note**
>
> Gli alias degli account non sono segreti e verranno visualizzati nell'URL della tua pagina di accesso pubblica. Non includere informazioni sensibili nell'alias del tuo account.
> L'URL originale contenente il tuo Account AWS ID rimane attivo e può essere utilizzato dopo aver creato il tuo Account AWS alias.

Per istruzioni sulla configurazione o la rimozione di un alias, consulta Creazione di un alias dell'account e. Eliminazione di un alias dell'account

Creazione di un alias dell'account

Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:

- `iam:ListAccountAliases`
- `iam:CreateAccountAlias`

Per creare un Account AWS alias

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione seleziona Pannello di controllo.
3. Nella sezione Account AWS , accanto ad Alias dell'account, scegli Crea. Se l'alias esiste già, scegli Modifica.
4. Nella finestra di dialogo, inserisci il nome che desideri utilizzare per l'alias e quindi seleziona Salva modifiche.

AWS CLI

Esegui il comando seguente:

- `aws iam create-account-alias`

API

Per creare un alias per l'URL della pagina di accesso della Console di gestione AWS , chiama l'operazione seguente:

- `CreateAccountAlias`

Eliminazione di un alias dell'account

Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:

- `iam:ListAccountAliases`
- `iam:DeleteAccountAlias`

Come eliminare l'alias di un account

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione seleziona Pannello di controllo.
3. Nella sezione Account AWS , accanto ad Alias dell'account, scegli Elimina.

AWS CLI

Per eliminare un alias Account AWS ID, esegui il seguente comando:

- `aws iam delete-account-alias`

Per confermare che l'alias dell'account è stato eliminato, prova a visualizzare l'alias Account AWS dell'ID eseguendo il comando seguente:

- `aws iam list-account-aliases`

API

Per eliminare un alias Account AWS ID, chiamate la seguente operazione:

- `DeleteAccountAlias`

Per confermare che l'alias dell'account è stato eliminato, prova a visualizzare l'alias dell' Account AWS ID eseguendo la seguente operazione:

- `ListAccountAliases`

> **Note**
>
> Dopo aver eliminato l'alias dell'account, l'unico URL di accesso per l'account si baserà sull'ID account. Qualsiasi tentativo di connessione all'URL dell'alias non riuscirà e non sarà reindirizzato.

Pianifica l'accesso al tuo AWS account

Durante la configurazione AWS, pianifica in che modo intendi che le persone accedano al tuo AWS account e alle tue risorse. Una soluzione di gestione delle identità ben progettata protegge il tuo account.

Origini di identità

Secondo le best practice IAM, consigliamo agli utenti umani e ai carichi di lavoro di utilizzare credenziali temporanee per accedere alle risorse. AWS Concedi credenziali temporanee tramite i ruoli IAM. Sia gli utenti federati in IAM che gli utenti nel Centro identità IAM (federati o creati nella directory del Centro identità IAM) utilizzano i ruoli IAM per accedere alle risorse.

Prima di iniziare a utilizzare AWS, pianifica come configurare le tue identità utilizzando uno dei seguenti approcci:

- Abilitazione di IAM Identity Center AWS Organizations e aggiunta di utenti in IAM Identity Center direttamente alla directory organizzativa.

Per istruzioni su come aggiungere utenti direttamente alla directory organizzativa di IAM Identity Center, consulta Aggiungere utenti

- Federazione del tuo provider di identità esterno esistente con il Centro identità IAM o IAM.

  Per istruzioni su come federare un provider di identità esterno alla directory organizzativa di IAM Identity Center, consulta l'apposito tutorial https://docs.aws.amazon.com/singlesignon/latest/userguide/tutorials.html introduttivo.

Gestione degli accessi

Identifica le AWS risorse e i servizi di cui i tuoi utenti hanno bisogno. Quindi definisci le autorizzazioni e le politiche di accesso per ogni utente, gruppo o ruolo.

- Se utilizzi IAM Identity Center, un provider di identità IAM, i ruoli IAM e le politiche di autorizzazione vengono creati in ogni AWS account della tua organizzazione. Questi ruoli e autorizzazioni corrispondono a quanto specificato quando assegni persone o gruppi ad applicazioni o AWS account.

  Per ulteriori informazioni, consulta Assegnare l'accesso utente e Configurare l'accesso Single Sign-On alle applicazioni.
- Puoi utilizzare l'account access manager , una funzionalità IAM che ti consente di assegnare i ruoli IAM esistenti che crei e gestisci agli utenti e ai gruppi di IAM Identity Center. Account Access Manager ti dà accesso al set completo di funzionalità dei ruoli IAM. Puoi usarlo insieme ai set di autorizzazioni o da solo.
- Se federate il vostro provider di identità direttamente con IAM nel vostro Account AWS, dovete creare un ruolo che gli utenti possano assumere. Il ruolo richiede una politica di fiducia e una politica di autorizzazioni. La politica di fiducia stabilisce chi può assumere il ruolo. La politica delle autorizzazioni indica a quali AWS azioni e risorse è consentito o negato l'accesso al ruolo.

  Per ulteriori informazioni, consulta Fornitori di identità e federazione in AWS

Per istruzioni su come aggiungere un ulteriore livello di sicurezza per l'accesso, consulta. Utilizzo dell'autenticazione a più fattori con le identità

Casi d'uso per utenti IAM

Gli utenti IAM che crei nel tuo Account AWS hanno credenziali a lungo termine che gestisci direttamente.

Quando si tratta di gestire l'accesso in AWS, gli utenti IAM non sono generalmente la scelta migliore. Esistono alcuni motivi principali per cui dovresti evitare di affidarti agli utenti IAM per la maggior parte dei casi d'uso.

Innanzitutto, gli utenti IAM sono progettati per account individuali, quindi non si adattano bene alla crescita dell'organizzazione. La gestione delle autorizzazioni e della sicurezza per un gran numero di utenti IAM può rapidamente diventare un problema.

Gli utenti IAM non dispongono inoltre della visibilità centralizzata e delle funzionalità di controllo che si ottengono con altre soluzioni di gestione delle AWS identità. Ciò può rendere più difficile mantenere la sicurezza e la conformità normativa.

Infine, l'implementazione delle best practice di sicurezza come l'autenticazione a più fattori, le policy di password e la separazione dei ruoli è molto più semplice con approcci di gestione delle identità più scalabili.

Invece di affidarti agli utenti IAM, consigliamo di utilizzare soluzioni più solide come IAM Identity Center con AWS Organizations o identità federate di provider esterni. Queste opzioni ti offriranno un migliore controllo, sicurezza ed efficienza operativa man mano che il tuo AWS ambiente cresce.

Di conseguenza, ti consigliamo di utilizzare gli utenti IAM solo per casi d'uso non supportati dagli utenti federati.

L'elenco seguente identifica i casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM in. AWS Puoi usare IAM per creare questi utenti IAM sotto l'ombrello del tuo AWS account e usare IAM per gestire le loro autorizzazioni.

- Accesso di emergenza al tuo account AWS
- Carichi di lavoro che non possono utilizzare ruoli IAM
  - AWS CodeCommit accesso
  - Accesso ad Amazon Keyspaces (per Apache Cassandra)
- Third-party AWS clienti
- Centro identità AWS IAM non è disponibile per il tuo account e non hai altri provider di identità

Creare un utente IAM per l'accesso di emergenza

Un utente IAM è un'identità all'interno dell'utente Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione.

Avere un utente IAM per l'accesso di emergenza è uno dei motivi consigliati per creare un utente IAM in modo da poter accedere al tuo Account AWS se il tuo provider di identità non è accessibile.

> **Note**
>
> Come best practice di sicurezza, consigliamo di fornire l'accesso alle risorse tramite la federazione delle identità invece di creare utenti IAM. Per informazioni su situazioni specifiche in cui è richiesto un utente IAM, consulta la sezione Quando creare un utente IAM invece di un ruolo.

Per creare un utente IAM per l'accesso di emergenza

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - `access-analyzer:ValidatePolicy`
> - `iam:AddUserToGroup`
> - `iam:AttachGroupPolicy`
> - `iam:CreateGroup`
> - `iam:CreateLoginProfile`
> - `iam:CreateUser`
> - `iam:GetAccountPasswordPolicy`
> - `iam:GetLoginProfile`
> - `iam:GetUser`
> - `iam:ListAttachedGroupPolicies`
> - `iam:ListAttachedUserPolicies`
> - `iam:ListGroupPolicies`

- `iam:ListGroups`
- `iam:ListGroupsForUser`
- `iam:ListPolicies`
- `iam:ListUserPolicies`
- `iam:ListUsers`

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere a nella Guida per AWS l'AWS Sign-In utente.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel riquadro di navigazione, seleziona Utenti, quindi seleziona Crea utente.

   > **Note**
   >
   > Se hai abilitato IAM Identity Center, Console di gestione AWS viene visualizzato un promemoria che ti ricorda che è meglio gestire l'accesso degli utenti in IAM Identity Center. In questa procedura, l'utente IAM che viene creato è destinato specificamente all'uso solo se non è disponibile il provider di identità.

4. Nella pagina Specify user details (Specifica dettagli utente), in User details (Dettagli utente), in User name (Nome utente), immetti il nome del nuovo utente. Questo è il nome di accesso per AWS. In questo esempio, inserisci **`EmergencyAccess`**.

   > **Note**
   >
   > I nomi utente possono essere una combinazione di un massimo di 64 lettere, cifre e i seguenti caratteri: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), trattino basso (_) e trattino (-). I nomi devono essere univoci nell'account. Non sono distinti per caso. Ad esempio, non è possibile creare due utenti chiamati TESTUSER e testuser. Quando un nome utente viene utilizzato in una policy o come parte di un ARN, il nome fa distinzione tra maiuscole e minuscole. Quando un nome utente viene visualizzato ai clienti nella console, ad esempio durante il processo di accesso, il nome utente non fa distinzione tra maiuscole e minuscole.

5. Seleziona la casella di controllo accanto a Fornisci l'accesso utente alla Console di gestione AWS- facoltativo, quindi scegli Desidero creare un utente IAM.
6. Per Console password (Password della console), seleziona Autogenerated password (Password generata automaticamente).
7. Seleziona la casella di controllo accanto a L'utente deve creare una nuova password all'accesso successivo (consigliato). Poiché questo utente IAM è destinato all'accesso di emergenza, un amministratore attendibile ne conserverà la password e la fornirà solo quando necessario.
8. Nella pagina Set permissions (Imposta autorizzazioni), in Permissions options (Opzioni di autorizzazione), seleziona Add user to group (Aggiungi utente al gruppo). Quindi, in User groups (Gruppi di utenti), seleziona Create group (Crea gruppo).
9. Nella pagina Create user group (Crea gruppo di utenti), in User group name (Nome gruppo di utenti), inserisci **`EmergencyAccessGroup`**. Quindi, in Politiche di autorizzazione, seleziona AdministratorAccess.
10. Scegli Crea gruppo di utenti per tornare alla pagina Imposta autorizzazioni.
11. In User groups (Gruppi di utenti), seleziona il nome del **`EmergencyAccessGroup`** creato in precedenza.
12. Seleziona Successivo per passare alla pagina Rivedi e crea.
13. Nella pagina Review and create (Rivedi e crea), consulta l'elenco dei membri del gruppo di utenti da aggiungere al nuovo utente. Una volta pronto per continuare, seleziona Create user (Crea utente).
14. Nella pagina Recupera password, seleziona Scarica il file .csv per salvare un file .csv con le informazioni sulle credenziali dell'utente (URL di connessione, nome utente e password).
15. Salva questo file per utilizzarlo se devi accedere a IAM e non hai accesso al tuo provider di identità.

Il nuovo utente IAM viene visualizzato nell'elenco Users (Utenti). Seleziona il link User name (Nome utente) per visualizzare i dettagli dell'utente.

AWS CLI

1. Crea un utente denominato **`EmergencyAccess`**.

   - aws iam create-user

```
aws iam create-user \
   --user-name EmergencyAccess
```

2. (Facoltativo) Concedere all'utente l'accesso alla Console di gestione AWS. Ciò richiede una password. Per creare una password per un utente IAM puoi utilizzare il parametro `--cli-input-json` per passare un file JSON contenente la password. Devi inoltre fornire all'utente l'URL della pagina di accesso del tuo account.

   - aws iam create-login-profile

   ```
   aws iam create-login-profile \
      --generate-cli-skeleton > create-login-profile.json
   ```

   - Apri il file `create-login-profile.json` in un editor di testo e inserisci una password conforme alla tua policy delle password, quindi salva il file. Ad esempio:

   ```
   {
    "UserName": "EmergencyAccess",
    "Password": "Ex@3dRA0djs",
    "PasswordResetRequired": false
   }
   ```

   - Usa nuovamente il comando `aws iam create-login-profile`, passando il parametro `--cli-input-json` per specificare il tuo file JSON.

   ```
   aws iam create-login-profile \
      --cli-input-json file://create-login-profile.json
   ```

   > **Note**
   >
   > Se la password che hai fornito nel file JSON viola la policy delle password del tuo account, riceverai un errore `PasswordPolicyViolation`. In tal caso, rivedi la

policy delle password per il tuo account e aggiorna la password nel file JSON per soddisfare i requisiti.

3. Crea **`EmergencyAccessGroup`**, allega la politica AWS gestita `AdministratorAccess` al gruppo e aggiungi l'**`EmergencyAccess`**utente al gruppo.

> **Note**
>
> Una policy gestita da AWS è una policy autonoma che viene creata e amministrata da AWS. Ogni policy ha il proprio nome della risorsa Amazon (ARN) che include il nome della policy. Ad esempio, `arn:aws:iam::aws:policy/IAMReadOnlyAccess` è una policy AWS gestita. Per ulteriori informazioni sugli ARN, consultare la pagina ARN IAM. Per un elenco delle politiche AWS gestite per Servizi AWS, vedere le politiche AWS gestite.

- aws iam create-group

```
aws iam create-group \
   --group-name EmergencyAccessGroup
```

- aws iam attach-group-policy

```
aws iam attach-group-policy \
   --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \
   --group-name >EmergencyAccessGroup
```

- aws iam add-user-to-group

```
aws iam add-user-to-group \
   --user-name EmergencyAccess \
   --group-name EmergencyAccessGroup
```

- Esegui il comando aws iam get-group per elencare **`EmergencyAccessGroup`** e i relativi membri.

```
aws iam get-group \
   --group-name EmergencyAccessGroup
```

Creare un utente IAM per carichi di lavoro che non possono utilizzare i ruoli IAM

> **Important**
>
> Come best practice, ti consigliamo di richiedere agli utenti umani di utilizzare credenziali temporanee per l'accesso. AWS
> In alternativa, puoi gestire le tue identità utente, incluso l'utente amministrativo, con Centro identità AWS IAM. Consigliamo di utilizzare Centro identità IAM per gestire l'accesso ai tuoi account e le autorizzazioni all'interno di questi account. Se utilizzi un provider di identità esterno, puoi configurare le autorizzazioni di accesso per le identità degli utenti nel Centro identità IAM.

Se il tuo caso d'uso richiede utenti IAM con accesso programmatico e credenziali a lungo termine, si consiglia di stabilire procedure per aggiornare le chiavi di accesso all'occorrenza. Per ulteriori informazioni, consulta Aggiornare le chiavi di accesso.

Per eseguire alcune attività di gestione di account e servizi, è necessario effettuare l'accesso utilizzando le credenziali dell'utente root. Per visualizzare le attività che richiedono l'accesso come utente root, consulta Attività che richiedono credenziali dell'utente root.

Per creare un utente IAM per carichi di lavoro che non possono utilizzare i ruoli IAM

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - `iam:AddUserToGroup`
> - `iam:AttachGroupPolicy`
> - `iam:CreateAccessKey`
> - `iam:CreateGroup`

- iam:CreateServiceSpecificCredential
- iam:CreateUser
- iam:GetAccessKeyLastUsed
- iam:GetAccountPasswordPolicy
- iam:GetAccountSummary
- iam:GetGroup
- iam:GetLoginProfile
- iam:GetPolicy
- iam:GetRole
- iam:GetUser
- iam:ListAccessKeys
- iam:ListAttachedGroupPolicies
- iam:ListAttachedUserPolicies
- iam:ListGroupPolicies
- iam:ListGroups
- iam:ListGroupsForUser
- iam:ListInstanceProfilesForRole
- iam:ListMFADevices
- iam:ListPolicies
- iam:ListRoles
- iam:ListRoleTags
- iam:ListSSHPublicKeys
- iam:ListServiceSpecificCredentials
- iam:ListSigningCertificates
- iam:ListUserPolicies
- iam:ListUserTags
- iam:ListUsers
- iam:UploadSSHPublicKey
- iam:UploadSigningCertificate

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere a nella Guida per AWS l'AWS Sign-In utente.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel riquadro di navigazione, scegli Utenti, quindi scegli Crea utente.
4. Nella pagina Specifica dettagli utente, procedi come segue:
   a. Per Nome utente, digita ***`WorkloadName`***. Sostituisci ***`WorkloadName`*** con il nome del carico di lavoro che utilizzerà l'account.
   b. Scegli Next (Successivo).
5. (Facoltativo) Nella pagina Imposta autorizzazioni, procedi nel modo seguente:
   a. Scegli Add user to group (Aggiungi utente al gruppo).
   b. Seleziona Crea gruppo.
   c. Nella finestra di dialogo Crea gruppo di utenti, in Nome gruppo di utenti, inserisci un nome che rappresenti l'uso dei carichi di lavoro nel gruppo. Per questo esempio, usa il nome **`Automation`**.
   d. In Criteri di autorizzazione seleziona la casella di controllo per la politica PowerUserAccess gestita.

      > **Tip**
      >
      > Inserisci Power nella casella di ricerca Policy di autorizzazione per trovare rapidamente la policy gestita.

   e. Scegli Create user group (Crea gruppo di utenti).
   f. Di nuovo nella pagina con l'elenco dei gruppi IAM, seleziona la casella di controllo per il nuovo gruppo di utenti. Scegli Aggiorna se il nuovo gruppo di utenti non viene visualizzato nell'elenco.
   g. Scegli Next (Successivo).
6. (Facoltativo) Nella sezione Tag aggiungi i metadati all'utente collegando i tag come coppie chiave-valore. Per ulteriori informazioni, consulta Tag per AWS Identity and Access Management risorse.

7. Verifica le appartenenze ai gruppi di utenti per il nuovo utente. Quando sei pronto per continuare, seleziona Crea utente.
8. Viene visualizzata una notifica di stato che informa che l'utente è stato creato correttamente. Seleziona Visualizza utente per accedere alla pagina dei dettagli dell'utente.
9. Seleziona la scheda Credenziali di sicurezza. Crea quindi le credenziali necessarie per il carico di lavoro.

   - Chiavi di accesso: seleziona Crea chiave di accesso per generare e scaricare le chiavi di accesso per l'utente.

     > **Important**
     >
     > Questa è la tua unica opportunità per visualizzare o scaricare le chiavi di accesso segrete e devi fornire queste informazioni ai tuoi utenti prima che possano utilizzare l' AWS API. Salva i nuovi ID chiave di accesso e Secret Access Key dell'utente in un luogo sicuro. Successivamente a questa fase non sarà più possibile accedere alle chiavi segrete.

   - Chiavi pubbliche SSH per AWS CodeCommit: seleziona Carica chiave pubblica SSH per caricare una chiave pubblica SSH in modo che l'utente possa comunicare con CodeCommit i repository tramite SSH.
   - Credenziali Git HTTPS per AWS CodeCommit: seleziona Genera credenziali per generare un set univoco di credenziali utente da utilizzare con i repository Git. Seleziona Scarica credenziali per salvare il nome utente e la password in un file .csv. Questa è l'unica volta in cui le informazioni sono disponibili. Se dimentichi o perdi la password, dovrai reimpostarla.
   - Credenziali per Amazon Keyspaces (per Apache Cassandra): seleziona Genera credenziali per generare le credenziali utente specifiche del servizio da utilizzare con Amazon Keyspaces. Seleziona Scarica credenziali per salvare il nome utente e la password in un file .csv. Questa è l'unica volta in cui le informazioni sono disponibili. Se dimentichi o perdi la password, dovrai reimpostarla.

     > **Important**
     >
     > Service-specific le credenziali sono credenziali a lungo termine associate a un utente IAM specifico e possono essere utilizzate solo per il servizio per cui sono state create. Per concedere ai ruoli IAM o alle identità federate le autorizzazioni per accedere a tutte le tue AWS risorse utilizzando credenziali temporanee,

utilizza l' AWS autenticazione con il plug-in di autenticazione Sigv4 per Amazon Keyspaces. Per ulteriori informazioni, consulta Utilizzo di credenziali temporanee per connettersi ad Amazon Keyspaces (per Apache Cassandra) utilizzando un ruolo IAM e il plugin SIGv4 nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).

- X.509 Firma dei certificati: seleziona Crea X.509 certificato se devi effettuare SOAP-protocol richieste sicure e ti trovi in una regione non supportata da. AWS Certificate Manager ACM è lo strumento preferito per il provisioning, la gestione e la distribuzione dei certificati del server. Per ulteriori informazioni sull'utilizzo di ACM, consulta la Guida per l'utente di AWS Certificate Manager .

Hai creato un utente con accesso programmatico e lo hai configurato con la PowerUserAccess funzione job. La politica delle autorizzazioni di questo utente garantisce l'accesso completo a tutti i servizi ad eccezione di IAM e. AWS Organizations

Puoi utilizzare questo stesso processo per fornire ai carichi di lavoro aggiuntivi l'accesso programmatico alle tue Account AWS risorse, se i carichi di lavoro non sono in grado di assumere ruoli IAM. Questa procedura ha utilizzato la policy PowerUserAccess gestita per assegnare le autorizzazioni. Per seguire la best practice del privilegio minimo, prendi in considerazione l'utilizzo di una policy più restrittiva o la creazione di una policy personalizzata che limiti l'accesso alle sole risorse richieste dal programma. Per informazioni sull'utilizzo delle politiche che limitano le autorizzazioni degli utenti a AWS risorse specifiche, consulta Gestione degli accessi per AWS risorse e. Esempi di policy basate su identità IAM Per aggiungere altri utenti al gruppo di utenti dopo averlo creato, consulta Modificare gli utenti nei gruppi IAM.

AWS CLI

1. Crea un utente denominato **`Automation`**.

   - aws iam create-user

   ```
   aws iam create-user \
       --user-name Automation
   ```

2. Crea un gruppo di utenti IAM denominato**`AutomationGroup`**, `PowerUserAccess` allega la policy AWS gestita al gruppo e quindi aggiungi l'**`Automation`**utente al gruppo.

> **Note**
>
> Una policy gestita da AWS è una policy autonoma che viene creata e amministrata da AWS. Ogni policy ha il proprio nome della risorsa Amazon (ARN) che include il nome della policy. Ad esempio, `arn:aws:iam::aws:policy/IAMReadOnlyAccess` è una policy AWS gestita. Per ulteriori informazioni sugli ARN, consultare la pagina ARN IAM. Per un elenco delle politiche AWS gestite per Servizi AWS, vedere le politiche AWS gestite.

- aws iam create-group

```
aws iam create-group \
    --group-name AutomationGroup
```

- aws iam attach-group-policy

```
aws iam attach-group-policy \
    --policy-arn arn:aws:iam::aws:policy/PowerUserAccess \
    --group-name AutomationGroup
```

- aws iam add-user-to-group

```
aws iam add-user-to-group \
    --user-name Automation \
    --group-name AutomationGroup
```

- Esegui il comando aws iam get-group per elencare **AutomationGroup** e i relativi membri.

```
aws iam get-group \
    --group-name AutomationGroup
```

3. Crea quindi le credenziali di sicurezza necessarie per il carico di lavoro.

- Crea chiavi di accesso per i test: aws iam create-access-key

```
aws iam create-access-key \
    --user-name Automation
```

Il risultato di questo comando mostra la chiave di accesso segreta e l'ID chiave di accesso. Registra e archivia queste informazioni in un posto sicuro. Se queste credenziali vengono perse, non possono essere recuperate e dovrai creare una nuova chiave di accesso.

> **Important**
>
> Queste chiavi di accesso dell'utente IAM sono credenziali a lungo termine che presentano un rischio per la sicurezza del tuo account. Una volta completato il test, ti consigliamo di eliminare queste chiavi di accesso. Se hai scenari in cui stai considerando le chiavi di accesso, verifica se puoi abilitare l'MFA per il tuo utente IAM del carico di lavoro e usa aws sts get-session-token per ottenere credenziali temporanee per la sessione invece di utilizzare le chiavi di accesso IAM.

- Carica le chiavi pubbliche SSH per AWS CodeCommit — aws iam upload-ssh-public-key

Nel seguente esempio si assume che le tue chiavi pubbliche SSH siano memorizzate nel file `sshkey.pub`.

```
aws upload-ssh-public-key \
    --user-name Automation \
    --ssh-public-key-body file://sshkey.pub
```

- Carica X.509 un certificato di firma: aws iam upload-signing-certificate

Carica un X.509 certificato se devi effettuare SOAP-protocol richieste sicure e ti trovi in una regione non supportata da. AWS Certificate Manager ACM è lo strumento preferito per il provisioning, la gestione e la distribuzione dei certificati del server. Per ulteriori informazioni sull'utilizzo di ACM, consulta la Guida per l'utente di AWS Certificate Manager .

L'esempio seguente presuppone che il certificato di X.509 firma sia archiviato nel file`certificate.pem`.

```
aws iam upload-signing-certificate \
--user-name Automation \
--certificate-body file://certificate.pem
```

Puoi utilizzare questo stesso processo per concedere a carichi di lavoro aggiuntivi l'accesso programmatico alle tue Account AWS risorse, se i carichi di lavoro non sono in grado di assumere ruoli IAM. Questa procedura ha utilizzato la policy PowerUserAccess gestita per assegnare le autorizzazioni. Per seguire la best practice del privilegio minimo, prendi in considerazione l'utilizzo di una policy più restrittiva o la creazione di una policy personalizzata che limiti l'accesso alle sole risorse richieste dal programma. Per informazioni sull'utilizzo delle politiche che limitano le autorizzazioni degli utenti a AWS risorse specifiche, consulta Gestione degli accessi per AWS risorse e. Esempi di policy basate su identità IAM Per aggiungere altri utenti al gruppo di utenti dopo averlo creato, consulta Modificare gli utenti nei gruppi IAM.

Utilizzo dell'autenticazione a più fattori con le identità

L'aggiunta dell'autenticazione a più fattori (MFA) per le tue identità è un'altra best practice di IAM. L'MFA è un livello di sicurezza aggiuntivo che richiede agli utenti di fornire fattori di autenticazione aggiuntivi dopo aver fornito nome utente e password per verificare la propria identità. Migliora in modo significativo la sicurezza rendendo molto più difficile agli aggressori ottenere un accesso non autorizzato anche nel caso in cui la password di un utente sia compromessa. La MFA è ampiamente adottata come best practice per proteggere l'accesso agli account online, ai servizi cloud e ad altre risorse sensibili. AWS supporta l'MFA per utenti root, utenti IAM, utenti in IAM Identity Center, Builder ID e utenti federati. Per una maggiore sicurezza, puoi creare policy che richiedono la configurazione dell'MFA prima di consentire a un utente di accedere alle risorse o intraprendere azioni specifiche e collegare queste policy ai tuoi ruoli IAM. Centro identità IAM è preconfigurato con l'MFA attivata per impostazione predefinita, in modo che tutti gli utenti di Centro identità IAM debbano accedere con l'MFA oltre al nome utente e alla password.

> **Note**
>
> Tutti i Account AWS tipi (account standalone, di gestione e account membri) richiedono la configurazione dell'MFA per l'utente root. Gli utenti devono registrare l'MFA entro 35 giorni dal

primo tentativo di accesso per accedere alla MFA se la Console di gestione AWS MFA non è già abilitata.

Per ulteriori informazioni, consulta Configure MFA in IAM Identity Center e AWS Multi-factor autenticazione in IAM.

Preparazione per le autorizzazioni con privilegi minimi

L'utilizzo delle autorizzazioni con privilegi minimi è uno dei suggerimenti di best practice di IAM. Lo scopo delle autorizzazioni con privilegi minimi è concedere agli utenti solo le autorizzazioni richieste per eseguire una determinata attività. Durante la configurazione, considera come intendi supportare le autorizzazioni con privilegi minimi. L'utente root, l'utente amministratore e l'utente IAM con accesso di emergenza dispongono di autorizzazioni avanzate che non sono necessarie per le attività quotidiane. Mentre stai imparando a conoscere AWS e testare diversi servizi, ti consigliamo di creare almeno un utente aggiuntivo in IAM Identity Center con autorizzazioni inferiori che puoi utilizzare in diversi scenari. È possibile utilizzare le policy IAM per definire le operazioni che possono essere eseguite su risorse specifiche in determinate condizioni e connettersi quindi a tali risorse con l'account con meno privilegi.

Se utilizzi il Centro identità IAM, per iniziare considera l'uso dei set di autorizzazioni del Centro identità IAM stesso. Per ulteriori informazioni, consulta la pagina Creazione di un set di autorizzazioni nella Guida per l'utente di Centro identità IAM.

Se non utilizzi Centro identità IAM, utilizza i ruoli IAM per definire le autorizzazioni per diverse entità IAM. Per ulteriori informazioni, consulta Creazione di ruoli IAM.

Sia i ruoli IAM che i set di autorizzazioni IAM Identity Center possono utilizzare policy AWS gestite basate sulle funzioni lavorative. Per i dettagli sulle autorizzazioni concesse da queste policy, consulta AWS politiche gestite per le funzioni lavorative.

> **Important**
>
> Tieni presente che le policy AWS gestite potrebbero non concedere autorizzazioni con privilegi minimi per i tuoi casi d'uso specifici perché sono disponibili per l'uso da parte di tutti i clienti. AWS Dopo la configurazione, consigliamo di utilizzare il Sistema di analisi degli accessi IAM per generare policy con privilegi minimi in funzione dell'attività di accesso

collegata in AWS CloudTrail. Per ulteriori informazioni sulla generazione delle policy, consulta IAM Access Analyzer policy generation.

Per iniziare, ti consigliamo di utilizzare le politiche AWS gestite per concedere le autorizzazioni. Dopo un periodo predefinito di inattività (ad esempio 90 giorni), è possibile esaminare i servizi a cui le persone e i carichi di lavoro hanno effettuato l'accesso. Quindi puoi creare una nuova politica gestita dal cliente con autorizzazioni ridotte per sostituire la politica AWS gestita. La nuova policy dovrebbe includere solo i servizi a cui è stato effettuato l'accesso durante il periodo di campionamento. Aggiorna le tue autorizzazioni per rimuovere la politica AWS gestita e allegare la nuova politica gestita dal cliente che hai creato.

Per istruzioni su come perfezionare le autorizzazioni, consulta. Revisione delle informazioni dell'ultimo accesso per il tuo AWS account Dopo aver identificato i servizi non utilizzati, consulta. Generazione di una policy basata sull'attività di accesso

Revisione delle informazioni dell'ultimo accesso per il tuo AWS account

Puoi visualizzare le informazioni sull'ultimo accesso al servizio per IAM utilizzando la console IAM o AWS l'API. AWS CLI Per informazioni importanti sui dati, sulle autorizzazioni necessarie, sulla risoluzione dei problemi e sulle regioni supportate, consulta Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

È possibile visualizzare le informazioni per i seguenti tipi di risorsa in IAM. In ogni caso, i dati includono i servizi consentiti per il periodo di reporting specificato:

- Utente IAM: visualizza l'ultima volta che l'utente ha provato ad accedere a ogni servizio consentito.
- Gruppo IAM: visualizza informazioni sull'ultima volta che un membro del gruppo IAM ha provato ad accedere a ogni servizio consentito. Questo report include anche il numero totale di membri che hanno tentato di accedere.
- Ruolo IAM: visualizza l'ultima volta che qualcuno ha utilizzato il ruolo nel tentativo di accedere a ogni servizio consentito.
- Policy: visualizza le informazioni sull'ultima volta che un utente o un ruolo ha provato ad accedere a ogni servizio consentito. Questo report include anche il numero totale di entità che hanno tentato di accedere.

> **Note**
>
> Prima di visualizzare i dati di accesso per una risorsa in IAM, assicurati di comprendere il periodo di riferimento, le entità incluse nel report e i tipi di policy valutati per i tuoi dati. Per ulteriori dettagli, consultare the section called "Cose da sapere sulle ultime informazioni di accesso".

Per ulteriori informazioni sulle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Per esaminare le informazioni relative all'ultimo accesso per un Account AWS

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Gruppi di utenti, Utenti, Ruoli o Policy.
4. Seleziona il nome di qualsiasi utente, gruppo di utenti, ruolo o policy per aprire la relativa pagina Riepilogo e seleziona la scheda Ultimo accesso. Visualizzare le seguenti informazioni, in base alla risorsa scelta:

   - Gruppo di utenti: visualizza l'elenco dei servizi a cui i membri del gruppo di utenti possono accedere. È inoltre possibile visualizzare l'ultima volta che un membro ha effettuato l'accesso al servizio, quali policy di gruppo ha utilizzato e quale membro del gruppo ha effettuato la richiesta. Scegli il nome della policy per scoprire se è una policy gestita o una policy del gruppo di utenti in linea. Scegli il nome del membro del gruppo per visualizzare tutti i membri del gruppo di utenti e il momento in cui hanno effettuato l'ultimo accesso al servizio.
   - Utente: visualizza l'elenco dei servizi a cui l'utente può accedere. È inoltre possibile visualizzare l'ultima volta che hanno effettuato l'accesso al servizio e i criteri associati attualmente all'utente. Scegli il nome della policy per sapere se si tratta di una policy gestita, di una policy utente in linea o di una policy in linea per il gruppo di utenti.

- Ruolo: visualizzare l'elenco dei servizi cui il ruolo può accedere, il suo ultimo accesso al servizio e le policy utilizzate. Scegliere il nome della policy per scoprire se è una policy gestita o una policy del ruolo inline.
- Policy: visualizza l'elenco dei servizi con le operazioni consentite nella policy. È inoltre possibile visualizzare l'ultima volta che il criterio è stato utilizzato per accedere al servizio e l'entità (utente o ruolo) utilizzata dal criterio. La data dell'Ultimo accesso include anche quando viene concesso l'accesso a questa policy tramite un'altra policy. Scegliere il nome dell'entità per scoprire a quali entità è collegata questa policy e l'ultimo accesso al servizio da parte dell'entità.

5. Nella colonna Servizio della tabella, scegli il nome di uno dei servizi che include le informazioni relative all'ultimo accesso a un'operazione per visualizzare un elenco delle operazioni di gestione alle quali le entità IAM hanno provato ad accedere. È possibile visualizzare la Regione AWS e un timestamp che indica l'ultimo tentativo di eseguire l'operazione da parte di un utente.
6. La colonna Ultimo accesso viene visualizzata per i servizi e le operazioni di gestione dei servizi che includono le informazioni relative all'ultimo accesso a un'operazione. Esaminare i seguenti risultati possibili restituiti in questa colonna. Questi risultati variano a seconda che sia consentito un servizio o un'azione, sia stato effettuato l'accesso e se viene monitorato da AWS per le ultime informazioni di accesso.

   <number of> giorni fa

   Numero di giorni dall'utilizzo del servizio o dell'azione nel periodo di registrazione. Il periodo di monitoraggio per i servizi è degli ultimi 400 giorni. Il periodo di monitoraggio delle operazioni Amazon S3 è iniziato il 12 aprile 2020. Il periodo di monitoraggio delle azioni di Amazon EC2, IAM e Lambda è iniziato il 7 aprile 2021. Il periodo di monitoraggio per tutti gli altri servizi è iniziato il 23 maggio 2023. Per ulteriori informazioni sulle date di inizio del monitoraggio per ciascuna di esse Regione AWS, consultaDove AWS tiene traccia delle ultime informazioni a cui si accede.

   Non accessibile nel periodo di tracciabilità

   Il servizio o l'azione tracciati non sono stati utilizzati da un'entità nel periodo di registrazione.

   È possibile disporre delle autorizzazioni per un'azione che non viene visualizzata nell'elenco. Ciò può verificarsi se le informazioni di monitoraggio per l'operazione non sono attualmente

incluse da AWS. Non è consigliabile prendere decisioni sulle autorizzazioni basate esclusivamente sull'assenza di informazioni di tracciamento. Si consiglia invece di utilizzare queste informazioni per informare e supportare la strategia generale di concessione di privilegi minimi. Controllare i criteri per verificare che il livello di accesso sia appropriato.

AWS CLI

Puoi utilizzare il AWS CLI per recuperare informazioni sull'ultima volta che una risorsa IAM del tuo computer Account AWS è stata utilizzata per tentare di accedere ai AWS servizi e alle azioni Amazon S3, Amazon EC2, IAM e Lambda. Una risorsa IAM può essere un utente, un gruppo di utenti, un ruolo o una policy.

- Genera un report per le risorse IAM in un Account AWS. La richiesta deve includere l'ARN della risorsa IAM (utente, gruppo di utenti, ruolo o policy) per cui desideri un report. È possibile specificare il livello di granularità che si desidera generare nel report per visualizzare i dettagli di accesso per i servizi o per entrambi i servizi e le azioni. Viene restituito un `job-id` che è possibile utilizzare nelle operazioni `get-service-last-accessed-details` e `get-service-last-accessed-details-with-entities` per monitorare `job-status` finché il processo viene completato.

  - aws iam -details generate-service-last-accessed

  a. Recuperare i dettagli sul report utilizzando il parametro `job-id` dal passaggio precedente.

     - aws iam get-service-last-accessed -details

     Questa operazione restituisce le seguenti informazioni, a seconda del tipo di risorsa richiesto nell'operazione `generate-service-last-accessed-details`:

     - Utente: restituisce un elenco dei servizi cui l'utente specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo dell'utente e l'ARN dell'utente.
     - Gruppo di utenti: restituisce un elenco dei servizi a cui i membri del gruppo di utenti specificato possono accedere utilizzando la policy collegata al gruppo di utenti. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo effettuato da qualsiasi membro del gruppo di utenti. Inoltre, restituisce l'ARN dell'utente e il numero

totale di membri del gruppo di utenti che hanno provato ad accedere al servizio. Usa l'GetServiceLastAccessedDetailsWithEntitiesoperazione per recuperare un elenco di tutti i membri.

- Ruolo: restituisce un elenco dei servizi cui il ruolo specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo del ruolo e l'ARN del ruolo.
- Policy: restituisce un elenco dei servizi per i quali la policy specificata consente l'accesso. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo di accesso al servizio da parte di un'entità (utente o ruolo), utilizzando la policy. Inoltre, restituisce l'ARN di quell'entità e il numero totale di entità che hanno tentato di accedere.

b. Scopri di più sulle entità che hanno utilizzato autorizzazioni di policy o gruppo di utenti in un tentativo di accesso a un servizio specifico. Questa operazione restituisce un elenco delle entità insieme all'ARN, l'ID, il nome, il percorso, il tipo (utente o ruolo) e l'ultimo tentativo di accesso al servizio di ogni entità. È anche possibile utilizzare questa operazione per gli utenti e i ruoli, ma restituisce informazioni solo su tale entità.

   - era iam get-service-last-accessed - details-with-entities

c. Scopri di più sulle policy basate sull'identità utilizzate da un'identità (utente, gruppo di utenti o ruolo) in un tentativo di accesso a un servizio specifico. Quando si specifica un'identità e un servizio, questa operazione restituisce un elenco delle policy di autorizzazione che l'identità può utilizzare per accedere al servizio specificato. Questa operazione fornisce lo stato attuale delle policy e non dipende dal report generato. Inoltre, non restituisce altri tipi di policy, come policy basate sulle risorse, liste di controllo degli accessi, AWS Organizations policy, limiti di autorizzazione IAM o policy di sessione. Per ulteriori informazioni, consulta Tipi di policy o Valutazione delle policy per le richieste all'interno di un singolo account.

   - come iam -access list-policies-granting-service

API

Puoi utilizzare l' AWS API per recuperare informazioni sull'ultima volta che una risorsa IAM è stata utilizzata per tentare di accedere ai AWS servizi e alle azioni Amazon S3, Amazon EC2, IAM e Lambda. Una risorsa IAM può essere un utente, un gruppo di utenti, un ruolo o una policy. È

possibile specificare il livello di granularità da generare nel report per visualizzare i dettagli relativi ai servizi o ai servizi e alle azioni.

1. Generare un report. La richiesta deve includere l'ARN della risorsa IAM (utente, gruppo di utenti, ruolo o policy) per cui desideri un report. Viene restituito un `JobId` che è possibile utilizzare nelle operazioni `GetServiceLastAccessedDetails` e `GetServiceLastAccessedDetailsWithEntities` per monitorare il `JobStatus` finché il processo viene completato.

   - GenerateServiceLastAccessedDetails

2. Recuperare i dettagli sul report utilizzando il parametro `JobId` dal passaggio precedente.

   - GetServiceLastAccessedDetails

   Questa operazione restituisce le seguenti informazioni, a seconda del tipo di risorsa richiesto nell'operazione `GenerateServiceLastAccessedDetails`:

   - Utente: restituisce un elenco dei servizi cui l'utente specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo dell'utente e l'ARN dell'utente.
   - Gruppo di utenti: restituisce un elenco dei servizi a cui i membri del gruppo di utenti specificato possono accedere utilizzando la policy collegata al gruppo di utenti. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo effettuato da qualsiasi membro del gruppo di utenti. Inoltre, restituisce l'ARN dell'utente e il numero totale di membri del gruppo di utenti che hanno provato ad accedere al servizio. Usa l'GetServiceLastAccessedDetailsWithEntitiesoperazione per recuperare un elenco di tutti i membri.
   - Ruolo: restituisce un elenco dei servizi cui il ruolo specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo del ruolo e l'ARN del ruolo.
   - Policy: restituisce un elenco dei servizi per i quali la policy specificata consente l'accesso. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo di accesso al servizio da parte di un'entità (utente o ruolo), utilizzando la policy. Inoltre, restituisce l'ARN di quell'entità e il numero totale di entità che hanno tentato di accedere.

3. Scopri di più sulle entità che hanno utilizzato autorizzazioni di policy o gruppo di utenti in un tentativo di accesso a un servizio specifico. Questa operazione restituisce un elenco delle

entità insieme all'ARN, l'ID, il nome, il percorso, il tipo (utente o ruolo) e l'ultimo tentativo di accesso al servizio di ogni entità. È anche possibile utilizzare questa operazione per gli utenti e i ruoli, ma restituisce informazioni solo su tale entità.

- GetServiceLastAccessedDetailsWithEntities

4. Scopri di più sulle policy basate sull'identità utilizzate da un'identità (utente, gruppo di utenti o ruolo) in un tentativo di accesso a un servizio specifico. Quando si specifica un'identità e un servizio, questa operazione restituisce un elenco delle policy di autorizzazione che l'identità può utilizzare per accedere al servizio specificato. Questa operazione fornisce lo stato attuale delle policy e non dipende dal report generato. Inoltre, non restituisce altri tipi di policy, come policy basate sulle risorse, liste di controllo degli accessi, policy, limiti delle autorizzazioni IAM o policy AWS Organizations di sessione. Per ulteriori informazioni, consulta Tipi di policy o Valutazione delle policy per le richieste all'interno di un singolo account.

   - ListPoliciesGrantingServiceAccess

Generazione di una policy basata sull'attività di accesso

Puoi utilizzare l'attività di accesso registrata AWS CloudTrail per un utente o un ruolo IAM per fare in modo che IAM Access Analyzer generi una policy gestita dal cliente per consentire l'accesso solo ai servizi di cui hanno bisogno utenti e ruoli specifici.

Quando Sistema di analisi degli accessi IAM genera una policy IAM, vengono restituite informazioni che consentono di personalizzare ulteriormente la policy. Quando viene generata una policy, è possibile restituire due categorie di informazioni:

- Policy con informazioni a livello di azione: per alcuni AWS servizi, come Amazon EC2, IAM Access Analyzer è in grado di identificare le azioni rilevate nei CloudTrail tuoi eventi ed elenca le azioni utilizzate nella policy che genera. Per un elenco dei servizi supportati, consulta Servizi di generazione di policy per Sistema di analisi degli accessi IAM. Per alcuni servizi, Sistema di analisi degli accessi IAM richiede l'aggiunta azioni per i servizi alla policy generata.
- Policy con informazioni sui livelli di servizio – Sistema di analisi degli accessi IAM utilizza le informazioni relative all'ultimo accesso per creare un modello di policy con tutti i servizi utilizzati di recente. Quando utilizzi Console di gestione AWS, ti chiediamo di esaminare i servizi e aggiungere azioni per completare la policy.

Per generare una policy basata sull'attività di accesso

Nella procedura seguente ridurremo le autorizzazioni concesse a un ruolo in modo che corrispondano all'utilizzo di un utente. Quando scegli un utente, scegli un utente il cui utilizzo esemplifica il ruolo. Molti clienti configurano account utente di prova con PowerUserautorizzazioni e poi fanno eseguire loro una serie specifica di attività per un breve periodo di tempo per determinare quale accesso è necessario per eseguire tali attività,

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel riquadro di navigazione, seleziona Utenti, quindi seleziona l'utente per visualizzare i dettagli dell'utente.
4. Nella scheda Autorizzazioni, in Genera politica basata sugli CloudTrail eventi, scegli Genera politica.
5. Nella pagina Genera policy, configura i seguenti elementi:
   - Per Seleziona periodo di tempo, scegli Ultimi 7 giorni.
   - Per analizzare il CloudTrail percorso, seleziona la regione e il percorso in cui viene registrata l'attività di questo utente.
   - Scegli Crea e utilizza un nuovo ruolo di servizio.
6. Scegli Genera policy, quindi attendi fino alla creazione del ruolo. Non aggiornare o uscire dalla pagina della console finché non viene visualizzato il messaggio di notifica Generazione della policy in corso.
7. Dopo aver generato la politica, è necessario esaminarla e personalizzarla in base alle esigenze con l'account IDs e ARNs le risorse. Inoltre, la policy generata automaticamente potrebbe non includere le informazioni a livello di azione necessarie per completare la policy. Per ulteriori informazioni, consulta Generazione delle policy di IAM Access Analyzer.

   Ad esempio, è possibile modificare la prima istruzione che include l'effetto `Allow` e l'elemento `NotAction` per consentire solo le azioni di Amazon EC2 e Amazon S3. A tale scopo, sostituirla con l'istruzione con l'ID `FullAccessToSomeServices`. La nuova policy sarà simile alla seguente policy di esempio.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
          "Sid": "FullAccessToSomeServices",
          "Effect": "Allow",
          "Action": [
              "ec2:*",
              "s3:*"
          ],
          "Resource": "*"
      },
      {
          "Effect": "Allow",
          "Action": [
              "iam:CreateServiceLinkedRole",
              "iam:DeleteServiceLinkedRole",
              "iam:ListRoles",
              "organizations:DescribeOrganization"
          ],
          "Resource": "*"
      }
  ]
}
```

8. Per supportare le best practice per assegnare privilegi minimi, rivedi e correggi eventuali errori, avvisi o suggerimenti restituiti durante la convalida delle policy.
9. Per ridurre ulteriormente le autorizzazioni delle tue policy per azioni e risorse specifiche, visualizza i tuoi eventi nella Cronologia degli eventi. CloudTrail Qui è possibile visualizzare informazioni dettagliate sulle operazioni e risorse specifiche a cui l'utente ha effettuato l'accesso. Per ulteriori informazioni, consulta Visualizzazione CloudTrail degli eventi nella CloudTrail console nella Guida per l'AWS CloudTrail utente.
10. Dopo aver esaminato e convalidato la policy, salvala con un nome descrittivo.
11. Vai alla pagina Ruoli e scegli il ruolo che le persone assumeranno quando eseguiranno le attività consentite dalla nuova policy.
12. Seleziona la scheda Autorizzazioni, scegli Aggiungi autorizzazioni, quindi seleziona Collega policy.

13. Nella pagina Collega policy di autorizzazione, nell'elenco Altre policy di autorizzazione, seleziona la policy che hai creato, quindi scegli Collega policy.
14. Si torna alla pagina dei dettagli del ruolo. Al ruolo sono associate due politiche, la politica AWS gestita precedente, ad esempio PowerUserAccess, e la nuova politica. Seleziona la casella di controllo relativa alla politica AWS gestita, quindi scegli Rimuovi. Quando ti viene chiesto di confermare la rimozione, scegli Rimuovi.

Gli utenti IAM, i principali federati SAML e OIDC e i carichi di lavoro che assumono questo ruolo ora hanno un accesso ridotto in base alla nuova policy che hai creato.

AWS CLI

È possibile utilizzare i seguenti comandi per generare una policy utilizzando AWS CLI.

Per generare una policy

- aws accessanalyzer start-policy-generation

Per visualizzare una policy generata

- aws access analyzer get-generated-policy

Per annullare una richiesta di generazione di policy

- aws access analyzer cancel-policy-generation

Per visualizzare un elenco di richieste di generazione di policy

- aws access analyzer list-policy-generations

API

È possibile seguire le seguenti operazioni per generare una policy utilizzando l'API AWS .

Per generare una policy

- StartPolicyGeneration

Per visualizzare una policy generata

- GetGeneratedPolicy

Per annullare una richiesta di generazione di policy

- CancelPolicyGeneration

Per visualizzare un elenco di richieste di generazione di policy

- ListPolicyGenerations

Utilizzo della ricerca per trovare risorse IAM

Mentre esamini i risultati degli accessi, puoi utilizzare la pagina di ricerca della console IAM come opzione più rapida per trovare le risorse IAM. Puoi cercare le risorse utilizzando nomi di risorse parziali o ARN.

Console

La funzione di ricerca della console IAM è in grado di trovare quanto segue:

- Nomi di entità IAM che corrispondono alle parole chiave della ricerca (per utenti, gruppi, ruoli, provider di identità e policy)
- Attività corrispondenti alle parole chiave usate per la ricerca

La funzionalità di ricerca della console IAM non restituisce informazioni su IAM Access Analyzer.

Ogni riga visualizzata nei risultati della ricerca è un link attivo. Ad esempio, puoi selezionare il nome utente nei risultati di ricerca, per andare direttamente alla pagina dei dettagli dell'utente. In alternativa, puoi selezionare collegamento all'operazione, come ad esempio Create User (Crea utente), per accedere alla pagina Create User (Crea utente).

> **Note**
>
> Per le ricerche delle chiavi di accesso è necessario immettere nella casella di ricerca l'ID completo della chiave di accesso. Il risultato della ricerca mostra l'utente associato a tale

chiave. A questo punto, potrai andare direttamente alla pagina dell'utente e gestirne la chiave di accesso.

Utilizza la pagina Cerca della console IAM per trovare gli elementi relativi a un determinato account.

Come cercare gli elementi nella console IAM

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere alla Guida per AWS l'AWS Sign-In utente.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel riquadro di navigazione selezionare Search (Cerca).
4. Nella casella Search (Cerca) digitare le parole chiave usate per la ricerca.
5. Nell'elenco dei risultati della ricerca selezionare un link per passare alla parte corrispondente della console.

Le icone riportate di seguito consentono di identificare i tipi di elementi trovati mediante una ricerca:

Icon	Description
	Utenti IAM
	Gruppi IAM
	Ruoli IAM
	Policy IAM
	Attività come la creazione di utenti o il collegamento di policy

Icon	Description
	Risultati della parola chiave **`delete`**

Esempi di frasi da cercare

Per le ricerche in IAM puoi utilizzare le frasi riportate di seguito: Sostituisci i termini in corsivo con i nomi di veri utenti IAM, gruppi, ruoli, chiavi di accesso, policy o gestori di identità da individuare.

- ***user_name*** o ***group_name*** o ***role_name*** o ***policy_name*** o ***identity_provider_name***
- ***access_key***
- **add user *user_name* to groups** o **add users to group *group_name***
- **remove user *user_name* from groups**
- **delete *user_name*** o **delete *group_name*** o **delete *role_name*** o **delete *policy_name*** o **delete *identity_provider_name***
- **manage access keys *user_name***
- **manage signing certificates *user_name***
- **users**
- **manage MFA for *user_name***
- **manage password for *user_name***
- **create role**
- **password policy**
- **edit trust policy for role *role_name***
- **show policy document for role *role_name***
- **attach policy to *role_name***
- **create managed policy**
- **create user**
- **create group**
- **attach policy to *group_name***
- **attach entities to *policy_name***
- **detach entities from *policy_name***

Migliori pratiche di sicurezza e casi d'uso in AWS Identity and Access Management

AWS Identity and Access Management (IAM) offre diverse funzionalità di sicurezza da prendere in considerazione durante lo sviluppo e l'implementazione delle proprie politiche di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa.

Queste best practice potrebbero non essere appropriate o sufficienti per il tuo ambiente specifico. Pertanto, considerale come informazioni utili piuttosto che come requisiti.

Per ottenere i vantaggi maggiori da IAM, leggi con attenzione le best practice consigliate. Un modo per fare ciò è verificare come IAM viene utilizzato negli scenari reali per funzionare con altri Servizi AWS.

Argomenti

- Best practice per la sicurezza in IAM
- Le migliori pratiche per gli utenti root Account AWS
- Casi d'uso di business per IAM

Best practice per la sicurezza in IAM

Follow us on Twitter

Per proteggere AWS le tue risorse, segui queste best practice per AWS Identity and Access Management (IAM).

Dopo aver implementato queste best practice, consulta altri argomenti Sicurezza in IAM e AWS STS di sicurezza, tra cui la convalida della conformità, la resilienza e la sicurezza dell'infrastruttura.

Argomenti

- Richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzo di credenziali temporanee
- Richiedi ai carichi di lavoro di utilizzare credenziali temporanee con ruoli IAM per accedere AWS
- Richiedere l'autenticazione a più fattori (MFA)

- Aggiornamento delle chiavi di accesso quando necessario per i casi d'uso che richiedono credenziali a lungo termine
- Segui le best practice per proteggere le credenziali di utente root
- Assegna le autorizzazioni con privilegi minimi
- Inizia con AWS politiche gestite e passaggio alle autorizzazioni con privilegi minimi
- Utilizzare IAM Access Analyzer per generare policy con privilegi minimi in base all'attività di accesso
- Esaminare e rimuovere regolarmente utenti, ruoli, autorizzazioni, criteri e credenziali inutilizzati
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso
- Verifica dell'accesso multi-account e pubblico alle risorse con IAM Access Analyzer
- Usa IAM Access Analyzer per convalidare le tue policy IAM per ottenere autorizzazioni sicure e funzionali
- Stabilisci guardrail delle autorizzazioni su più account
- Utilizzare i limiti delle autorizzazioni per delegare la gestione delle autorizzazioni all'interno di un account

Richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzo di credenziali temporanee

Utenti umani, noti anche come identità umane, sono le persone, gli amministratori, gli sviluppatori, gli operatori e i consumatori delle tue applicazioni. Devono avere un'identità per accedere agli AWS ambienti e alle applicazioni dell'utente. Gli utenti umani membri dell'organizzazione sono noti anche come identità della forza lavoro. Gli utenti umani possono anche essere utenti esterni con cui collabori e che interagiscono con AWS le tue risorse. Possono farlo tramite un browser web, un'applicazione client, un'app mobile o strumenti interattivi a riga di comando.

Richiedi agli utenti umani di utilizzare credenziali temporanee per l'accesso. AWSÈ possibile utilizzare un provider di identità per consentire agli utenti umani di fornire un accesso federato Account AWS assumendo ruoli, che forniscono credenziali temporanee. Per una gestione centralizzata degli accessi, consigliamo di utilizzare Centro identità AWS IAM (IAM Identity Center) per gestire l'accesso ai tuoi account e le autorizzazioni all'interno di questi account. Puoi gestire le tue identità utente con IAM Identity Center o gestire le autorizzazioni di accesso per le identità degli utenti in IAM Identity Center da un provider di identità esterno. Per ulteriori informazioni, consulta Che cos'è Centro identità AWS IAM? nella Guida per l'utente di Centro identità AWS IAM .

Puoi utilizzare l'account access manager , una funzionalità IAM che ti consente di assegnare ruoli IAM esistenti negli account della tua organizzazione agli utenti e ai gruppi di IAM Identity Center. Account Access Manager ti dà accesso al set completo di funzionalità dei ruoli IAM, tra cui politiche di fiducia personalizzate, tag di ruolo per ABAC e percorsi di ruolo configurabili. Puoi usarlo insieme ai set di autorizzazioni o da solo.

Per ulteriori informazioni sui ruoli, consulta Termini e concetti dei ruoli.

Richiedi ai carichi di lavoro di utilizzare credenziali temporanee con ruoli IAM per accedere AWS

Un carico di lavoro è una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione o un processo di backend. Il carico di lavoro può avere applicazioni, strumenti operativi e componenti che richiedono le credenziali per effettuare richieste ai Servizi AWS, ad esempio richieste di lettura dei dati da Amazon S3.

Quando crei su un servizio di AWS elaborazione, come Amazon EC2 o Lambda, AWS fornisce le credenziali temporanee di un ruolo IAM a quella risorsa di calcolo. Le applicazioni scritte utilizzando un AWS SDK scopriranno e utilizzeranno queste credenziali temporanee per accedere alle AWS risorse e non è necessario distribuire credenziali di lunga durata per un utente IAM ai carichi di lavoro in esecuzione. AWS

I carichi di lavoro eseguiti all'esterno AWS, ad esempio i server locali, i server di altri provider cloud o le piattaforme gestite di integrazione continua e distribuzione continua (CI/CD), possono comunque utilizzare credenziali temporanee. Tuttavia, dovrai fornire queste credenziali temporanee al tuo carico di lavoro. Di seguito sono riportati i modi in cui puoi fornire credenziali temporanee ai tuoi carichi di lavoro:

- Puoi utilizzare IAM Roles Anywhere per richiedere AWS credenziali temporanee per il tuo carico di lavoro utilizzando un X.509 certificato proveniente dalla tua infrastruttura a chiave pubblica (PKI).
- Puoi chiamare l' AWS STS `AssumeRoleWithSAML`API per richiedere AWS credenziali temporanee per il tuo carico di lavoro utilizzando un'asserzione SAML di un provider di identità (IdP) esterno configurato all'interno del tuo. Account AWS
- Puoi chiamare l' AWS STS `AssumeRoleWithWebIdentity`API per richiedere AWS credenziali temporanee per il tuo carico di lavoro utilizzando un token web JSON (JWT) di un IdP configurato all'interno del tuo. Account AWS
- Puoi richiedere AWS credenziali temporanee dal tuo dispositivo IoT utilizzando l'autenticazione Mutual Transport Layer Security (MTLS) utilizzando. AWS IoT Core

Alcuni supportano Servizi AWS anche integrazioni per fornire credenziali temporanee ai carichi di lavoro eseguiti all'esterno di: AWS

- Amazon Elastic Container Service (Amazon ECS) Anywhere ti consente di eseguire attività Amazon ECS sulle tue risorse di calcolo e fornisce credenziali AWS temporanee per le attività Amazon ECS in esecuzione su tali risorse di calcolo.
- Amazon Elastic Kubernetes Service Hybrid Nodes ti consente di unire le tue risorse di calcolo in esecuzione all'esterno degli AWS as nodi a un cluster Amazon EKS. Amazon EKS può fornire credenziali temporanee ai pod Amazon EKS in esecuzione sulle tue risorse di calcolo.
- AWS Systems ManagerHybrid Activationsti consente di gestire le risorse di calcolo in esecuzione al di fuori dell' AWS utilizzo di SSM e fornisce AWS credenziali temporanee all'agente SSM in esecuzione sulle tue risorse di calcolo.

Richiedere l'autenticazione a più fattori (MFA)

Ti consigliamo di utilizzare i ruoli IAM per utenti umani e carichi di lavoro che accedono alle tue risorse AWS in modo che utilizzino credenziali temporanee. Tuttavia, per gli scenari in cui hai bisogno di utenti IAM o root nel tuo account, richiedi MFA per una maggiore sicurezza. Con MFA, gli utenti dispongono di un dispositivo che genera una risposta a una richiesta di autenticazione. Per completare la procedura di accesso, sono necessarie le credenziali dell'utente e la risposta generata dal dispositivo. Per ulteriori informazioni, consulta AWS Multi-factor autenticazione in IAM. Ti consigliamo di utilizzare un MFA resistente al phishing, come passkey e chiavi di sicurezza, laddove possibile. Per ulteriori informazioni, consulta Assegna una passkey o una chiave di sicurezza nel Console di gestione AWS.

Se utilizzi IAM Identity Center per la gestione centralizzata degli accessi per gli utenti umani, potrai utilizzare la funzionalità MFA di IAM Identity Center quando l'origine dell'identità è configurata con l'archivio di identità di IAM Identity Center, AWS Managed Microsoft AD o AD Connector. Per ulteriori informazioni sull'MFA in IAM Identity Center, consulta Multi-factor l'autenticazione nella Guida per l'utente. Centro identità AWS IAM

Aggiornamento delle chiavi di accesso quando necessario per i casi d'uso che richiedono credenziali a lungo termine

Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare credenziali a lungo termine come le chiavi di accesso. Tuttavia, per gli scenari in cui sono necessari utenti IAM con accesso a livello di programmazione e credenziali a lungo termine, si consiglia di aggiornare le

chiavi di accesso all'occorrenza, ad esempio quando un dipendente lascia l'azienda. Ti consigliamo di utilizzare informazioni utilizzate per l'ultimo accesso IAM per aggiornare e rimuovere le chiavi di accesso in modo sicuro. Per ulteriori informazioni, consulta Aggiornare le chiavi di accesso.

Esistono casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM in. AWS Alcuni dei casi d'uso sono i seguenti:

- Carichi di lavoro che non possono utilizzare ruoli IAM: è possibile eseguire un carico di lavoro da una posizione che deve accedere a AWS. In alcune situazioni, non è possibile utilizzare i ruoli IAM per fornire credenziali temporanee, ad esempio per WordPress i plugin. In queste situazioni, per autenticarti a AWS usa le chiavi di accesso a lungo termine dell'utente IAM per quel carico di lavoro.
- Third-party AWS clienti: se utilizzi strumenti che non supportano l'accesso con IAM Identity Center, come AWS client o fornitori di terze parti che non sono ospitati su IAM AWS, utilizza le chiavi di accesso a lungo termine degli utenti IAM.
- AWS CodeCommit accesso: se utilizzi CodeCommit per archiviare il codice, puoi utilizzare un utente IAM con chiavi SSH o credenziali specifiche del servizio CodeCommit per l'autenticazione nei tuoi repository. Si consiglia di eseguire questa operazione oltre a utilizzare un utente di IAM Identity Center per l'autenticazione normale. Gli utenti di IAM Identity Center sono le persone della tua forza lavoro che hanno bisogno di accedere alle tue o alle tue applicazioni cloud. Account AWS Per consentire agli utenti di accedere ai tuoi CodeCommit repository senza configurare gli utenti IAM, puoi configurare l'utilità. git-remote-codecommit Per ulteriori informazioni su IAM e CodeCommit, consulta. Credenziali IAM per CodeCommit: credenziali Git, chiavi SSH e chiavi di accesso AWS Per ulteriori informazioni sulla configurazione dell'git-remote-codecommitutilità, consulta Connessione ai AWS CodeCommit repository con credenziali rotanti nella Guida per l'utente. AWS CodeCommit
- Accesso ad Amazon Keyspaces (per Apache Cassandra): in una situazione in cui non è possibile utilizzare gli utenti in IAM Identity Center, ad esempio per scopi di test per la compatibilità con Cassandra, puoi utilizzare un utente IAM con credenziali specifiche del servizio per l'autenticazione con Amazon Keyspaces. Gli utenti di IAM Identity Center sono le persone della tua forza lavoro che hanno bisogno di accedere alle tue Account AWS o alle tue applicazioni cloud. Puoi anche connetterti ad Amazon Keyspaces utilizzando credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di credenziali temporanee per connettersi ad Amazon Keyspaces utilizzando un ruolo IAM e il plugin SIGv4 nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).

Segui le best practice per proteggere le credenziali di utente root

Quando crei un Account AWS, stabilisci le credenziali dell'utente root per accedere al. Console di gestione AWS Proteggi le tue credenziali utente root nello stesso modo in cui proteggeresti altre informazioni personali sensibili. Per comprendere meglio come proteggere e dimensionare i processi degli utenti root, consulta Le migliori pratiche per gli utenti root Account AWS.

Assegna le autorizzazioni con privilegi minimi

Quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Potresti iniziare con autorizzazioni generiche mentre esplori le autorizzazioni necessarie per il tuo carico di lavoro o il caso d'uso. Man mano che il tuo caso d'uso matura, puoi lavorare per ridurre le autorizzazioni concesse per lavorare con il privilegio minimo. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Politiche e autorizzazioni in AWS Identity and Access Management.

Inizia con AWS politiche gestite e passaggio alle autorizzazioni con privilegi minimi

Per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite di AWS che concedono autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Tieni presente che le policy AWS gestite potrebbero non concedere autorizzazioni con privilegi minimi per i tuoi casi d'uso specifici perché sono disponibili per l'uso da parte di tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo Policy gestite dal cliente specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta AWS policy gestite. Per ulteriori informazioni sulle politiche AWS gestite progettate per funzioni lavorative specifiche, vedere. AWS politiche gestite per le funzioni lavorative

Utilizzare IAM Access Analyzer per generare policy con privilegi minimi in base all'attività di accesso

Per concedere solo le autorizzazioni richieste per eseguire un'attività, puoi generare policy in funzione dell'attività di accesso che hai effettuato l'accesso in AWS CloudTrail. IAM Access Analyzer analizza i servizi e le azioni utilizzati dai tuoi ruoli IAM e quindi genera una policy dettagliata che puoi utilizzare. Dopo aver testato ogni policy generata, puoi distribuirla nell'ambiente di produzione. In questo modo si garantisce di concedere solo le autorizzazioni necessarie ai carichi di lavoro. Per ulteriori informazioni sulla generazione delle policy, consulta IAM Access Analyzer policy generation.

Esaminare e rimuovere regolarmente utenti, ruoli, autorizzazioni, criteri e credenziali inutilizzati

Potresti avere utenti, ruoli, autorizzazioni, policy o credenziali IAM che non servono più nel tuo Account AWS. IAM fornisce le ultime informazioni di accesso per aiutarti a identificare gli utenti, i ruoli, le autorizzazioni, le policy e le credenziali che non ti servono più in modo da poterli rimuovere. In questo modo puoi ridurre il numero di utenti, ruoli, autorizzazioni, criteri e credenziali da monitorare. È possibile utilizzare queste informazioni per perfezionare le policy IAM e aderire meglio al principio del privilegio minimo. Per ulteriori informazioni, consulta Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso

È possibile specificare le condizioni che stabiliscono che una dichiarazione di policy è attiva. In questo modo, è possibile concedere l'accesso ad azioni e risorse, ma solo se la richiesta di accesso soddisfa condizioni specifiche. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando TLS. È inoltre possibile utilizzare le condizioni per concedere l'accesso alle azioni di servizio, ma solo se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per ulteriori informazioni, consulta Elementi della policy JSON IAM: Condizione.

Verifica dell'accesso multi-account e pubblico alle risorse con IAM Access Analyzer

Prima di concedere le autorizzazioni per l'accesso pubblico o con più account AWS, ti consigliamo di verificare se tale accesso è richiesto. Puoi utilizzare IAM Access Analyzer per visualizzare in anteprima e analizzare l'accesso multi-account e pubblico per i tipi di risorse supportati. Puoi farlo esaminando i risultati generati da IAM Access Analyzer. Questi risultati consentono di verificare che i controlli di accesso alle risorse garantiscano l'accesso previsto. Inoltre, quando aggiorni le autorizzazioni pubbliche e multi-account, puoi verificare l'effetto delle modifiche prima di distribuire nuovi controlli di accesso alle tue risorse. Inoltre, IAM Access Analyzer monitora continuamente i tipi di risorse supportati e genera un risultato per le risorse che consentono l'accesso multi-account o pubblico. Per ulteriori informazioni, consulta Anteprima dell'accesso con API IAM Access Analyzer.

Usa IAM Access Analyzer per convalidare le tue policy IAM per ottenere autorizzazioni sicure e funzionali

Convalida le policy che crei per assicurarti che aderiscano al linguaggio delle policy IAM (JSON) e alle best practice IAM. È possibile convalidare le policy utilizzando la validazione delle policy di IAM Access Analyzer. IAM Access Analyzer fornisce oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Durante la creazione di nuove policy o la modifica di policy esistenti nella console, IAM Access Analyzer fornisce suggerimenti per aiutarti a perfezionare e convalidare le policy prima di salvarle. Inoltre, consigliamo di rivedere e convalidare tutte le policy esistenti. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer. Per ulteriori informazioni sui controlli delle policy di IAM Access Analyzer, consulta Documentazione di riferimento sui controlli delle policy di IAM Access Analyzer.

Stabilisci guardrail delle autorizzazioni su più account

Man mano che scalate i carichi di lavoro, separateli utilizzando più account gestiti con. AWS Organizations Ti consigliamo di utilizzare le policy di controllo dei AWS Organizations servizi (SCP) per stabilire barriere di autorizzazioni per controllare l'accesso di tutti i principali (ruoli e utenti IAM) nei tuoi account. Ti consigliamo di utilizzare le policy di controllo delle risorse (RCP) AWS Organizations per stabilire i guardrail delle autorizzazioni per controllare l'accesso alle risorse AWS in tutta l'organizzazione. Gli SCP e gli RCP sono tipi di policy organizzative che puoi utilizzare per gestire le autorizzazioni nell'organizzazione a livello di AWS organizzazione, unità organizzativa (OU) o account.

Tuttavia, le SCP e le RCP da sole non sono sufficienti a concedere le autorizzazioni a principali e risorse nella tua organizzazione. Nessuna autorizzazione viene concessa dalle SCP e dalle RCP. Per concedere le autorizzazioni, è necessario collegare le policy basate su identità o le policy basate su risorse agli utenti o ai ruoli IAM o alle risorse degli account. Per ulteriori informazioni, consulta SRA Building Blocks — AWS Organizations, accounts and guardrails.

Utilizzare i limiti delle autorizzazioni per delegare la gestione delle autorizzazioni all'interno di un account

In alcuni scenari, è possibile che tu intenda delegare la gestione delle autorizzazioni all'interno di un account ad altri. Ad esempio, potresti consentire agli sviluppatori di creare e gestire ruoli per i loro carichi di lavoro. Quando deleghi le autorizzazioni ad altri, usa Limiti delle autorizzazioni per impostare le autorizzazioni massime delegate. Un limite delle autorizzazioni è una funzione

avanzata per l'utilizzo di una policy gestita per impostare il numero massimo di autorizzazioni che una policy basata su identità può concedere a un ruolo IAM. Il limite delle autorizzazioni non concedere l'accesso di per sé. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM.

Le migliori pratiche per gli utenti root Account AWS

Quando ne crei per la prima volta Account AWS, inizi con un set predefinito di credenziali con accesso completo a tutte le AWS risorse del tuo account. Questa identità è chiamata utente root di Account AWS. Ti consigliamo vivamente di non accedere all'utente Account AWS root a meno che tu non abbia un'attività che richiede le credenziali dell'utente root. È necessario proteggere le credenziali dell'utente root e i meccanismi di ripristino dell'account per evitare che le credenziali altamente privilegiate vengano esposte a usi non autorizzati.

In caso di Account AWS gestione multipla AWS Organizations, consigliamo di rimuovere le credenziali dell'utente root dagli account dei membri per prevenire l'uso non autorizzato. Puoi rimuovere la password dell'utente root, le chiavi di accesso, i certificati per la firma e disattivare ed eliminare l'autenticazione a più fattori (MFA). Gli account dei membri non possono accedere al proprio utente root o eseguirne il recupero della password. Per ulteriori informazioni, consulta Gestire centralmente l'accesso root per gli account membri.

Invece di accedere all'utente root, crea un utente amministrativo per le attività quotidiane.

- Se ne hai una nuova Account AWS, consulta. Configurare il tuo Account AWS
- Per una Account AWS gestione multipla AWS Organizations, consulta Configurare Account AWS l'accesso per un utente amministrativo di IAM Identity Center.

Con il tuo utente amministrativo, puoi quindi creare identità aggiuntive per gli utenti che necessitano di accedere alle risorse del tuo Account AWS. Ti consigliamo vivamente di richiedere agli utenti di autenticarsi con credenziali temporanee al momento dell'accesso. AWS

- Per uso singolo e autonomo Account AWS, utilizzalo Ruoli IAM per creare identità nel tuo account con autorizzazioni specifiche. I ruoli sono destinai a essere assunti da chiunque ne abbia bisogno. Inoltre, un ruolo non ha credenziali a lungo termine standard associate (password o chiavi di accesso). Invece, quando assumi un ruolo, ti fornisce credenziali di sicurezza temporanee per la tua sessione di ruolo. A differenza dei ruoli IAM, Utenti IAM dispongono di credenziali a lungo termine come password e chiavi di accesso. Ove possibile, le best practice raccomandano di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso.

- Per più utenti Account AWS gestiti tramite AWS Organizations, utilizza la forza lavoro di IAM Identity Center. Con IAM Identity Center, puoi gestire centralmente gli utenti di tutti i tuoi account Account AWS e le autorizzazioni all'interno di tali account. Gestisci le identità degli utenti con IAM Identity Center o con un provider di identità esterno. Per ulteriori informazioni, consulta Che cos'è Centro identità AWS IAM? nella Guida per l'utente di Centro identità AWS IAM .

Argomenti

- Proteggi le credenziali di utente root per impedirne l'uso non autorizzato
- Utilizza una password dell'utente root sicura per proteggere l'accesso
- Abilita l'autenticazione a più fattori (MFA) per la sicurezza dell'utente root
- Non creare chiavi di accesso per l'utente root
- Utilizza l'approvazione di più persone per l'accesso come utente root laddove possibile
- Usa un indirizzo email di gruppo per le credenziali dell'utente root
- Limita l'accesso ai meccanismi di recupero dell'account
- Proteggi i tuoi AWS Organizations credenziali utente root dell'account
- Monitora l'accesso e l'utilizzo

Proteggi le credenziali di utente root per impedirne l'uso non autorizzato

Proteggi le credenziali dell'utente root e usale solo per le attività che le richiedono. Per prevenire l'uso non autorizzato, non condividere la password dell'utente root, l'MFA, le chiavi di accesso, le coppie di chiavi o i certificati di firma con nessuno, ad eccezione di coloro che hanno una necessità aziendale rigorosa di accedere all'utente root. CloudFront

Non memorizzate la password dell'utente root con strumenti che dipendono da un account Servizi AWS a cui si accede utilizzando la stessa password. Se perdi o dimentichi la password dell'utente root, non potrai accedere a questi strumenti. Si consiglia di dare priorità alla resilienza e di richiedere a due o più persone di autorizzare l'accesso alla posizione di archiviazione. L'accesso alla password o alla sua posizione di archiviazione deve essere registrato e monitorato.

Utilizza una password dell'utente root sicura per proteggere l'accesso

Consigliamo di utilizzare una password complessa e univoca. Strumenti come i gestori di password con potenti algoritmi di generazione di password possono aiutarti a raggiungere questi obiettivi. AWS richiede che la password soddisfi le seguenti condizioni:

- Deve avere un minimo di 8 caratteri e un massimo di 128 caratteri.
- Deve includere almeno tre dei seguenti tipi di caratteri: maiuscole, minuscole, numeri e i simboli ! @ # $ % ^ & * () <> [] {} | _+-=.
- Non deve essere identica al tuo Account AWS nome o indirizzo email.

Per ulteriori informazioni, consulta Cambia la password per Utente root dell'account AWS.

Abilita l'autenticazione a più fattori (MFA) per la sicurezza dell'utente root

Poiché un utente root può eseguire azioni privilegiate, è fondamentale aggiungere MFA per l'utente root come secondo fattore di autenticazione oltre all'indirizzo e-mail e alla password come credenziali di accesso. È possibile registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi MFA attualmente supportati con l'utente Account AWS root.

Ti consigliamo vivamente di abilitare più dispositivi MFA per le credenziali dell'utente root per fornire maggiore flessibilità e resilienza nella tua strategia di sicurezza. Tutti i tipi di Account AWS (account standalone, di gestione e account membro) richiedono la configurazione dell'MFA per l'utente root. Se l'MFA non è ancora abilitata, gli utenti devono registrare l'MFA entro 35 giorni dal primo tentativo di accesso per accedere alla Console di gestione AWS .

- Le chiavi di sicurezza hardware certificate FIDO sono fornite da fornitori terzi. Per ulteriori informazioni, consulta Abilitare una chiave di sicurezza FIDO per l' Account AWS utente root.
- Token TOTP hardware: un dispositivo hardware che genera un codice numerico a sei cifre basato sull'algoritmo TOTP (password monouso). Per ulteriori informazioni, vedere Abilitare un token TOTP hardware per l'utente Account AWS root.
- Un'applicazione di autenticazione virtuale che viene eseguita su un telefono o altro dispositivo e simula un dispositivo fisico. Per ulteriori informazioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root.

Non creare chiavi di accesso per l'utente root

Le chiavi di accesso consentono di eseguire comandi nell'interfaccia a riga di AWS comando (AWS CLI) o di utilizzare le operazioni API da uno degli AWS SDK. Ti consigliamo vivamente di non creare chiavi di accesso per l'utente root perché l'utente root ha pieno accesso a tutte Servizi AWS le risorse dell'account, incluse le informazioni di fatturazione.

Poiché solo alcune attività richiedono l'utente root e in genere queste attività vengono eseguite raramente, si consiglia di accedere per Console di gestione AWS eseguire le attività dell'utente root. Per l'accesso programmatico, invece di creare chiavi di accesso per l'utente root, utilizza il `aws login` comando con le tue credenziali root per autenticarti negli SDK e negli SDK. AWS CLI Questo approccio fornisce credenziali temporanee e ruotate automaticamente che migliorano il livello di sicurezza eliminando la necessità di gestire le chiavi di accesso a lungo termine. Prima di creare le chiavi di accesso, esamina la Alternative alle chiavi di accesso a lungo termine.

Utilizza l'approvazione di più persone per l'accesso come utente root laddove possibile

Prendi in considerazione l'utilizzo dell'approvazione da parte di più persone per assicurarti che nessuna persona possa accedere sia all'MFA che alla password dell'utente root. Alcune aziende aggiungono un ulteriore livello di sicurezza configurando un gruppo di amministratori con accesso alla password e un altro gruppo di amministratori con accesso alla MFA. Per eseguire l'accesso utilizzando le credenziali dell'utente root è necessario che si riuniscano due membri, uno di ciascun gruppo.

Usa un indirizzo email di gruppo per le credenziali dell'utente root

Utilizza un indirizzo e-mail gestito dalla tua azienda e inoltra i messaggi ricevuti direttamente a un gruppo di utenti. Se AWS devi contattare il proprietario dell'account, questo approccio riduce il rischio di ritardi nella risposta, anche se le persone sono in vacanza, sono ammalate o hanno lasciato l'attività. L'indirizzo e-mail utilizzato per l'utente root non deve essere utilizzato per altri scopi.

Limita l'accesso ai meccanismi di recupero dell'account

Assicurati di sviluppare un processo per gestire i meccanismi di recupero delle credenziali degli utenti root nel caso in cui sia necessario accedervi durante un'emergenza, ad esempio l'acquisizione del tuo account amministrativo.

- Assicurati di avere accesso alla casella di posta elettronica dell'utente root in modo da poter reimpostare una password dell'utente root persa o dimenticata.
- Se l'MFA per l'utente Account AWS root viene perso, danneggiato o non funziona, puoi accedere utilizzando un altro MFA registrato con le stesse credenziali dell'utente root. Se hai perso l'accesso a tutte le MFA, hai bisogno che sia il numero di telefono che l'e-mail siano aggiornati e accessibili per ripristinare l'MFA. Per i dettagli, consulta Recupero di un dispositivo MFA per utenti root.

- Se scegli di non memorizzare la password dell'utente root e l'MFA, il numero di telefono registrato nell'account può essere utilizzato come metodo alternativo per recuperare le credenziali dell'utente root. Assicurati di avere accesso al numero di telefono di contatto, mantieni il numero di telefono aggiornato e limita gli utenti che possono accedere alla gestione del numero di telefono.

Nessuno dovrebbe avere accesso sia alla casella di posta elettronica che al numero di telefono, poiché entrambi sono canali di verifica per recuperare la password dell'utente root. È importante che due gruppi di persone gestiscano questi canali. Un gruppo ha accesso al tuo indirizzo email principale e un altro gruppo che ha accesso al numero di telefono principale per recuperare l'accesso al tuo account come utente root.

Proteggi i tuoi AWS Organizations credenziali utente root dell'account

Passando a una strategia con più account con AWS Organizations, ognuno di voi Account AWS ha le proprie credenziali di utente root che è necessario proteggere. L'account che usi per creare la tua organizzazione è l'account di gestione e gli altri account dell'organizzazione sono account membro.

Protezione delle credenziali dell'utente root per l'account di gestione

AWS richiede la registrazione dell'MFA per l'utente root dell'account di gestione dell'organizzazione. La registrazione MFA deve essere completata durante il primo tentativo di accesso o entro il periodo di prova di 35 giorni. Se la MFA non è abilitata entro questo periodo, sarà richiesta la registrazione prima di poter accedere alla Console di gestione AWS. Per ulteriori informazioni, consulta Multi-factor autenticazione per Utente root dell'account AWS.

Proteggi le credenziali utente root per gli account dei membri

Se lo utilizzi AWS Organizations per gestire più account, ci sono due strategie che puoi adottare per proteggere l'accesso degli utenti root al tuo AWS Organizations.

- Centralizza l'accesso root e rimuovi le credenziali dell'utente root dagli account membro. Rimuovi le credenziali dell'utente root, le chiavi di accesso, i certificati per la firma e disattiva ed elimina l'autenticazione a più fattori (MFA). Quando si impiega questa strategia, gli account dei membri non possono accedere al proprio utente root o eseguirne il recupero della password. Per ulteriori informazioni, consulta Gestire centralmente l'accesso root per gli account membri.
- Proteggi le credenziali utente root dei tuoi AWS Organizations account con l'autenticazione a più fattori per migliorare la sicurezza degli account. Per ulteriori informazioni, consulta Multi-factor autenticazione per Utente root dell'account AWS.

Per i dettagli, consulta Accessing member accounts in your organization nella Guida per l'utente di AWS Organizations .

Imposta i controlli di sicurezza preventivi in AWS Organizations utilizzando una politica di controllo del servizio (SCP)

Se gli account membro della tua organizzazione hanno le credenziali utente root abilitate, puoi applicare una SCP per limitare l'accesso all'utente root dell'account membro. Negare tutte le azioni degli utenti root negli account dei membri, ad eccezione di alcune azioni di tipo root, aiuta a prevenire l'accesso non autorizzato. Per i dettagli, consulta utilizza una SCP per limitare ciò che le operazioni che un utente root nei tuoi account membri può eseguire.

Monitora l'accesso e l'utilizzo

Ti consigliamo di utilizzare gli attuali meccanismi di tracciamento per monitorare, avvisare e segnalare l'accesso e l'uso delle credenziali dell'utente root, compresi gli avvisi che annunciano l'accesso e l'utilizzo dell'utente root. I seguenti servizi possono aiutare a verificare che l'utilizzo delle credenziali dell'utente root venga monitorato ed eseguire controlli di sicurezza che possono aiutare a prevenire l'uso non autorizzato.

> **Note**
>
> CloudTrail registra diversi eventi di accesso per l'utente root e le sessioni dell'utente root privilegiato. Queste sessioni con privilegi consentono di eseguire attività che richiedono credenziali dell'utente root negli account dei membri dell'organizzazione. È possibile utilizzare l'evento di accesso per identificare le azioni intraprese dall'account di gestione o da un amministratore delegato utilizzando `sts:AssumeRoot`. Per ulteriori informazioni, consulta Tenere traccia delle attività con privilegi in CloudTrail.

- Se desideri ricevere una notifica sull'attività di accesso dell'utente root nel tuo account, puoi sfruttare Amazon CloudWatch per creare una regola Events che rileva quando vengono utilizzate le credenziali dell'utente root e attiva una notifica all'amministratore della sicurezza. Per i dettagli, consulta Monitorare e notificare l'attività dell'utente root. Account AWS
- Se desideri impostare notifiche per avvisarti delle azioni approvate dall'utente root, puoi utilizzare Amazon EventBridge insieme ad Amazon SNS per scrivere una EventBridge regola per tracciare l'utilizzo dell'utente root per l'azione specifica e inviarti una notifica utilizzando un argomento di

Amazon SNS. Per un esempio, consulta Inviare una notifica quando viene creato un oggetto Amazon S3.

- Se lo utilizzi già GuardDuty come servizio di rilevamento delle minacce, puoi estenderne la capacità per notificarti quando le credenziali dell'utente root vengono utilizzate nel tuo account.

Gli avvisi dovrebbero includere, ma non esclusivamente, l'indirizzo e-mail utilizzato per l'utente root stesso. Controlla che ci siano delle prassi attive perché il personale che riceve un avviso di questo tipo comprenda come convalidare che è previsto l'accesso utente root e come sottoporre la questione ai livelli gerarchici superiori se ritiene che sia in corso un incidente di sicurezza. Per un esempio di come configurare gli avvisi, consulta Monitorare e notificare l'attività dell'utente Account AWS root.

Valuta la conformità con l'MFA per l'utente root

I seguenti servizi consentono di valutare la conformità MFA per le credenziali dell'utente root.

> MFA-related le regole vengono restituite non conformi se si segue la procedura consigliata di rimuovere le credenziali dell'utente root.
>
> Consigliamo di rimuovere le credenziali dell'utente root dagli account membri dell'organizzazione per evitare un uso non autorizzato. Dopo aver rimosso le credenziali dell'utente root, inclusa la MFA, questi account membro vengono valutati come non applicabili.

- AWS Config fornisce regole per monitorare la conformità alle best practice per l'utente root. È possibile utilizzare regole AWS Config gestite per applicare l'MFA per le credenziali dell'utente root. AWS Config può anche identificare le chiavi di accesso per l'utente root.
- Security Hub CSPM offre una visione completa dello stato della sicurezza AWS e aiuta a valutare AWS l'ambiente rispetto agli standard e alle best practice del settore della sicurezza, ad esempio l'utilizzo dell'MFA per l'utente root e la mancanza di chiavi di accesso per l'utente root. Per informazioni dettagliate sulle regole disponibili, consulta i AWS Identity and Access Management controlli nella Guida per l'utente di Security Hub CSPM.
- Trusted Advisor fornisce un controllo di sicurezza in modo da sapere se l'MFA non è abilitato sull'account utente root. Per ulteriori informazioni, consulta MFA sull'account root nella Guida per l'utente di AWS .

Se devi segnalare un problema di sicurezza sul tuo account, consulta Segnalazione di e-mail sospette o Segnalazione di vulnerabilità. In alternativa, puoi contattare AWS per ricevere assistenza e indicazioni aggiuntive.

Casi d'uso di business per IAM

Un semplice caso d'uso aziendale per IAM può aiutarti a comprendere i modi di base in cui potresti implementare il servizio per controllare l' AWS accesso dei tuoi utenti. Il caso d'uso viene descritto in termini generali, senza i meccanismi del modo in cui si desidera utilizzare l'API IAM per ottenere i risultati desiderati.

Questo caso d'uso esamina due modi tipici in cui un'azienda fittizia chiamata Example Corp può utilizzare IAM. Il primo scenario considera Amazon Elastic Compute Cloud (Amazon EC2). Il secondo considera Amazon Simple Storage Service (Amazon S3).

Per ulteriori informazioni sull'utilizzo di IAM con altri servizi di AWS, consultaAWS servizi supportati da IAM.

Argomenti

- Configurazione iniziale di Example Corp
- Caso d'uso per IAM con Amazon EC2
- Caso d'uso per IAM con Amazon S3

Configurazione iniziale di Example Corp

Nikki Wolf e Mateo Jackson sono i fondatori di Example Corp. Dopo aver avviato l'azienda, creano Account AWS e configurano Centro identità AWS IAM(IAM Identity Center) per creare account amministrativi da utilizzare con le loro risorse. AWS Quando si configura l'accesso all'account per l'utente amministrativo, il Centro identità IAM crea un ruolo IAM corrispondente. Questo ruolo, controllato da IAM Identity Center, viene creato nell'area pertinente Account AWS e le politiche specificate nel set di AdministratorAccess autorizzazioni sono allegate al ruolo.

Poiché ora dispongono di account di amministratore, Nikki e Mateo non devono più utilizzare il proprio utente root per accedere all' Account AWS. Pianificano l'uso dell'utente root solo per completare le attività che possono essere eseguite soltanto dall'utente root. Dopo aver esaminato le best practice di sicurezza, configurano l'autenticazione a più fattori MFA per le credenziali dell'utente root e decidono come proteggere tali credenziali.

Man mano che l'azienda cresce, assume dipendenti che lavorano come sviluppatori, amministratori, tester, manager e amministratori di sistema. Nikki è responsabile delle operazioni, mentre Mateo gestisce i team di ingegneria. Hanno creato un server di dominio Active Directory per gestire gli account dei dipendenti e gestire l'accesso alle risorse interne dell'azienda.

Per consentire ai dipendenti di accedere alle AWS risorse, utilizzano IAM Identity Center per connettere l'Active Directory della propria azienda al proprio Account AWS.

Poiché hanno collegato Active Directory al Centro identità IAM, gli utenti, il gruppo e l'appartenenza al gruppo vengono sincronizzati e definiti. Devono assegnare set di autorizzazioni e ruoli ai diversi gruppi per fornire agli utenti il corretto livello di accesso alle AWS risorse. Utilizzano AWS politiche gestite per le funzioni lavorative in Console di gestione AWS per creare questi set di autorizzazioni:

- Amministratore
- Fatturazione
- Sviluppatori
- Amministratori di rete
- Amministratori di database
- Amministratori di sistema
- Utenti del gruppo Supporto

Quindi assegnano i set di autorizzazioni ai ruoli assegnati ai rispettivi gruppi di Active Directory.

Per una guida dettagliata che descrive la configurazione iniziale del Centro identità IAM, consulta Nozioni di base nella Guida per l'utente di Centro identità AWS IAM . Per ulteriori informazioni sul provisioning dell'accesso utente del Centro identità IAM, consulta Accesso Single Sign-on agli account AWS nella Guida per l'utente di Centro identità AWS IAM .

Caso d'uso per IAM con Amazon EC2

Un'azienda come Example Corp normalmente utilizza IAM per interagire con servizi come Amazon EC2. Per capire questa parte del caso d'uso, è necessaria una conoscenza di base di Amazon EC2. Per ulteriori informazioni su Amazon EC2, consulta la Amazon EC2 User Guide.

Autorizzazioni Amazon EC2 per i gruppi di utenti

Per fornire il controllo «perimetrale», Nikki allega una policy al gruppo di utenti. AllUsers Questa politica rifiuta qualsiasi AWS richiesta da parte di un utente se l'indirizzo IP di origine è esterno alla rete aziendale di Example Corp.

Presso Example Corp, diversi gruppi IAM richiedono autorizzazioni diverse:

- Amministratori di sistema: è necessaria l'autorizzazione per creare e gestire le AMI, le istanze, gli snapshot, i volumi, i gruppi di sicurezza e così via. Nikki allega la policy `AmazonEC2FullAccess` AWS gestita al gruppo di SysAdmins utenti che concede ai membri del gruppo il permesso di utilizzare tutte le azioni di Amazon EC2.
- Sviluppatori: è necessaria la possibilità di collaborare solo con le istanze. Mateo quindi crea e collega una policy al gruppo di utenti Sviluppatori che consente agli sviluppatori di chiamare `DescribeInstances`, `RunInstances`, `StopInstances`, `StartInstances` e `TerminateInstances`.

  > **Note**
  >
  > Amazon EC2 utilizza chiavi SSH, password Windows e gruppi di sicurezza per controllare chi ha accesso al sistema operativo di istanze Amazon EC2 specifiche. Non esiste un metodo nel sistema IAM per consentire o rifiutare l'accesso al sistema operativo di una determinata istanza.

- Utenti del gruppo Supporto: non devono essere in grado di eseguire operazioni Amazon EC2 eccetto elencare risorse Amazon EC2 correntemente disponibili. Pertanto, Nikki crea e collega una policy al gruppo di utenti Supporto che consente di chiamare solo le operazioni API "Describe" di Amazon EC2.

Per esempi di come potrebbero apparire queste policy, consulta la sezione Using AWS Identity Esempi di policy basate su identità IAM and Access Management nella Amazon EC2 User Guide.

Modifica della funzione lavorativa dell'utente

A un certo punto, uno degli sviluppatori, Paulo Santos, cambia le funzioni lavorative e diventa un responsabile. In qualità di responsabile, Paulo entra a far parte del gruppo di utenti Supporto in modo da poter aprire casi di supporto per i suoi sviluppatori. Mateo sposta Paulo dal gruppo di utenti Sviluppatori al gruppo di utenti Supporto. Come risultato di questa mossa, la sua possibilità di

interagire con le istanze Amazon EC2 è limitata. Non può avviare istanze. Inoltre, non può arrestare o terminare le istanze esistenti, anche se era l'utente che ha avviato l'istanza. Può elencare solo le istanze che gli utenti di Example Corp hanno lanciato.

Caso d'uso per IAM con Amazon S3

Le aziende come Example Corp in genere utilizzano IAM anche con Amazon S3. John ha creato un bucket Amazon S3 per l'azienda chiamato amzn-s3-demo-bucket.

Creazione di altri utenti e gruppi di utenti

Come dipendenti, Zhang Wei e Mary Major devono essere in grado di creare i propri dati nel bucket dell'azienda. Devono anche leggere e scrivere dati condivisi sui quali lavorano tutti gli sviluppatori. Per farlo, Mateo dispone logicamente i dati in amzn-s3-demo-bucket utilizzando uno schema di prefisso della chiave Amazon S3 come illustrato nella seguente figura.

```
/amzn-s3-demo-bucket
    /home
        /zhang
        /major
    /share
        /developers
        /managers
```

Mateo divide il `/amzn-s3-demo-bucket` in un set di directory principali per ogni dipendente e un'area condivisa per gruppi di sviluppatori e responsabili.

A questo punto Mateo crea un set di policy per assegnare le autorizzazioni agli utenti e ai gruppi di utenti:

- Accesso alla directory principale per Zhang: Mateo collega una policy a Wei che gli permette di leggere, scrivere ed elencare tutti gli oggetti con il prefisso della chiave Amazon S3 `/amzn-s3-demo-bucket/home/zhang/`
- Accesso alla directory principale per Major: Mateo collega una policy a Mary che le permette di leggere, scrivere ed elencare tutti gli oggetti con il prefisso della chiave Amazon S3 `/amzn-s3-demo-bucket/home/major/`
- Accesso alla directory condivisa per il gruppo di utenti Sviluppatori: Mateo collega una policy al gruppo di utenti che consente agli sviluppatori di leggere, scrivere ed elencare qualsiasi oggetto in `/amzn-s3-demo-bucket/share/developers/`

- Accesso alla directory condivisa per il gruppo di utenti Responsabili: Mateo collega una policy al gruppo di utenti che consente ai responsabili di leggere, scrivere ed elencare qualsiasi oggetto in `/amzn-s3-demo-bucket/share/managers/`

> **Note**
>
> Amazon S3 non fornisce automaticamente l'autorizzazione a un utente che crea un bucket o un oggetto di eseguire altre operazioni su quell'oggetto o bucket. Pertanto, nelle policy IAM è necessario fornire esplicitamente agli utenti l'autorizzazione per utilizzare le risorse Amazon S3 che creano.

Per esempi della probabile struttura di queste policy, consulta Controllo accessi nella Guida per l'utente di Amazon Simple Storage Service. Per informazioni su come le policy vengono valutate in fase di runtime, consulta Logica di valutazione delle policy.

Modifica della funzione lavorativa dell'utente

A un certo punto, uno degli sviluppatori, Zhang Wei, cambia le funzioni lavorative e diventa un responsabile. Si presuppone che non abbia più bisogno di accedere ai documenti nella directory `share/developers`. Mateo, come amministratore, sposta Wei nel gruppo di utenti `Managers` e lo rimuove dal gruppo `Developers`. Con quella semplice riassegnazione, Wei ottiene automaticamente tutte le autorizzazioni concesse al gruppo di utenti `Managers`, ma non è più in grado di accedere a dati nella directory `share/developers`.

Integrazione con un business di terze parti

Le organizzazioni spesso lavorano con aziende partner, consulenti e appaltatori. Example Corp ha un partner che si chiama Widget Company e un dipendente di Widget Company che si chiama Shirley Rodriguez deve inserire dati in un bucket perché siano utilizzati da Example Corp. Nikki crea un gruppo di utenti chiamato WidgetCo e un utente denominato `Shirley` e aggiunge Shirley al gruppo di utenti. WidgetCo Nikki crea anche un bucket speciale per Shirley chiamato amzn-s3-demo-bucket1.

Nikki aggiorna le policy esistenti o ne aggiunge di nuove per aiutare l'azienda partner Widget Company. Ad esempio, Nikki può creare una nuova politica che nega ai membri del gruppo di WidgetCo utenti la possibilità di utilizzare qualsiasi azione diversa dalla scrittura. Questa policy

è necessaria solo se è presente una policy ampia che offre a tutti gli utenti l'accesso a un'ampia gamma di operazioni Amazon S3.

Tutorial IAM

I seguenti tutorial presentano end-to-end procedure complete per le attività comuni per AWS Identity and Access Management (IAM). Gli scenari presentati sono solo esempi con nomi di società e utenti fittizi destinati a essere usati in un ambiente di laboratorio. Il loro scopo è di fornire linee guida di carattere generico. Non devono essere utilizzati direttamente nell'ambiente di produzione, senza un'accurata opera di revisione e adattamento alle necessità esclusive del tuo ambiente lavorativo.

Esercitazioni

- Tutorial IAM: Delega l'accesso attraverso AWS account che utilizzano ruoli IAM
- Tutorial IAM: Creazione e collegamento della prima policy gestita dal cliente
- Tutorial IAM: Definizione delle autorizzazioni di accesso AWS risorse basate sui tag
- Tutorial IAM: consentire agli utenti di gestire le proprie credenziali e impostazioni MFA
- Tutorial IAM: usa un CloudFormation modello per creare un SAML Identity Provider (IdP)
- Tutorial IAM: usa un CloudFormation modello per creare un ruolo IAM federato SAML
- Tutorial IAM: usa un CloudFormation modello per creare un ruolo IAM federato SAML Identity Provider (IdP) e SAML

Tutorial IAM: Delega l'accesso attraverso AWS account che utilizzano ruoli IAM

> **Important**
>
> Le best practice IAM consigliano di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee invece di utilizzare utenti IAM con credenziali a lungo termine. Ti consigliamo di utilizzare gli utenti IAM solo per casi d'uso specifici non supportati dagli utenti federati.

In questo tutorial viene descritto come utilizzare un ruolo per delegare l'accesso a risorse che si trovano in diversi Account AWS di tua proprietà denominati Destination e Originating. Puoi condividere le risorse di un account con gli utenti di un altro account. Configurando in questo modo l'accesso fra account, non dovrai creare singoli utenti IAM per ogni account. Inoltre, gli utenti non devono uscire da un account e accedere a un altro per utilizzare risorse in Account AWS diversi.

Dopo aver configurato il ruolo, puoi vedere come utilizzarlo tramite Console di gestione AWS, AWS CLI, e l'API.

In questo tutorial, l'account Destination gestisce i dati delle applicazioni a cui accedono diverse applicazioni e team. In ciascun account puoi archiviare le informazioni sull'applicazione in bucket Amazon S3. Gestisci gli utenti IAM nell'account Originating, dove hai due ruoli utente IAM: sviluppatori e analisti. Gli sviluppatori e gli analisti utilizzano l'account Originating per generare dati condivisi da più microservizi. Entrambi i ruoli dispongono delle autorizzazioni per lavorare nell'account Originating e accedere alle sue risorse. Occasionalmente, uno sviluppatore deve aggiornare i dati condivisi nell'account Destination. Gli sviluppatori archiviano questi dati in un bucket Amazon S3 denominato `amzn-s3-demo-bucket-shared-container`.

Alla fine di questo tutorial, si dispone di quanto segue:

- Utenti nell'account Originating (l'account attendibile) che possono assumere un ruolo specifico nell'account Destination.
- Un ruolo nell'account Destination (l'account attendibile) che può accedere a uno specifico bucket Amazon S3.
- Il bucket `amzn-s3-demo-bucket-shared-container` nell'account Destination.

Gli sviluppatori possono utilizzare il ruolo in Console di gestione AWS per accedere al `amzn-s3-demo-bucket-shared-container` bucket nell'account di destinazione. Inoltre, possono accedere al bucket utilizzando chiamate API autenticate tramite credenziali provvisorie fornite dal ruolo. La stessa operazione eseguita da un analista avrà esito negativo.

Questo flusso di lavoro ha tre fasi di base:

Creare un ruolo dell'account Destination

Innanzitutto, si utilizza Console di gestione AWS per stabilire un rapporto di fiducia tra l'account di destinazione (numero ID) e l'account di origine (numero ID). Si inizia creando un ruolo IAM denominato. UpdateData Quando crei il ruolo, devi definire l'account Originating come entità attendibile e specificare una policy di autorizzazioni che consenta agli utenti attendibili di aggiornare il bucket `amzn-s3-demo-bucket-shared-container`.

Concedi autorizzazione per l'accesso al ruolo

In questa sezione, puoi modificare la policy del ruolo per negare l'accesso al ruolo `UpdateData` agli analisti. Perché gli analisti hanno PowerUser accesso a questo scenario e tu devi negare esplicitamente la possibilità di utilizzare il ruolo.

Accesso al test tramite cambio di ruoli

Infine, in qualità di sviluppatore utilizzerai il ruolo `UpdateData` per aggiornare il bucket `amzn-s3-demo-bucket-shared-container` nell'account Destination. Puoi vedere come accedere al ruolo tramite la AWS console, l'API e l' AWS CLI API.

Considerazioni

Prima di utilizzare i ruoli IAM per delegare l'accesso alle risorse Account AWS, è importante considerare quanto segue:

- Non è possibile passare a un ruolo se l'accesso è stato effettuato come Utente root dell'account AWS.
- I ruoli IAM e le policy basate sulle risorse delegano l'accesso tra account solo all'interno di una singola partizione. Ad esempio, si supponga di disporre di un account nella regione Stati Uniti occidentali (California settentrionale) nella partizione `aws` standard. Hai anche un account nella regione Cina (Pechino) nella partizione `aws-cn`. Non è possibile utilizzare una policy basata sulle risorse Amazon S3 nel tuo account nella regione Cina (Pechino) per consentire l'accesso agli utenti del tuo account `aws` standard.
- Puoi usarlo Centro identità AWS IAM per facilitare il single sign-on (SSO) per esterni Account AWS (account esterni al tuo) utilizzando Security Assertion Markup Language (SAML AWS Organizations). Per informazioni dettagliate, consulta Integrazione di dati esterni Account AWSCentro identità AWS IAM per una gestione centralizzata degli accessi con fatturazione indipendente utilizzando SAML 2.0
- Puoi associare ruoli a AWS risorse come istanze o funzioni di Amazon EC2. AWS Lambda Per informazioni dettagliate, vedi Crea un ruolo per delegare le autorizzazioni a un AWS service.
- Se desideri che un'applicazione assuma un ruolo in un'altra Account AWS, puoi utilizzare l' AWS SDK per l'assunzione di ruoli tra account. Per ulteriori informazioni, consulta Autenticazione e accesso nella Guida di riferimento agli SDK AWS e agli strumenti.
- Il cambio di ruolo utilizzando il Console di gestione AWS solo funziona con gli account che non richiedono un. `ExternalId` Ad esempio, si supponga di concedere l'accesso al proprio account

a terzi e di richiedere un `ExternalId` in un elemento `Condition` nella policy di autorizzazione. In tal caso, la terza parte può accedere al tuo account solo utilizzando l' AWS API o uno strumento a riga di comando. Le terze parti non possono utilizzare la console perché non sono in grado di fornire un valore per `ExternalId`. Per ulteriori informazioni su questo scenarioAccesso a Account AWS di proprietà di terzi, consulta la sezione Come abilitare l'accesso tra più account Console di gestione AWS nel AWS Security Blog.

Prerequisiti

Questo tutorial presuppone che tu abbia a disposizione quanto segue:

- Due account separati Account AWS che puoi utilizzare, uno per rappresentare l'account di origine e uno per rappresentare l'account di destinazione.
- Utenti e gruppi nell'account Originating, creati e configurati in questo modo:

Mansione	Utente	Permissions
Developer	David	Entrambi gli utenti possono accedere e utilizzare l' Console di gestione AWS account di origine.
Analista	Jane	

- Non è necessario creare utenti nell'account Destination.
- Un bucket Amazon S3 creato nell'account Destination. Nel tutorial puoi chiamarlo `amzn-s3-demo-bucket-shared-container`, ma dato che i nomi dei bucket S3 devono essere univoci a livello globale, dovrai selezionare un nome diverso.

Creare un ruolo dell'account Destination

Puoi consentire agli utenti di uno di accedere Account AWS alle risorse di un altro Account AWS. In questo tutorial, ciò verrà effettuato creando un ruolo che definisca chi può accedervi e quali autorizzazioni concedere agli utenti che lo assumono.

In questo passaggio del tutorial, dovrai creare il ruolo nell'account Destination e impostare l'account Originating come entità attendibile. Inoltre, dovrai limitare le autorizzazioni del ruolo al solo accesso di lettura e scrittura per il bucket `amzn-s3-demo-bucket-shared-container`. Chiunque abbia ricevuto l'autorizzazione a usare il ruolo potrà leggere e scrivere nel bucket `shared-container`.

Prima di poter creare un ruolo, è necessario l'ID dell'account di Originating Account AWS. A ciascuno Account AWS è assegnato un identificatore ID account univoco.

Per ottenere l'Originating Account AWS ID

1. Accedi Console di gestione AWS come amministratore dell'account Originating e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nella console IAM, scegli il tuo nome utente nella barra di navigazione in alto a destra. Di solito ha il seguente aspetto: ***username*** @ ***account_ID_number_or_alias***.

   Per questo scenario, puoi utilizzare l'ID account 111111111111 per l'account Originating. Tuttavia, devi utilizzare un ID account valido se stai usando questo scenario nell'ambiente di test.

Per creare un ruolo nell'account Destination che possa essere utilizzato dall'account Originating

1. Accedi Console di gestione AWS come amministratore dell'account di destinazione e apri la console IAM.
2. Prima di creare il ruolo, prepara la policy gestita che definisce le autorizzazioni per i requisiti del ruolo. La policy verrà collegata al ruolo in una fase successiva.

   Impostare l'accesso in lettura e scrittura al bucket `amzn-s3-demo-bucket-shared-container`. Sebbene AWS fornisca alcune policy gestite da Amazon S3, non ce n'è una che fornisca l'accesso in lettura e scrittura a un singolo bucket Amazon S3. Se lo desideri, puoi creare una policy personalizzata.

   Nel pannello di navigazione, seleziona Policy e Crea policy.
3. Seleziona la scheda JSON e copia il testo dal documento della seguente policy JSON. Incollare il testo nella casella di testo JSON, sostituendo l'ARN della risorsa (`arn:aws:s3:::shared-container`) con quello per reale per il bucket Amazon S3.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
```

```
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
       ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-shared-container"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-shared-container/*"
    }
  ]
}
```

L'azione `ListAllMyBuckets` concede l'autorizzazione per elencare tutti i bucket di proprietà del mittente autenticato della richiesta. L'autorizzazione `ListBucket` consente agli utenti di visualizzare gli oggetti del bucket `amzn-s3-demo-bucket-shared-container`. Le autorizzazioni `GetObject`, `PutObject` e `DeleteObject` consentono agli utenti di visualizzare, aggiornare ed eliminare i contenuti del bucket `amzn-s3-demo-bucket-shared-container`.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

4. Nella pagina Verifica policy, digita **`read-write-app-bucket`** come nome della policy. Esamina le autorizzazioni concesse dalla policy, quindi scegli Crea policy per salvare il lavoro.

   La nuova policy viene inserita nell'elenco delle policy gestite.

5. Nel riquadro di navigazione, scegli Ruoli, quindi Crea ruolo.

6. Scegli il tipo di ruolo Un Account AWS.
7. Per ID account, digita l'ID dell'account Originating.

   Questo tutorial utilizza l'ID account di esempio **111111111111** per l'account di Originating. Tuttavia, è consigliabile utilizzare un ID account valido. Se utilizzi un ID account non valido, come ad esempio **111111111111**, IAM non ti consentirà di creare il nuovo ruolo.

   Per il momento non è necessario richiedere un ID esterno, né chiedere agli utenti un'autenticazione a più fattori (MFA) per assumere il ruolo. Tali opzioni possono rimanere deselezionate. Per ulteriori informazioni, consulta AWS Multi-factor autenticazione in IAM.
8. Selezionare Next:Permissions (Avanti:Autorizzazioni) per impostare le autorizzazioni associate al ruolo.
9. Seleziona la casella di controllo accanto alla policy creata in precedenza.

   > **Suggerimento**
   >
   > In Filter (Filtro) selezionare Customer managed (Gestite dal cliente) per visualizzare solo le policy create. Il filtro nasconde le policy create da AWS per semplificare la ricerca.

   Quindi, seleziona Successivo.
10. (Facoltativo) Aggiungi metadati al ruolo collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.
11. (Facoltativo) In Descrizione, inserisci una descrizione per il nuovo ruolo.
12. Dopo avere rivisto il ruolo, fare clic su Create Role (Crea ruolo).

    Il ruolo `UpdateData` viene visualizzato nell'elenco dei ruoli.

A questo punto, è necessario ottenere l'Amazon Resource Name (ARN), un identificatore univoco per il ruolo. Quando modifichi il ruolo dello sviluppatore nell'account Originating, specifica l'ARN del ruolo dell'account Destination per concedere o negare le autorizzazioni.

Per ottenere l'ARN per UpdateData

1. Nel pannello di navigazione della console IAM seleziona Ruoli.

2. Nell'elenco dei ruoli, selezionare il ruolo `UpdateData`.
3. Nella sezione Summary (Riepilogo) del riquadro dei dettagli, copiare il valore Role ARN (ARN ruolo).

   L'ID dell'account Destination è 999999999999, pertanto l'ARN del ruolo sarà `arn:aws:iam::999999999999:role/UpdateData`. Assicurati di fornire l' Account AWS ID reale per l'account di destinazione.

A questo punto, hai stabilito un rapporto di fiducia tra gli account Destination e Originating. È stato possibile creando un ruolo nell'account Destination che identifica l'account Originating come principale attendibile. Inoltre, hai definito le operazioni consentite agli utenti che passano al ruolo `UpdateData`.

Quindi, modifica le autorizzazioni per il ruolo da sviluppatore.

Concedi autorizzazione per l'accesso al ruolo

A questo punto, sia gli analisti che gli sviluppatori dispongono delle autorizzazioni che consentono di gestire i dati dell'account Originating. Usare i seguenti passaggi necessari per aggiungere le autorizzazioni per il cambio di ruolo.

Per modificare il ruolo Sviluppatori per consentire loro di passare al UpdateData ruolo

1. Accedi come amministratore nell'account Originating e apri la console IAM.
2. Seleziona Ruoli e quindi Developers.
3. Nella scheda Autorizzazioni, scegli Aggiungi autorizzazioni, quindi Crea policy in linea.
4. Scegli la scheda JSON.
5. Aggiungi la seguente istruzione di policy per consentire l'azione `AssumeRole` sul ruolo `UpdateData` nell'account Destination. Assicurati di modificare *`DESTINATION-ACCOUNT-ID`* l'`Resource`elemento con l' Account AWS ID effettivo dell'account di destinazione.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
```

```
        "Resource": "arn:aws:iam::111122223333:role/UpdateData"
    }
}
```

L'effetto `Allow` consente in modo esplicito al gruppo Sviluppatori l'accesso al ruolo `UpdateData` dell'account Destination. Qualsiasi sviluppatore potrà accedere al ruolo.

6. Scegliere Esamina policy.
7. Digita un nome, ad esempio **`allow-assume-S3-role-in-destination`**.
8. Scegli Crea policy.

Nella maggior parte degli ambienti, la procedura seguente risulta superflua. Se, tuttavia, utilizzi PowerUserAccess le autorizzazioni, alcuni gruppi potrebbero già essere in grado di cambiare ruolo. La procedura descritta di seguito mostra come aggiungere un'autorizzazione `"Deny"` al gruppo Analisti, per impedire ai suoi membri di assumere il ruolo. Se questa procedura non è necessaria nel tuo ambiente, è preferibile non aggiungerla. Le autorizzazioni `"Deny"` di tipo generale sono più complicate da gestire e comprendere. Utilizza le autorizzazioni `"Deny"` solo quando non sono disponibili alternative migliori.

Per modificare il ruolo Analisti e negare l'autorizzazione ad assumere tale ruolo **`UpdateData`**

1. Scegli Ruoli, quindi scegli Analisti.
2. Nella scheda Autorizzazioni, scegli Aggiungi autorizzazioni, quindi Crea policy in linea.
3. Scegli la scheda JSON.
4. Aggiungere la seguente istruzione di policy per negare l'operazione `AssumeRole` nel ruolo `UpdateData`. Assicurati di modificare l'`Resource`elemento *`DESTINATION-ACCOUNT-ID`* inserendo l' Account AWS ID effettivo dell'account di destinazione.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "sts:AssumeRole",
        "Resource": "arn:aws:iam::111122223333:role/UpdateData"
    }
```

```
}
```

L'effetto `Deny` impedisce in modo esplicito al gruppo Analisi l'accesso al ruolo `UpdateData` dell'account Destination. Se un analista prova ad accedere al ruolo riceve un messaggio di accesso negato.

5. Scegliere Esamina policy.
6. Digita un nome come **`deny-assume-S3-role-in-destination`**.
7. Scegli Crea policy.

Il gruppo Sviluppatori ora dispone delle autorizzazioni per utilizzare il ruolo `UpdateData` nell'account Destination. Il ruolo Analista invece non può utilizzare il ruolo `UpdateData`.

Successivamente, vedrai come David, uno sviluppatore, può accedere al bucket `amzn-s3-demo-bucket-shared-container` nell'account Destination. David può accedere al bucket da Console di gestione AWS AWS CLI, the o dall' AWS API.

Accesso al test tramite cambio di ruoli

Al termine dei primi due passaggi del tutorial, disponi di un ruolo che concede l'accesso a una risorsa dell'account Destination. Hai anche creato un ruolo nell'account Originating con utenti autorizzati a utilizzare tale ruolo. Questo passaggio illustra come testare il passaggio a quel ruolo da Console di gestione AWS AWS CLI, the AWS e API.

Per ottenere assistenza su problemi comuni che possono verificarsi durante l'utilizzo dei ruoli IAM, consulta Risoluzione dei problemi relativi ai ruoli IAM.

Cambio di ruoli (Console)

Se David ha bisogno di aggiornare i dati nell'account di destinazione in Console di gestione AWS, può farlo utilizzando Switch Role. Specificando l'ID account o l'alias e il nome del ruolo, le sue autorizzazioni passano immediatamente a quelle consentite dal ruolo. Potrà quindi utilizzare la console per lavorare con il bucket `amzn-s3-demo-bucket-shared-container`, ma non sarà in grado utilizzare le altre risorse dell'account Destination. Inoltre, mentre utilizza il ruolo, David non può sfruttare i suoi privilegi di utente avanzato, validi per l'account Originating, perché non si possono attivare più set di autorizzazioni contemporaneamente.

IAM fornisce due alternative per accedere alla pagina Switch Role (Cambia ruolo):

- David riceve dal suo amministratore un link che rimanda a una configurazione "Switch Role" (Cambia ruolo) predefinita. Il collegamento viene fornito all'amministratore nella pagina finale della procedura guidata Create role (Crea ruolo) oppure nella pagina Role Summary (Riepilogo ruolo) per un ruolo tra più account. Selezionando questo link, David viene indirizzato alla pagina Switch Role (Cambia ruolo) con i campi Account ID (ID account) e Role name (Nome ruolo) già compilati. David non deve fare altro che selezionare Cambia ruoli.
- Anziché spedire un'e-mail con il link, l'amministratore invia il numero Account ID (ID account) e i valori per Role Name (Nome ruolo). Per cambiare ruolo, David deveinserire manualmente i valori. Tale procedura viene descritta di seguito.

Come assumere un ruolo

1. David accede Console di gestione AWS utilizzando il suo utente normale nell'account Originating.
2. Seleziona il link che l'amministratore gli ha inviato per e-mail. A questo punto, David viene indirizzato alla pagina Switch Role (Cambia ruolo) che contiene già le informazioni relative all'ID account o all'alias e il nome del ruolo.

   oppure

   David seleziona il proprio nome nel menu Identity (Identità) della barra di navigazione, quindi sceglie Switch Roles (Cambia ruolo).

   Se questa è la prima volta che David cerca di accedere alla pagina Switch Role (Cambia ruolo) in questo modo, verrà visualizzata una pagina introduttiva di Switch Role (Cambia ruolo) In questa pagina sono riportate ulteriori informazioni su come il cambio di ruolo può consentire agli utenti di gestire le risorse su più Account AWS. In questa pagina, David deve selezionare Switch Role (Cambia ruolo) per completare il resto della procedura.
3. Successivamente, per accedere al ruolo, David dovrà digitare manualmente il numero di ID dell'account Destination (`999999999999`) e il nome del ruolo (`UpdateData`).

   Inoltre, David vuole monitorare quali ruoli e autorizzazioni associate sono attualmente attivi su IAM. Per tenere traccia di queste informazioni, digita `Destination` nella casella di testo Nome di visualizzazione, seleziona l'opzione di colore rosso e quindi seleziona Cambia ruolo.
4. Ora David può utilizzare la console Amazon S3 per lavorare con il bucket Amazon S3 o con qualsiasi altra risorsa per la quale il ruolo `UpdateData` dispone di autorizzazioni.

5. Al termine, David può tornare alle sue autorizzazioni originali. A tale scopo, seleziona il nome del ruolo Destination nella barra di navigazione, quindi seleziona Torna a David @ 111111111111.
6. Se dovesse avere nuovamente bisogno di cambiare ruolo, David potrà selezionare il menu Identità nel riquadro di navigazione e troverà già presente la voce Destination. Non dovrà fare altro che selezionare tale voce per cambiare immediatamente ruolo, senza immettere nuovamente l'account ID e il nome del ruolo.

Cambia ruolo (AWS CLI)

Se David dovesse avere bisogno di lavorare nell'ambiente Destination, alla riga di comando, può farlo tramite la AWS CLI. Esegue il comando `aws sts assume-role` e trasferisce l'ARN del ruolo per ottenere le credenziali di sicurezza provvisorie per quel ruolo. Quindi configura tali credenziali nelle variabili di ambiente in modo che AWS CLI i comandi successivi funzionino utilizzando le autorizzazioni del ruolo. Mentre utilizza il ruolo, David non può sfruttare i suoi privilegi di utente avanzato, validi per l'account Originating, perché non si possono attivare più set di autorizzazioni contemporaneamente.

Tutte le chiavi di accesso e i token sono solo esempi e non possono essere utilizzati come mostrato. Sostituiscili con i valori appropriati del tuo ambiente reale.

Come assumere un ruolo

1. David apre una finestra del prompt dei comandi e conferma che il AWS CLI client sta funzionando eseguendo il comando:

   ```
   aws help
   ```

   > **Note**
   >
   > L'ambiente predefinito di David utilizza le credenziali utente `David` ottenute dal profilo predefinito creato con il comando `aws configure`. Per ulteriori informazioni, consulta Configurazione della AWS Command Line Interface nella Guida per l'utente di AWS Command Line Interface .

2. Inizia il processo di cambio ruolo eseguendo il seguente comando per passare al ruolo `UpdateData` dell'account Destination. Ha ricevuto l'ARN del ruolo dall'amministratore che ha creato il ruolo. Il comando richiede anche un nome di sessione (qualsiasi testo è valido).

```
aws sts assume-role --role-arn "arn:aws:iam::999999999999:role/UpdateData" --role-
session-name "David-ProdUpdate"
```

A questo punto David potrà consultare quanto segue:

```
{
    "Credentials": {
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "AQoDYXdzEGcaEXAMPLE2gsYULo
+Im5ZEXAMPLEeYjs1M2FUIgIJx9tQqNMBEXAMPLE
CvSRyh0FW7jEXAMPLEW+vE/7s1HRpXviG7b+qYf4nD00EXAMPLEmj4wxS04L/
uZEXAMPLECihzFB5lTYLto9dyBgSDy
EXAMPLE9/
g7QRUhZp4bqbEXAMPLENwGPyOj59pFA4lNKCIkVgkREXAMPLEjlzxQ7y52gekeVEXAMPLEDiB9ST3Uuysg
sKdEXAMPLE1TVastU1A0SKFEXAMPLEiywCC/Cs8EXAMPLEpZgOs+6hz4AP4KEXAMPLERbASP
+4eZScEXAMPLEsnf87e
NhyDHq6ikBQ==",
        "Expiration": "2014-12-11T23:08:07Z",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    }
}
```

3. David può trovare le tre parti di cui ha bisogno nella sezione Credentials (Credenziali) dell'output.

   - `AccessKeyId`
   - `SecretAccessKey`
   - `SessionToken`

   David deve configurare l' AWS CLI ambiente per utilizzare questi parametri nelle chiamate successive. Per informazioni sui vari modi di configurare le credenziali, consulta Configurare AWS Command Line Interface. Non può utilizzare il comando `aws configure`, perché non supporta l'acquisizione il token della sessione. Tuttavia, può inserire manualmente le informazioni in un file di configurazione. Poiché si tratta di credenziali provvisorie, con una durata relativamente breve, è più facile per aggiungerle all'ambiente della sessione corrente della riga di comando.

4. Per aggiungere i tre valori all'ambiente, David taglia e incolla l'output del passaggio precedente nei comandi seguenti. Per risolvere i problemi con gli accapo presenti nel token della sessione,

è possibile tagliare e incollare in un semplice editor di testo. Il testo deve essere inserito come un'unica stringa lunga, anche se qui viene riportato spezzato, per motivi di chiarezza.

L'esempio seguente mostra i comandi forniti nell'ambiente Windows, dove "set" è il comando per creare una variabile di ambiente. Su un computer Linux o MacOS, bisogna utilizzare invece il comando "export". Tutte le altre parti dell'esempio sono valide per tutti i tre ambienti.

Per dettagli sull'utilizzo di Tools for Windows Powershell, consulta Passa a un ruolo IAM (Strumenti per Windows PowerShell)

```
set AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
set AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
set AWS_SESSION_TOKEN=AQoDYXdzEGcaEXAMPLE2gsYULo
+Im5ZEXAMPLEeYjs1M2FUIgIJx9tQqNMBEXAMPLECvS
Ryh0FW7jEXAMPLEW+vE/7s1HRpXviG7b+qYf4nD00EXAMPLEmj4wxS04L/
uZEXAMPLECihzFB5lTYLto9dyBgSDyEXA
MPLEKEY9/
g7QRUhZp4bqbEXAMPLENwGPyOj59pFA4lNKCIkVgkREXAMPLEjlzxQ7y52gekeVEXAMPLEDiB9ST3UusKd
EXAMPLE1TVastU1A0SKFEXAMPLEiywCC/Cs8EXAMPLEpZgOs+6hz4AP4KEXAMPLERbASP
+4eZScEXAMPLENhykxiHen
DHq6ikBQ==
```

A questo punto, tutti i comandi successivi vengono eseguiti con le autorizzazioni del ruolo identificato da tali credenziali (nel caso di David, il ruolo `UpdateData`).

> **Important**
>
> Puoi salvare le impostazioni di configurazione e le credenziali utilizzate di frequente nei file gestiti da AWS CLI. Per ulteriori informazioni, consulta File di configurazione e delle credenziali esistenti nella Guida per l'utente di AWS Command Line Interface .

5. Eseguire il comando per accedere alle risorse dell'account Destination. In questo esempio, David si limita a elencare il contenuto del suo bucket S3 con il comando seguente.

```
aws s3 ls s3://shared-container
```

Poiché i nomi del bucket Amazon S3 sono universalmente univoci, non è necessario specificare quale ID account possiede il bucket. Per accedere alle risorse per altri AWS servizi, consulta

la AWS CLI documentazione di quel servizio per i comandi e la sintassi necessari per fare riferimento alle sue risorse.

Utilizzando AssumeRole (AWS API)

Per aggiornare l'account Destination da codice, David effettua una chiamata `AssumeRole` per assumere il ruolo `UpdateData`. La chiamata restituisce le credenziali provvisorie, utilizzabili per accedere al bucket `amzn-s3-demo-bucket-shared-container` dell'account Destination. David può utilizzare tali credenziali per effettuare chiamate API per aggiornare il bucket `amzn-s3-demo-bucket-shared-container`. Tuttavia, non sarà in grado di effettuare chiamate API per accedere alle altre risorse dell'account Destination, anche se dispone di autorizzazioni da utente avanzato per l'account Originating.

Come assumere un ruolo

1. David richiama `AssumeRole` come parte di un'applicazione Deve specificare l'ARN di `UpdateData`: `arn:aws:iam::999999999999:role/UpdateData`.

   La risposta alla chiamata `AssumeRole` include le credenziali temporanee con un `AccessKeyId` e un `SecretAccessKey`. Include anche un'ora `Expiration` che indica quando le credenziali scadono e sarà necessario richiederne di nuove. Quando configuri il concatenamento dei ruoli con l' AWS SDK, molti provider di credenziali aggiornano automaticamente le credenziali prima che scadano.
2. David utilizza le credenziali provvisorie per inviare una chiamata `s3:PutObject` per aggiornare il bucket `amzn-s3-demo-bucket-shared-container`. Trasferisce le credenziali alla chiamata API come parametro `AuthParams`. Dato che le credenziali provvisorie del ruolo forniscono solo un accesso in lettura e scrittura al bucket `amzn-s3-demo-bucket-shared-container`, tutte le altre azioni nell'account Destination sono negate.

Per un esempio di codice (con Python), consultare Passa a un ruolo IAM (AWS API).

Risorse aggiuntive

Le seguenti risorse possono aiutarti a saperne di più sugli argomenti trattati in questo tutorial:

- Per ulteriori informazioni sugli utenti IAM, consulta Identità IAM.
- Per ulteriori informazioni sui bucket Amazon S3, consulta Creazione di un bucket nella Guida per l'utente di Amazon Simple Storage Service.

- Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Riepilogo

Hai completato il tutorial per l'accesso alle API di più account. Hai creato un ruolo per stabilire la relazione di trust con un altro account e hai definito le operazioni che possono essere eseguite dalle entità affidabili. Successivamente, hai modificato una policy del ruolo per controllare quali utenti IAM possono accedere al ruolo. Come risultato, gli sviluppatori dell'account Originating possono aggiornare il bucket `amzn-s3-demo-bucket-shared-container` dell'account Destination, utilizzando credenziali provvisorie.

Tutorial IAM: Creazione e collegamento della prima policy gestita dal cliente

In questo tutorial, utilizzerai la Console di gestione AWS per creare una policy gestita dai clienti e poi allegherai tale policy a un utente IAM del tuo Account AWS. La policy che crei consente a un utente di test IAM di accedere direttamente a Console di gestione AWS con autorizzazioni di sola lettura.

Questo flusso di lavoro ha tre fasi di base:

Fase 1: creazione della policy

Per impostazione predefinita, gli utenti IAM non hanno autorizzazioni per alcuna operazione. Non possono accedere alla Console di gestione AWS né gestire i dati al suo interno, a meno che non venga loro permesso di farlo. In questa fase crei una policy gestita dal cliente che permette agli utenti collegati di accedere alla console.

Fase 2: collegamento della policy

Quando colleghi una policy a un utente, l'utente eredita tutte le autorizzazioni di accesso associate alla policy. In questa fase, colleghi la nuova policy a un utente di test.

Fase 3: test dell'accesso utente

Una volta che la policy è collegata, puoi effettuare l'accesso come utente e testare la policy.

Prerequisiti

Per eseguire le fasi in questo tutorial, devi disporre di quanto segue:

- E a Account AWS cui puoi accedere come utente IAM con autorizzazioni amministrative.
- Un utente IAM di test che non dispone di autorizzazioni assegnate o di appartenenze ai gruppi, come illustrato di seguito:

Nome utente	Gruppo	Permissions
PolicyUser	<nessuno>	<nessuno>

Fase 1: creazione della policy

In questo passaggio, crei una policy gestita dai clienti che consente a qualsiasi utente collegato di accedere Console di gestione AWS con accesso in sola lettura ai dati IAM.

Per creare la policy per l'utente di test

1. Accedi alla console IAM all'indirizzo https://console.aws.amazon.com/iam/con il tuo utente con autorizzazioni di amministratore.
2. Nel pannello di navigazione, seleziona Policies (Policy).
3. Nel riquadro del contenuto seleziona Create policy (Crea policy).
4. Seleziona l'opzione JSON e copia il testo dal seguente documento della policy JSON. Incolla il testo nella casella di testo JSON.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [ {
        "Effect": "Allow",
        "Action": [
            "iam:GenerateCredentialReport",
            "iam:Get*",
            "iam:List*"
        ],
        "Resource": "*"
    } ]
}
```

5. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o se si seleziona Rivedi policy nella scheda Editor visivo, IAM potrebbe ristrutturare la policy per ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

6. Nella pagina Verifica policy, digita **`UsersReadOnlyAccessToIAMConsole`** come nome della policy. Esamina le autorizzazioni concesse dalla policy, quindi scegli Crea policy per salvare il lavoro.

   La nuova policy appare nell'elenco delle policy gestite ed è pronta a collegare.

Fase 2: collegamento della policy

Collega quindi la policy appena creata all'utente di test IAM.

Per collegare la policy all'utente di test

1. Nel pannello di navigazione della console IAM seleziona Policy.
2. Nella parte superiore dell'elenco delle policy, nella casella di ricerca, inizia a digitare **`UsersReadOnlyAccessToIAMConsole`** finché non viene visualizzata la policy. Quindi scegli il pulsante di opzione accanto UsersReadOnlyAccessToIAMConsoleall'elenco.
3. Fai clic sul pulsante Actions (Operazioni), quindi scegli Attach (Collega).
4. In Entità IAM scegli l'opzione per filtrare in base a Utenti.
5. Nella casella di ricerca, inizia a digitare **`PolicyUser`** fino a quando l'utente non è visibile nell'elenco. Quindi seleziona la casella accanto a tale utente nell'elenco.
6. Scegli Collega policy.

La policy è stata collegata all'utente di test IAM quindi ora l'utente dispone di accesso in sola lettura alla console IAM.

Fase 3: test dell'accesso utente

Per questa esercitazione, consigliamo di verificare l'accesso autenticandosi come utente di prova in modo da poter visualizzare la potenziale esperienza degli utenti.

Per testare l'accesso accedendo con l'utente di test

1. Accedi alla console IAM all'indirizzo https://console.aws.amazon.com/iam/con il tuo utente `PolicyUser` di prova.
2. Esplora le pagine della console e prova a creare un nuovo utente o gruppo. Tieni presente che `PolicyUser` può visualizzare i dati, ma non può creare o modificare i dati IAM esistenti.

Risorse correlate

Per informazioni correlate, consulta le seguenti risorse:

- Policy gestite e policy inline
- Controlla l'accesso degli utenti IAM a Console di gestione AWS

Riepilogo

Hai completato tutte le fasi necessarie per creare e collegare una policy gestita dal cliente. Di conseguenza, è possibile accedere alla console IAM con il proprio account di test per verificare l'esperienza degli utenti.

Tutorial IAM: Definizione delle autorizzazioni di accesso AWS risorse basate sui tag

Attribute-based il controllo degli accessi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag alle risorse IAM, incluse le entità IAM (utenti o ruoli) e alle AWS risorse. È possibile definire policy che utilizzano le chiavi condizionali sui tag per concedere autorizzazioni ai principali sulla base dei relativi tag. Quando utilizzi i tag per controllare l'accesso alle tue AWS risorse, consenti ai tuoi team e alle tue risorse di crescere con meno modifiche alle AWS policy. Le politiche ABAC sono più flessibili delle AWS politiche tradizionali, che richiedono di elencare ogni singola risorsa. Per

ulteriori informazioni su ABAC e i suoi vantaggi rispetto alle policy tradizionali, consulta Definire le autorizzazioni basate su attributi con l'autorizzazione ABAC.

> **Note**
>
> È necessario passare un singolo valore per ogni tag di sessione. Servizio di token di sicurezza AWS non supporta tag di sessione con più valori.

Argomenti

- Panoramica del tutorial
- Prerequisiti
- Fase 1: creazione degli utenti di test
- Fase 2: creazione della policy ABAC
- Fase 3: creazione di ruoli
- Fase 4: verifica della creazione di segreti
- Fase 5: verifica della visualizzazione dei segreti
- Fase 6: verifica della scalabilità
- Fase 7: verifica dell'aggiornamento e dell'eliminazione dei segreti
- Riepilogo
- Risorse correlate
- Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC

Panoramica del tutorial

Questo tutorial mostra come creare e testare una policy che consente ai ruoli IAM con tag del principale di accedere alle risorse con i tag corrispondenti. Quando un principale effettua una richiesta ad AWS, le autorizzazioni vengono concesse in base al fatto che i tag del principale e quelli della risorsa corrispondano. Questa strategia consente alle persone di visualizzare o modificare solo AWS le risorse necessarie per il proprio lavoro.

Scenario

Si supponga di essere uno sviluppatore responsabile in una grande azienda denominata Example Corporation e di essere un amministratore IAM esperto. Si ha familiarità con la creazione e la

gestione di utenti, ruoli e policy IAM. Si desidera garantire che i gli ingegneri dedicati allo sviluppo e i membri del team di garanzia della qualità possano accedere alle risorse di cui hanno bisogno. C'è anche bisogno di una strategia in grado di ridimensionarsi man mano che l'azienda cresce.

Scegliete di utilizzare i tag AWS delle risorse e i tag principali dei ruoli IAM per implementare una strategia ABAC per i servizi che la supportano, a cominciare AWS Secrets Manager da. Per informazioni su quali servizi supportano l'autorizzazione basata sui tag, consulta AWS servizi supportati da IAM. Per sapere quali chiavi di condizione di tagging puoi utilizzare in una policy con le azioni e le risorse di ciascun servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. Puoi configurare il tuo provider di identità web SAML-based o il tuo provider di identità web a cui passare i tag di AWS sessione. Quando i dipendenti si federano in AWS, i loro attributi vengono applicati al committente risultante. AWSÈ quindi possibile utilizzare ABAC per consentire o negare le autorizzazioni sulla base di tali attributi. Per informazioni su come l'utilizzo di tag di sessione con un'identità federata SAML differisce da questa esercitazione, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.

I membri dei team Engineering e Quality Assurance fanno parte del progetto Pegasus o Unicorn . È possibile scegliere i seguenti valori di tag lunghi 3 caratteri per progetto e team:

- `access-project` = `peg` per il progetto Pegasus
- `access-project` = `uni` per il progetto Unicorn
- `access-team` = `eng` per il team di Engineering
- `access-team` = `qas` per il team di Quality Assurance

Inoltre, scegli di richiedere il tag di allocazione dei `cost-center` costi per abilitare i report di AWS fatturazione personalizzati. Per ulteriori informazioni, consulta la pagina sull'utilizzo dei tag per l'allocazione dei costi nella Guida per l'utente di Gestione costi e fatturazione AWS .

Riepilogo delle scelte principali

- I dipendenti eseguono l'accesso con le credenziali dell'utente IAM e quindi assumono il ruolo IAM associato ai relativi team e il progetto. Se l'azienda dispone di un proprio sistema di identità, puoi configurare la federazione per consentire ai dipendenti di assumere un ruolo senza dover passare attraverso gli utenti IAM. Per ulteriori informazioni, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.
- La stessa policy è collegata a tutti i ruoli. Le operazioni sono consentite o negate in base ai tag.

- I dipendenti possono creare nuove risorse, ma solo se collegano alla risorsa gli stessi tag che sono applicati al loro ruolo. In questo modo i dipendenti possono visualizzare la risorsa dopo averla creata. Gli amministratori non sono più tenuti ad aggiornare le policy con l'ARN delle nuove risorse.
- I dipendenti possono leggere le risorse di proprietà del loro team, indipendentemente dal progetto.
- I dipendenti possono aggiornare ed eliminare le risorse di proprietà del proprio team e progetto.
- Gli amministratori IAM possono aggiungere un nuovo ruolo per i nuovi progetti. Possono creare e associare tag a un nuovo utente IAM per consentire l'accesso al ruolo appropriato. Gli amministratori non sono tenuti a modificare una policy per supportare un nuovo progetto o membro del team.

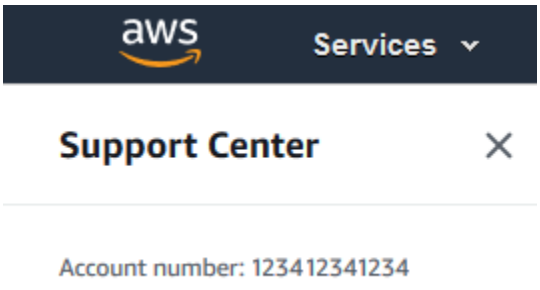
In questa esercitazione, verranno associati tag a tutte le risorse e ai ruoli del progetto e aggiunte policy ai ruoli per consentire il comportamento precedentemente descritto. La policy risultante consente ai ruoli `Create`, `Read`, `Update` e `Delete` l'accesso alle risorse contrassegnate con gli stessi tag di progetto e team. La policy consente inoltre l'accesso tra progetti in modalità `Read` per le risorse contrassegnate con lo stesso team.
Il diagramma mostra due progetti in cui i ruoli sono limitati all'accesso in sola lettura all'esterno del progetto e dispongono delle autorizzazioni per creare, leggere, aggiornare ed eliminare risorse nel proprio progetto.

Prerequisiti

Per eseguire queste fasi in questo tutorial, è necessario quanto segue:

- E a Account AWS cui puoi accedere come utente con autorizzazioni amministrative.
- L'ID account a 12 cifre, che si utilizza per creare i ruoli nel passaggio 3.

  Per trovare il numero ID del tuo AWS account utilizzando il Console di gestione AWS, scegli Assistenza nella barra di navigazione in alto a destra, quindi scegli Centro assistenza. Il numero dell'account (ID) viene visualizzato nel riquadro di navigazione a sinistra.



- Creazione e modifica di utenti, ruoli e policy IAM nella Console di gestione AWS. Tuttavia, se hai bisogno di aiuto per ricordare una procedura di gestione di IAM, questo tutorial fornisce i collegamenti da cui potrai visualizzare le istruzioni dettagliate.

Fase 1: creazione degli utenti di test

A scopo di test, crea quattro utenti IAM con le autorizzazioni per assumere ruoli con gli stessi tag. In questo modo è più facile aggiungere più utenti ai team. Quando si associano i tag agli utenti, questi ottengono automaticamente l'accesso per assumere il ruolo corretto. Non è necessario aggiungere gli utenti alla policy di trust del ruolo se lavorano su un solo progetto e in un solo team.

1. Creare la seguente policy gestita dal cliente denominata `access-assume-role`. Per ulteriori informazioni sulla creazione della policy JSON, consulta Creazione di policy IAM.

   Policy ABAC: assumere qualsiasi ruolo ABAC, ma solo quando i tag utente e ruolo corrispondono

   La policy seguente consente a un utente di assumere qualsiasi ruolo nell'account con il prefisso `access-` nel nome. Il ruolo deve inoltre essere taggato con gli stessi tag di progetto, team e centro di costi dell'utente.

   Per utilizzare questa policy, sostituisci il testo segnaposto in corsivo con le tue informazioni.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "TutorialAssumeRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::111122223333:role/access-*",
            "Condition": {
                "StringEquals": {
                    "iam:ResourceTag/access-project": "${aws:PrincipalTag/
access-project}",
                    "iam:ResourceTag/access-team": "${aws:PrincipalTag/
access-team}",
```

```
                    "iam:ResourceTag/cost-center": "${aws:PrincipalTag/cost-
center}"
                }
            }
        }
    ]
}
```

Per ridimensionare questa esercitazione a un numero elevato di utenti, è possibile collegare la policy a un gruppo e aggiungere ogni utente al gruppo. Per ulteriori informazioni, consultare Creare gruppi IAM e Modificare gli utenti nei gruppi IAM.

2. Creare i seguenti utenti IAM e collegare la policy di autorizzazione `access-assume-role`. Assicurarsi di selezionare Fornire l'accesso agli utenti a Console di gestione AWS, poi aggiungere i seguenti tag.

Nome utente	Chiave tag utente	Valore tag utente
accesso- Arnav-peg-eng	`access-project`	`peg`
	`access-team`	`eng`
	`cost-center`	`987654`
accesso- Mary-peg-qas	`access-project`	`peg`
	`access-team`	`qas`
	`cost-center`	`987654`
accesso- Saanvi-uni-eng	`access-project`	`uni`
	`access-team`	`eng`
	`cost-center`	`123456`
accesso- Carlos-uni-qas	`access-project`	`uni`
	`access-team`	`qas`

Nome utente	Chiave tag utente	Valore tag utente
	`cost-center`	`123456`

Fase 2: creazione della policy ABAC

Creare la seguente policy denominata **`access-same-project-team`**. Questa policy verrà aggiunta ai ruoli in un passaggio successivo. Per ulteriori informazioni sulla creazione della policy JSON, consulta Creazione di policy IAM.

Per ulteriori policy che è possibile adattare a questa esercitazione, vedere le pagine seguenti:

- Controllo dell'accesso per i principali IAM
- Amazon EC2: consente l'avvio o l'arresto di istanze EC2 che un utente ha contrassegnato, a livello programmatico e nella console
- EC2: avvio o arresto di istanze in base alla corrispondenza dei tag della risorsa e del principale
- EC2: avvia o interrompe le istanze in base ai tag
- IAM: assumere ruoli che dispongono di un tag specifico

Policy ABAC: accesso alle risorse di Secrets Manager solo quando il tag del principale e quello della risorsa corrispondono

La policy seguente consente ai principali di creare, leggere, modificare ed eliminare risorse, ma solo quando tali risorse sono contrassegnate con le stesse coppie chiave-valore del principale. Quando un principale crea una risorsa, deve aggiungere i tag `access-project`, `access-team` e `cost-center` con valori corrispondenti ai tag del principale. La policy consente anche l'aggiunta di tag facoltativi `Name` o `OwnedBy`.

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
     {
         "Sid": "AllActionsSecretsManagerSameProjectSameTeam",
         "Effect": "Allow",
         "Action": "secretsmanager:*",
```

```
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/access-project": "${aws:PrincipalTag/access-
project}",
                "aws:ResourceTag/access-team": "${aws:PrincipalTag/access-
team}",
                "aws:ResourceTag/cost-center": "${aws:PrincipalTag/cost-center}"
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": [
                    "access-project",
                    "access-team",
                    "cost-center",
                    "Name",
                    "OwnedBy"
                ]
            },
            "StringEqualsIfExists": {
                "aws:RequestTag/access-project": "${aws:PrincipalTag/access-
project}",
                "aws:RequestTag/access-team": "${aws:PrincipalTag/access-team}",
                "aws:RequestTag/cost-center": "${aws:PrincipalTag/cost-center}"
            }
        }
    },
    {
        "Sid": "AllResourcesSecretsManagerNoTags",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetRandomPassword",
            "secretsmanager:ListSecrets"
        ],
        "Resource": "*"
    },
    {
        "Sid": "ReadSecretsManagerSameTeam",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:Describe*",
            "secretsmanager:Get*",
            "secretsmanager:List*"
        ],
        "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/access-team": "${aws:PrincipalTag/access-team}"
                }
            }
        },
        {
            "Sid": "DenyUntagSecretsManagerReservedTags",
            "Effect": "Deny",
            "Action": "secretsmanager:UntagResource",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringLike": {
                    "aws:TagKeys": "access-*"
                }
            }
        },
        {
            "Sid": "DenyPermissionsManagement",
            "Effect": "Deny",
            "Action": "secretsmanager:*Policy",
            "Resource": "*"
        }
    ]
}
```

Che cosa fa questa policy?

- L'istruzione `AllActionsSecretsManagerSameProjectSameTeam` consente tutte le operazioni relative a questo servizio su tutte le risorse correlate, ma solo se i tag di risorsa corrispondono ai tag del principale. Aggiungendo `"Action": "secretsmanager:*"` alla policy, la policy stessa cresce man mano che Secrets Manager cresce. Se Secrets Manager aggiunge una nuova operazione API, non è necessario aggiungere tale operazione all'istruzione. L'istruzione implementa ABAC utilizzando tre blocchi di condizione. La richiesta è consentita solo se tutti e tre i blocchi restituiscono true.
  - Il primo blocco di condizione di questa istruzione restituisce true se le chiavi tag specificate sono presenti nella risorsa e i loro valori corrispondono ai tag del principale. Questo blocco restituisce false per i tag non corrispondenti o per le operazioni che non supportano il tag delle risorse. Per sapere quali azioni non sono consentite da questo blocco, vedi Azioni, risorse e chiavi di condizione per AWS Secrets Manager. Questa pagina mostra che le operazioni eseguite sul

tipo di risorsa Segreto supportano la chiave di condizione `secretsmanager:ResourceTag/tag-key`. Alcune azioni di Secrets Manager non supportano tale tipo di risorsa, inclusi `GetRandomPassword` e `ListSecrets`. Per consentire tali azioni, è necessario creare istruzioni aggiuntive.

- Il secondo blocco di condizione restituisce true se ogni chiave del tag passata nella richiesta è incluso nell'elenco specificato. Questo viene fatto utilizzando `ForAllValues` con l'operatore condizionale `StringEquals`. Se non vengono passate chiavi o un sottoinsieme del set di chiavi, la condizione restituisce true. Ciò consente operazioni `Get*` che non consentono il passaggio di tag nella richiesta. Se il richiedente include una chiave del tag che non è nell'elenco, la condizione restituisce false. Ogni chiave dei tag passata nella richiesta deve corrispondere a un elemento di tale elenco. Per ulteriori informazioni, consulta Operatori dell'insieme per le chiavi di contesto multivalore.
- Il terzo blocco di condizioni restituisce true se la richiesta supporta il passaggio dei tag, se tutti e tre i tag sono presenti e se corrispondono ai valori del tag del principale. Questo blocco restituisce true anche se la richiesta non supporta il passaggio di tag. Tale risultato è ottenuto tramite l'utilizzo di `...IfExists` nell'operatore condizionale. Il blocco restituisce false se non vi è alcun tag passato durante un'operazione che li supporta o se le chiavi e i valori dei tag non corrispondono.

- L'istruzione `AllResourcesSecretsManagerNoTags` permette le operazioni `GetRandomPassword` e `ListSecrets` che non sono consentite dalla prima istruzione.
- L'istruzione `ReadSecretsManagerSameTeam` permette le operazioni di sola lettura se il principale è contrassegnato con lo stesso tag di team di accesso della risorsa. Ciò è consentito indipendentemente dal progetto o dal tag del centro di costi.
- L'istruzione `DenyUntagSecretsManagerReservedTags` rifiuta le richieste di rimuovere da Secrets Manager i tag con chiavi che iniziano con il prefisso "access-". Questi tag vengono utilizzati per controllare l'accesso alle risorse, pertanto la rimozione dei tag potrebbe rimuovere le autorizzazioni.
- L'istruzione `DenyPermissionsManagement` nega l'accesso per creare, modificare o eliminare policy basate sulle risorse di Secrets Manager. Queste policy possono essere utilizzate per modificare le autorizzazioni dei segreti.

> **Important**
>
> Questa policy utilizza una strategia per consentire tutte le operazioni per un servizio, ma negando esplicitamente le operazioni di modifica delle autorizzazioni. Il rifiuto di

un'operazione sostituisce qualsiasi altra policy che consente al principale di eseguire tale operazione. Ciò può avere risultati imprevisti. Come best practice, usare il rifiuto esplicito solo quando non vi è alcuna circostanza in cui debba essere consentita tale operazione. In caso contrario, consentire un elenco di singole operazioni in modo che le operazioni indesiderate vengano negate per impostazione predefinita.

Fase 3: creazione di ruoli

Crea i seguenti ruoli IAM e collega la policy **`access-same-project-team`** creata nella fase precedente. Per ulteriori informazioni sulla creazione dei ruoli IAM, consulta Creazione di un ruolo per fornire le autorizzazioni a un utente IAM. Se decidi di utilizzare la federazione anziché gli utenti e i ruoli IAM, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.

Funzione processo	Nome ruolo	Tag di ruolo	Descrizione del ruolo
Progetto Pegasus Engineering	access-peg-engineering	access-project = `peg` access-team = `eng` cost-center = `987654`	Consente agli ingegneri di leggere tutte le risorse ingegneristiche e di creare e gestire le risorse ingegneristiche di Pegasus.
Progetto Pegasus Quality Assurance	access-peg-quality-assurance	access-project = `peg` access-team = `qas` cost-center = `987654`	Consente al team QA di leggere tutte le risorse di QA e di creare e gestire tutte le risorse QA di Pegasus.
Progetto Unicorn Engineering	access-uni-engineering	access-project= `uni` access-team = `eng`	Consente agli ingegneri di

Funzione processo	Nome ruolo	Tag di ruolo	Descrizione del ruolo
		cost-center = 123456	leggere tutte le risorse ingegneristiche e creare e gestire le risorse ingegneristiche di Unicorn.
Progetto Unicorn Quality Assurance	access-uni-quality-assurance	access-project = uni access-team = qas cost-center = 123456	Consente al team QA di leggere tutte le risorse QA e di creare e gestire tutte le risorse QA di Unicorn.

Fase 4: verifica della creazione di segreti

La policy di autorizzazione collegata ai ruoli consente ai dipendenti di creare segreti. Questo è consentito solo se il segreto è contrassegnato con il relativo progetto, team e centro di costi. Verifica che le autorizzazioni funzionino come previsto accedendo come i tuoi utenti, assumendo il ruolo corretto e testando l'attività in Secrets Manager.

Per testare la creazione di un segreto con e senza i tag richiesti

1. Nella finestra principale del browser, resta connesso come utente amministratore in modo da poter esaminare utenti, ruoli e policy in IAM. Utilizzare una finestra di navigazione in incognito del browser o un browser separato per i test. Lì, accedi come utente `access-Arnav-peg-eng` IAM e apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Provare a passare al ruolo `access-uni-engineering`.

   Questa operazione ha esito negativo perché i valori de tag `access-project` e `cost-center` non corrispondono all'utente `access-Arnav-peg-eng` e al ruolo `access-uni-engineering`.

Per ulteriori informazioni sul cambio di ruolo in Console di gestione AWS, vedere Passare da un utente a un ruolo IAM (console)

3. Passare al ruolo `access-peg-engineering`.
4. Archiviare un nuovo segreto utilizzando le seguenti informazioni. Per informazioni su come archiviare un segreto, consulta Creazione di un segreto di base nella Guida per l'utente di AWS Secrets Manager .

> **Important**
>
> Secrets Manager visualizza avvisi che segnalano che non disponi delle autorizzazioni per i servizi AWS aggiuntivi che funzionano con Secrets Manager. Ad esempio, per creare credenziali per un database Amazon RDS, è necessario disporre dell'autorizzazione per descrivere istanze RDS, cluster RDS e cluster Amazon Redshift. Puoi ignorare questi avvisi poiché non stai utilizzando questi AWS servizi specifici in questo tutorial.

1. Nella sezione Seleziona tipo di segreto, scegliere Altro tipo di segreti. Nelle due caselle di testo, inserire `test-access-key` e `test-access-secret`.
2. Nel campo Nome segreto inserire il valore `test-access-peg-eng`.
3. Aggiungere diverse combinazioni di tag dalla tabella seguente e visualizzare il comportamento previsto.
4. Scegliere Memorizza per provare a creare il segreto. Se l'archiviazione non riesce, torna alle pagine della console di Secrets Manager precedenti e utilizza il set di tag successivo dalla tabella seguente. L'ultimo set di tag è consentito e creerà con successo il segreto.

La tabella seguente mostra le combinazioni di tag ABAC per il ruolo `test-access-peg-eng`.

Valore tag **access-project**	Valore tag **access-team**	Valore tag **cost-center**	Tag aggiuntivi	Comportamento previsto
(nessuno)	(nessuno)	(nessuno)	(nessuno)	Negato perché il valore del tag `access-project` non corrispon de al valore del ruolo di `peg`.
`uni`	`eng`	`987654`	(nessuno)	Negato perché il valore del tag `access-project` non corrispon de al valore del ruolo di `peg`.
`peg`	`qas`	`987654`	(nessuno)	Negato perché il valore del tag `access-team` non corrisponde al valore del ruolo di `eng`.
`peg`	`eng`	`123456`	(nessuno)	Negato perché il valore del tag `cost-center` non corrisponde al valore del ruolo di `987654`.
`peg`	`eng`	`987654`	Proprieta rio = Jane	Negato perché il tag aggiuntiv o `owner` non è consentito dalla policy, anche se tutti e tre i tag richiesti sono presenti e i relativi valori corrispondono ai valori del ruolo.
`peg`	`eng`	`987654`	Nome = Jane	Consentito perché tutti e tre i tag richiesti sono presenti e i loro valori corrispondono ai valori del ruolo. È anche possibile includere il tag opzionale `Name`.

5. Disconnettersi e ripetere i primi tre passaggi di questa procedura per ciascuno dei seguenti ruoli e valori dei tag. Nel quarto passaggio di questa procedura, verificare tutti i set di tag mancanti, tag facoltativi, tag non consentiti e valori di tag non validi selezionati. Quindi utilizzare i tag richiesti per creare un segreto con i seguenti tag e nome.

Nome utente	Nome ruolo	Nome segreto	Tag segreto
accesso- Mary-peg-qas	access-peg-quality-assurance	test-access-peg-qas	access-project = `peg` access-team = `qas` cost-center = `987654`
accesso- Saanvi-uni-eng	access-uni-engineering	test-access-uni-eng	access-project = `uni` access-team = `eng` cost-center = `123456`
accesso- Carlos-uni-qas	access-uni-quality-assurance	test-access-uni-qas	access-project = `uni` access-team = `qas` cost-center = `123456`

Fase 5: verifica della visualizzazione dei segreti

La policy collegata a ciascun ruolo consente ai dipendenti di visualizzare eventuali segreti contrassegnati con il nome del team, indipendentemente dal progetto. Verifica che le autorizzazioni funzionino come previsto testando i ruoli in Secrets Manager.

Per testare la visualizzazione di un segreto con e senza i tag richiesti

1. Accedi come uno dei seguenti utenti IAM:
   - `access-Arnav-peg-eng`
   - `access-Mary-peg-qas`
   - `access-Saanvi-uni-eng`
   - `access-Carlos-uni-qas`
2. Passa al ruolo corrispondente:

- `access-peg-engineering`
- `access-peg-quality-assurance`
- `access-uni-engineering`
- `access-uni-quality-assurance`

Per ulteriori informazioni sul cambio di ruolo in Console di gestione AWS, vedere[Passare da un utente a un ruolo IAM (console)](#).

3. Nel riquadro di navigazione a sinistra, scegli l'icona del menu per espanderlo, quindi scegliere Segreti.
4. Dovrebbero essere visualizzati tutti e quattro i segreti nella tabella, indipendentemente dal proprio ruolo attuale. Ciò accade perché la policy denominata `access-same-project-team` consente l'operazione `secretsmanager:ListSecrets` per tutte le risorse.
5. Scegli il nome di uno dei segreti.
6. Nella pagina dei dettagli del segreto, i tag del ruolo determinano se è possibile visualizzare il contenuto della pagina. Confrontare il nome del proprio ruolo con il nome del segreto. Se condividono lo stesso nome del team, i tag `access-team` corrispondono. Se non corrispondono, l'accesso viene negato.

   La tabella seguente mostra il comportamento di visualizzazione del segreto ABAC per ogni ruolo.

Nome ruolo	Nome segreto	Comportamento previsto
`access-peg-engineering`	`test-access-peg-eng`	Consentito
	test-access-peg-qas	Negato
	test-access-uni-eng	Consentito
	test-access-uni-qas	Negato
access-peg-quality-assurance	test-access-peg-eng	Negato
	test-access-peg-qas	Consentito
	test-access-uni-eng	Negato

Nome ruolo	Nome segreto	Comportamento previsto
	test-access-uni-qas	Consentito
access-uni-engineering	test-access-peg-eng	Consentito
	test-access-peg-qas	Negato
	test-access-uni-eng	Consentito
	test-access-uni-qas	Negato
access-uni-quality-assurance	test-access-peg-eng	Negato
	test-access-peg-qas	Consentito
	test-access-uni-eng	Negato
	test-access-uni-qas	Consentito

7. Nel percorso di navigazione nella parte superiore della pagina, scegliere Segreti per tornare all'elenco dei segreti. Ripetere i passaggi di questa procedura utilizzando ruoli diversi per verificare se è possibile visualizzare ciascuno dei segreti.

Fase 6: verifica della scalabilità

Un motivo importante per utilizzare il controllo degli accessi basato su attributi (ABAC) invece del controllo di accesso basato su ruoli (RBAC) è la scalabilità. Man mano che la tua azienda aggiunge nuovi progetti, team o persone AWS, non è necessario aggiornare le ABAC-driven politiche. Ad esempio, si supponga che Example Company stia finanziando un nuovo progetto dal nome in codice Centaur. Un ingegnere di nome Saanvi Sarkar sarà l'ingegnere responsabile di Centaur continuando a lavorare al progetto Unicorn . Saanvi esaminerà anche i lavori per il progetto Peg. Ci sono anche diversi ingegneri appena assunti, tra cui Nikhil Jayashankar, che lavoreranno solo al progetto Centaur .

Per aggiungere il nuovo progetto a AWS

1. Accedi come utente amministratore IAM e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione sulla sinistra, scegli Ruoli e aggiungi un ruolo IAM denominato `access-cen-engineering`. Collega la policy delle autorizzazioni **`access-same-project-team`** al ruolo e aggiungere i seguenti tag di ruolo:

   - `access-project` = `cen`
   - `access-team` = `eng`
   - `cost-center` = `101010`

3. Nel riquadro di navigazione sinistro, scegli Utenti.
4. Aggiungi un nuovo utente denominato `access-Nikhil-cen-eng`, collega la policy denominata `access-assume-role` e aggiungi i seguenti tag utente.

   - `access-project` = `cen`
   - `access-team` = `eng`
   - `cost-center` = `101010`

5. Utilizza le procedure in Fase 4: verifica della creazione di segreti e Fase 5: verifica della visualizzazione dei segreti. In un'altra finestra del browser, verifica che Nikhil possa creare segreti solo per il team di ingegneria di Centaur e che possa visualizzare tutti i segreti dei team di ingegneria.
6. Nella finestra principale del browser in cui hai effettuato l'accesso come amministratore, scegli l'utente `access-Saanvi-uni-eng`.
7. Nella scheda Autorizzazioni rimuovi la policy di autorizzazione access-assume-role.
8. Aggiungi la seguente policy inline denominata `access-assume-specific-roles`. Per ulteriori informazioni sull'aggiunta di una policy inline a un utente, consulta Per incorporare una policy inline per un utente o un ruolo (console).

   Policy ABAC: assunzione dei soli ruoli specifici

   Questa policy consente a Saanvi di assumere i ruoli ingegneristici per i progetti Pegasus e Centaur. È necessario creare questa policy personalizzata perché IAM non supporta tag multivalore. Non è possibile etichettare l'utente di Saanvi con `access-project` = `peg` e `access-project` = `cen`. Inoltre, il modello di AWS autorizzazione non può corrispondere a entrambi i valori. Per ulteriori informazioni, consulta Regole per l'etichettatura in IAM e AWS STS. È invece necessario specificare manualmente i due ruoli che può assumere.

   Per utilizzare questa policy, sostituisci il testo segnaposto in corsivo con le tue informazioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "TutorialAssumeSpecificRoles",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::111122223333:role/access-peg-engineering",
                "arn:aws:iam::111122223333:role/access-cen-engineering"
            ]
        }
    ]
}
```

9. Utilizza le procedure in Fase 4: verifica della creazione di segreti e Fase 5: verifica della visualizzazione dei segreti. In un'altra finestra del browser, conferma che Saanvi possa assumere entrambi i ruoli. Verifica che sia in grado di creare segreti solo per i suoi progetto, team e centro di costo, a seconda dei tag del ruolo. Verifica anche che possa visualizzare i dettagli su eventuali segreti di proprietà del team di ingegneria, inclusi quelli che ha appena creato.

Fase 7: verifica dell'aggiornamento e dell'eliminazione dei segreti

La policy `access-same-project-team` collegata ai ruoli consente ai dipendenti di aggiornare ed eliminare eventuali segreti contrassegnati con il proprio progetto, team e centro costi. Verifica che le autorizzazioni funzionino come previsto testando i ruoli in Secrets Manager.

Per verificare l'aggiornamento e l'eliminazione di un segreto con e senza i tag richiesti

1. Accedi come uno dei seguenti utenti IAM:

   - `access-Arnav-peg-eng`
   - `access-Mary-peg-qas`
   - `access-Saanvi-uni-eng`
   - `access-Carlos-uni-qas`

- `access-Nikhil-cen-eng`

2. Passa al ruolo corrispondente:

   - `access-peg-engineering`
   - `access-peg-quality-assurance`
   - `access-uni-engineering`
   - `access-peg-quality-assurance`
   - `access-cen-engineering`

   Per ulteriori informazioni sul cambio di ruolo in Console di gestione AWS, vedere Passare da un utente a un ruolo IAM (console).

3. Per ogni ruolo, prova ad aggiornare la descrizione del segreto e quindi prova a eliminare i seguenti segreti. Per ulteriori informazioni, consulta Modifica di un segreto ed Eliminazione e ripristino di un segreto nella Guida per l'utente di AWS Secrets Manager .

   La seguente tabella riporta il comportamento di aggiornamento ed eliminazione dei segreti ABAC per ogni ruolo.

Nome ruolo	Nome segreto	Comportamento previsto
access-peg-engineering	test-access-peg-eng	Consentito
	test-access-uni-eng	Negato
	test-access-uni-qas	Negato
access-peg-quality-assurance	test-access-peg-qas	Consentito
	test-access-uni-eng	Negato
access-uni-engineering	test-access-uni-eng	Consentito
	test-access-uni-qas	Negato
access-peg-quality-assurance	test-access-uni-qas	Consentito

Riepilogo

Tutte le fasi necessarie per utilizzare i tag per il controllo degli accessi basato su attributi (ABAC) sono state completate correttamente. L'utente ha imparato a definire una strategia di tagging. Tale strategia è stata applicata alle proprie entità e risorse. È stata creata e applicata una policy che impone la strategia a Secrets Manager. L'utente ha anche imparato che ABAC è facilmente ridimensionabile nel momento in cui si aggiungono nuovi progetti e membri del team. Di conseguenza, sarai in grado di accedere alla console IAM con i ruoli di test e scoprire come utilizzare i tag per ABAC in AWS.

> **Note**
>
> L'utente ha aggiunto policy che consentono operazioni solo in condizioni specifiche. Se si applica un criterio diverso agli utenti o ai ruoli con autorizzazioni più ampie, è possibile che le azioni non siano limitate a richiedere l'assegnazione di tag. Ad esempio, se concedi a un utente le autorizzazioni amministrative complete utilizzando la policy `AdministratorAccess` AWS gestita, queste politiche non limitano tale accesso. Per ulteriori informazioni su come vengono determinate le autorizzazioni quando sono coinvolte più policy, vedere In che modo AWS la logica del codice di applicazione valuta le richieste per consentire o negare l'accesso.

Risorse correlate

Per informazioni correlate, consulta le seguenti risorse:

- Definire le autorizzazioni basate su attributi con l'autorizzazione ABAC
- AWS chiavi di contesto delle condizioni globali
- Creazione di un ruolo per fornire le autorizzazioni a un utente IAM
- Tag per AWS Identity and Access Management risorse
- Controllo dell'accesso alle AWS risorse tramite tag
- Passare da un utente a un ruolo IAM (console)
- Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC

Per informazioni su come monitorare i tag nel tuo account, consulta Monitorare le modifiche dei tag sulle AWS risorse con flussi di lavoro serverless e Amazon Events. CloudWatch

Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC

Attribute-based il controllo degli accessi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag alle risorse IAM, incluse le entità IAM (utenti o ruoli) e alle AWS risorse. Quando le entità vengono utilizzate per effettuare richieste a AWS, diventano principali e tali entità includono tag.

È inoltre possibile passare i tag di sessione quando si assume un ruolo o si federa un utente. È quindi possibile definire policy che utilizzano le chiavi condizionali sui tag per concedere autorizzazioni alle entità sulla base dei relativi tag. Quando si utilizzano i tag per controllare l'accesso alle risorse AWS , si consente ai team e alle risorse di crescere con la necessità di un minor numero di modifiche alle policy AWS . Le politiche ABAC sono più flessibili rispetto alle AWS politiche tradizionali, che richiedono di elencare ogni singola risorsa. Per ulteriori informazioni su ABAC e i suoi vantaggi rispetto alle policy tradizionali, consulta Definire le autorizzazioni basate su attributi con l'autorizzazione ABAC.

Se la tua azienda utilizza un provider di SAML-based identità (IdP) per gestire le identità degli utenti aziendali, puoi utilizzare gli attributi SAML per un controllo granulare degli accessi in. AWS Gli attributi possono includere identificatori del centro di costo, indirizzi e-mail degli utenti, reparti di appartenenza e assegnazioni di progetto. Quando si passano questi attributi come tag di sessione, è quindi possibile controllare l'accesso ad AWS in base a tali tag di sessione.

Per completare l'esercitazione su ABAC passando gli attributi SAML all'entità sessione, completare le attività descritte in Tutorial IAM: Definizione delle autorizzazioni di accesso AWS risorse basate sui tag, con le modifiche incluse in questa sezione.

Prerequisiti

Per eseguire la procedura che prevede l'utilizzo dei tag di sessione SAML per ABAC, è necessario disporre preventivamente quanto segue:

- Accesso a un SAML-based IdP in cui è possibile creare utenti di test con attributi specifici.
- La possibilità di effettuare l'accesso come utente con autorizzazioni di amministratore.
- Creazione e modifica di utenti, ruoli e policy IAM nella Console di gestione AWS. Tuttavia, se hai bisogno di aiuto per ricordare una procedura di gestione di IAM, il tutorial ABAC fornisce i collegamenti da cui è possibile visualizzare le istruzioni dettagliate.
- Esperienza nella configurazione di un SAML-based IdP in IAM. Per visualizzare maggiori dettagli e i collegamenti alla documentazione IAM dettagliata, consulta Passaggio dei tag di sessione utilizzando AssumeRoleWithSAML.

Fase 1: creazione degli utenti di test

Seguire le istruzioni in Fase 1: creazione degli utenti di test. Poiché le identità sono definite dal provider, non è necessario aggiungere gli utenti IAM per i propri dipendenti.

Fase 2: creazione della policy ABAC

Per creare la policy gestita specificata in IAM, seguire le istruzioni riportate in Fase 2: creazione della policy ABAC.

Fase 3: creazione e configurazione del ruolo SAML

Quando utilizzi il tutorial ABAC per SAML, devi eseguire passaggi aggiuntivi per creare il ruolo, configurare l'IdP SAML e abilitare l'accesso. Console di gestione AWS Per ulteriori informazioni, consulta Fase 3: creazione di ruoli.

Fase 3A: creazione del ruolo SAML

Creare un singolo ruolo in relazione di trust con il provider di identità SAML e l'utente `test-session-tags` creato nel passaggio 1. L'esercitazione ABAC utilizza ruoli separati con diversi tag di ruolo. Poiché si stanno passando i tag di sessione dal proprio IdP SAML, c'è bisogno di un solo ruolo. Per sapere come creare un SAML-based ruolo, consulta. Creare un ruolo per una federazione SAML 2.0 (console)

Denomina il ruolo `access-session-tags`. Collegare la policy di autorizzazione `access-same-project-team` al ruolo. Modificare la policy di trust al fine di utilizzare la policy riportata di seguito. Per istruzioni dettagliate su come modificare la relazione di trust di un ruolo, consulta Aggiornamento di una policy di attendibilità del ruolo.

La seguente policy di trust del ruolo consente al provider di identità SAML e all'utente `test-session-tags` di assumere il ruolo. Al momento dell'assunzione del ruolo, è necessario passare i tre tag di sessione specificati. L'operazione `sts:TagSession` è necessaria per consentire il passaggio dei tag di sessione.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "AllowSamlIdentityAssumeRole",
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRoleWithSAML",
                "sts:TagSession"
            ],
            "Principal": {"Federated":"arn:aws:iam::123456789012:saml-
provider/ExampleCorpProvider"},
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/cost-center": "*",
                    "aws:RequestTag/access-project": "*",
                    "aws:RequestTag/access-team": [
                        "eng",
                        "qas"
                    ]
                },
                "StringEquals": {"SAML:aud": "https://region-
code.signin.aws.amazon.com/saml"}
            }
        }
    ]
}
```

La `AllowSamlIdentityAssumeRole` dichiarazione consente ai membri dei team di ingegneria e controllo qualità di assumere questo ruolo quando entrano a far parte dell'IdP AWS di Example Corporation. Il provider SAML `ExampleCorpProvider` è definito in IAM. L'amministratore ha già impostato l'asserzione SAML per passare i tre tag di sessione richiesti. L'asserzione può passare tag aggiuntivi, ma questi tre devono essere presenti. Gli attributi dell'identità possono presentare qualsiasi valore per i tag `access-project` e `cost-center`. Tuttavia, il valore dell'attributo `access-team` deve corrispondere a `eng` o `qas` a indicare che l'identità corrisponde al team di Engineering o di Quality Assurance.

Passaggio 3B: configurazione dell'IdP SAML

Configurare l'IdP SAML affinché passi gli attributi `cost-center`, `access-project` e `access-team` come tag di sessione. Per ulteriori informazioni, consulta Passaggio dei tag di sessione utilizzando AssumeRoleWithSAML.

Per passare questi attributi come tag di sessione, includere i seguenti elementi nell'asserzione SAML.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:cost-center">
  <AttributeValue>987654</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:access-project">
  <AttributeValue>peg</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:access-team">
  <AttributeValue>eng</AttributeValue>
</Attribute>
```

Fase 3C: attivazione dell'accesso alla console

Abilita l'accesso alla console per gli utenti SAML federati. Per ulteriori informazioni, consulta Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS.

Fase 4: verifica della creazione di segreti

Entra nell'uso del ruolo. Console di gestione AWS `access-session-tags` Per ulteriori informazioni, consulta Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS. Quindi seguire le istruzioni in Fase 4: verifica della creazione di segreti per creare segreti. Utilizzare diverse identità SAML con attributi da abbinare ai tag indicati nell'esercitazione ABAC. Per ulteriori informazioni, consulta Fase 4: verifica della creazione di segreti.

Fase 5: verifica della visualizzazione dei segreti

Seguire le istruzioni descritte in Fase 5: verifica della visualizzazione dei segreti per visualizzare i segreti creati nel passaggio precedente. Utilizzare diverse identità SAML con attributi da abbinare ai tag indicati nell'esercitazione ABAC.

Fase 6: verifica della scalabilità

Seguire le istruzioni descritte in Fase 6: verifica della scalabilità per testare la scalabilità. A tale scopo, aggiungi una nuova identità nel tuo SAML-based IdP con i seguenti attributi:

- `cost-center = 101010`
- `access-project = cen`
- `access-team = eng`

Fase 7: verifica dell'aggiornamento e dell'eliminazione dei segreti

Seguire le istruzioni descritte in Fase 7: verifica dell'aggiornamento e dell'eliminazione dei segreti per aggiornare ed eliminare i segreti. Utilizzare diverse identità SAML con attributi da abbinare ai tag indicati nell'esercitazione ABAC.

> **Important**
>
> Eliminare tutti i segreti creati per evitare addebiti in fattura. Per informazioni sui prezzi di Secrets Manager, consulta Prezzi di AWS Secrets Manager.

Riepilogo

Sono stati completati tutti i passaggi necessari per utilizzare i tag di sessione SAML e i tag delle risorse per la gestione delle autorizzazioni.

> **Note**
>
> L'utente ha aggiunto policy che consentono operazioni solo in condizioni specifiche. Se si applica un criterio diverso agli utenti o ai ruoli con autorizzazioni più ampie, è possibile che le azioni non siano limitate a richiedere l'assegnazione di tag. Ad esempio, se concedi a un utente le autorizzazioni amministrative complete utilizzando la policy `AdministratorAccess` AWS gestita, queste politiche non limitano tale accesso. Per ulteriori informazioni su come vengono determinate le autorizzazioni quando sono coinvolte più policy, vedere In che modo AWS la logica del codice di applicazione valuta le richieste per consentire o negare l'accesso.

Tutorial IAM: consentire agli utenti di gestire le proprie credenziali e impostazioni MFA

Puoi consentire agli utenti di gestire i loro dispositivi e le credenziali di autenticazione a più fattori (MFA) nella pagina Credenziali di sicurezza. Puoi utilizzarlo Console di gestione AWS per configurare le credenziali (chiavi di accesso, password, certificati di firma e chiavi pubbliche SSH), eliminare o disattivare le credenziali non necessarie e abilitare i dispositivi MFA per i tuoi utenti. Sebbene sia utile per un numero ridotto di utenti, è un'operazione che potrebbe richiedere molto tempo se il numero di

utenti cresce. Mostrare come abilitare queste best practice senza sovraccaricare gli amministratori è l'obiettivo di questo tutorial.

Questo tutorial mostra come consentire agli utenti di accedere ai AWS servizi, ma solo quando accedono con MFA. Se non sono registrati con un dispositivo MFA, gli utenti non possono accedere ad altri servizi.

Questo flusso di lavoro ha tre fasi di base.

Fase 1: creazione di una policy per applicare l'accesso MFA

Crea una policy gestita dal cliente che impedisce tutte le azioni eccetto le poche operazioni IAM. Queste eccezioni consentono a un utente di modificare le sue credenziali e gestire i dispositivi MFA nella pagina Credenziali di sicurezza. Per ulteriori informazioni sull'accesso alla pagina, consulta Come gli utenti IAM possono cambiare le proprie password (console).

Fase 2: Collegamento delle policy al gruppo di utenti di test

Crea un gruppo di utenti i cui membri hanno accesso completo a tutte le operazioni Amazon EC2 se effettuano l'accesso con MFA. Per creare un gruppo di utenti di questo tipo, alleghi sia la politica AWS gestita richiamata `AmazonEC2FullAccess` sia la politica gestita dai clienti che hai creato nel primo passaggio.

Fase 3: test dell'accesso dell'utente

Accedi come utente di prova per verificare che l'accesso ad Amazon EC2 sia bloccato fino a quando l'utente non crea un dispositivo MFA. L'utente può quindi accedere utilizzando tale dispositivo.

Prerequisiti

Per eseguire queste fasi in questo tutorial, è necessario quanto segue:

- E a Account AWS cui puoi accedere come utente IAM con autorizzazioni amministrative.
- Il numero ID dell'account, che si digita nella policy nella Fase 1.

  Per trovare il numero ID dell'account nella barra di navigazione in alto sulla pagina, selezionare Support (Supporto) e selezionare Support Center (Centro di supporto). Puoi trovare l'ID dell'account nel menu Supporto di questa pagina.
- Un dispositivo MFA virtuale (basato su software), una chiave di sicurezza FIDO o un dispositivo MFA basato su hardware.

- Un utente IAM di prova che è membro di un gruppo come segue:

Nome utente	Istruzioni per il nome utente	Nome gruppo di utenti	Aggiungere utente come un membro	Istruzioni per il gruppo di utenti
MFAUser	Seleziona solo l'opzione per Enable console access – optional (Abilita , l'accesso alla console - facoltativo) e assegna una password.	EC2MFA	MFAUser	NON collegare policy o concedere autorizzazioni a questo gruppo di utenti.

Fase 1: creazione di una policy per applicare l'accesso MFA

Per iniziare, crea una policy gestita dal cliente IAM che nega tutte le autorizzazioni tranne quelle richieste per gli utenti IAM per gestire le credenziali e i dispositivi MFA.

1. Accedi alla console di AWS gestione come utente con credenziali di amministratore. Per aderire alle best practice di IAM, non accedere con le tue Utente root dell'account AWS credenziali.

   > **Important**
   >
   > Le migliori pratiche IAM consigliano di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee anziché utilizzare utenti IAM con credenziali a lungo termine. Ti consigliamo di utilizzare gli utenti IAM solo per casi d'uso specifici non supportati dagli utenti federati.

2. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

3. Nel pannello di navigazione, scegli Policy e Crea policy.

4. Selezionare la scheda JSON e copiare il testo dal documento della seguente policy JSON: AWS: consente agli utenti IAM autenticati con MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

5. Incolla il testo della policy nella casella di testo JSON. Risolvi eventuali avvisi di sicurezza, errori o avvertenze generali generati durante la convalida delle policy, quindi scegli Successivo.

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Tuttavia, la policy qui sopra include l'elemento `NotAction`, che non è supportato nell'editor visivo. Per questa policy, verrà visualizzata una notifica nella scheda Visual Editor (Editor visivo). Torna alla scheda JSON per continuare a lavorare con questa policy.
   > Questo esempio di policy non consente agli utenti di reimpostare la password durante il primo accesso a Console di gestione AWS . Ti consigliamo di non concedere autorizzazioni ai nuovi utenti fino a quando non hanno effettuato l'accesso e reimpostato la password.

6. Nella pagina Verifica policy, digita **`Force_MFA`** come nome della policy. Per la descrizione della policy, digita **`This policy allows users to manage their own passwords and MFA devices but nothing else unless they authenticate with MFA.`**. Nell'area Tag, puoi facoltativamente aggiungere coppie chiave-valore di tag alla policy gestita dal cliente. Esamina le autorizzazioni concesse dalla policy, quindi scegli Crea policy per salvare il lavoro.

   La nuova policy appare nell'elenco delle policy gestite ed è pronta a collegare.

Fase 2: Collegamento delle policy al gruppo di utenti di test

In seguito collega le due policy al gruppo di utenti IAM di test, che verrà utilizzato per concedere le autorizzazioni protette mediante MFA.

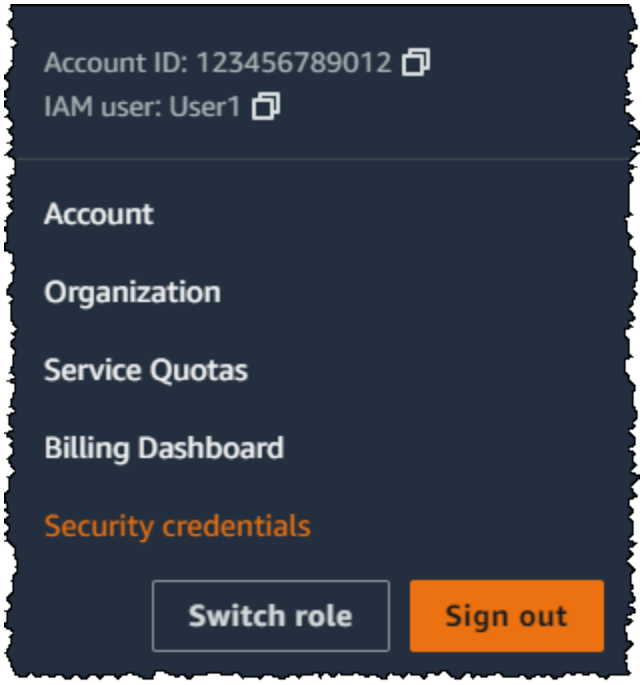
1. Nel pannello di navigazione seleziona Gruppi di utenti.
2. Nella casella di ricerca digita **`EC2MFA`** e seleziona il nome del gruppo (non la casella di controllo) nell'elenco.
3. Nella scheda Permissions (Autorizzazioni), scegli Add permissions (Aggiungi autorizzazioni), quindi Attach policies (Collega policy).
4. Nella pagina Allega criteri di autorizzazione al gruppo EC2 MFA, nella casella di ricerca, digitare. **`EC2Full`** Quindi seleziona la casella di controllo accanto ad Amazon EC2 FullAccess nell'elenco. Non salvare ancora le modifiche.
5. Nella casella di ricerca, digita **`Force`** e seleziona la casella di controllo accanto a Force_MFA nell'elenco.

6. Scegli Collega policy.

Fase 3: test dell'accesso dell'utente

In questa parte del tutorial, effettuare l'accesso come utente di prova e verificare che la policy funzioni correttamente.

1. Accedi al tuo annuncio **`MFAUser`** con Account AWS la password che hai assegnato nella sezione precedente. Utilizzare l'URL: `https://<alias or account ID number>.signin.aws.amazon.com/console`
2. Seleziona EC2 per aprire la console Amazon EC2 e verifica che l'utente non disponga di autorizzazioni per effettuare alcuna operazione.
3. Selezionare il nome utente `MFAUser` in alto a destra nella barra di navigazione e scegli Security Credentials (Credenziali di sicurezza).



4. Aggiungere un dispositivo MFA. Nella sezione Multi-Factor Authentication (MFA) (Autenticazione a più fattori), scegliere Assign MFA device (Assegna dispositivo MFA).

> **Note**
>
> Potrebbe essere visualizzato un errore che indica che non si è autorizzati a eseguire `iam:DeleteVirtualMFADevice`. Questo può accadere se qualcuno in precedenza ha iniziato ad assegnare un dispositivo MFA virtuale a questo utente e ha annullato

il processo. Per continuare, l'utente o un altro amministratore devono eliminare il dispositivo MFA virtuale esistente non assegnato dell'utente. Per ulteriori informazioni, consulta Non sono autorizzato a eseguire: iam: DeleteVirtualMFADevice.

5. Per questo tutorial, utilizziamo un dispositivo MFA (basato su software), ad esempio Google Authenticator app su un cellulare. Scegli l'app Authenticator, quindi fai clic su Next (Successivo).

   IAM genera e visualizza le informazioni di configurazione per il dispositivo MFA virtuale, tra cui il codice grafico QR. Il grafico è una rappresentazione della chiave di configurazione segreta che è disponibile per l'inserimento manuale su dispositivi che non supportano i codici QR.

6. Aprire l'app MFA virtuale. (Per un elenco di app che si possono utilizzare per ospitare dispositivi MFA virtuali, consulta Applicazioni MFA virtuali.) Se l'app MFA virtuale supporta più account (più dispositivi MFA virtuali), selezionare l'opzione che consente di creare un nuovo account (un nuovo dispositivo virtuale MFA).

7. Determinare se l'app MFA supporta i codici QR e procedere in uno dei seguenti modi:

   - Nella procedura guidata, scegliere Show QR code (Mostra codice QR). Quindi utilizzare l'app per la scansione del codice QR. Ad esempio, è possibile selezionare l'icona della fotocamera o un'opzione simile a Scannerizza codice ed eseguire la scansione del codice tramite la fotocamera del dispositivo.
   - Nella procedura guidata Set up device (Configura dispositivo), scegli Show secret key (Mostra chiave segreta) e digita la chiave segreta nell'app MFA.

   Al termine, il dispositivo MFA virtuale avvia la generazione di password una tantum.

8. Nella procedura guidata Set up device (Configura dispositivo), nella casella Enter the code from your authenticator app (Immetti il codice dall'app di autenticazione), digita la password una tantum che appare nel dispositivo MFA virtuale. Scegli Register MFA (Registra MFA).

   Important

   Invia la richiesta immediatamente dopo la generazione del codice. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, il dispositivo MFA è correttamente associato all'utente. Tuttavia, il dispositivo MFA non è sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo MFA virtuale è ora pronto per l'uso con. AWS

9. Uscire dalla console ed effettuare nuovamente l'accesso come **`MFAUser`** Questa volta AWS ti viene richiesto un codice MFA dal tuo telefono. Una volta ottenuto, digitare il codice nella casella e selezionare Submit (Invia).
10. Seleziona EC2 per aprire nuovamente la console Amazon EC2. In questo momento è possibile visualizzare tutte le informazioni ed eseguire tutte le azioni desiderate. Se si accede a qualsiasi altra console come questo utente, vengono visualizzati messaggi di accesso negato. Il motivo è che le policy in questo tutorial concedono l'accesso solo a Amazon EC2.

Risorse correlate

Per ulteriori informazioni, consulta i seguenti argomenti:

- AWS Multi-factor autenticazione in IAM
- Accesso abilitato con MFA

Tutorial IAM: usa un CloudFormation modello per creare un SAML Identity Provider (IdP)

Per configurare la federazione SAML per il tuo AWS account, devi creare un SAML Identity Provider (IdP). Questo tutorial mostra come utilizzare un CloudFormation modello per creare un IdP SAML che stabilisca la fiducia tra e il tuo IdP esterno. AWS

Il modello crea un IdP SAML configurato con il documento di metadati dell'IdP. I ruoli IAM federati possono quindi fare riferimento a questo IdP per consentire agli utenti autenticati del tuo IdP esterno di accedere alle risorse. AWS

La risorsa implementata è costituita da un IdP SAML configurato con il documento di metadati del tuo IdP e impostazioni di crittografia facoltative.

Prerequisiti

Questo tutorial presuppone che tu abbia a disposizione quanto segue:

- Python 3.6 o versione successiva installato sul computer locale per eseguire il comando Python utilizzato in questo tutorial per formattare il file XML dei metadati SAML del tuo IdP.

- Un documento di metadati SAML del tuo IdP esterno salvato come file XML.

Crea un IdP SAML utilizzando CloudFormation

Per creare l'IdP SAML, crea un CloudFormation modello e utilizzalo per creare uno stack contenente la risorsa IdP.

Creazione del modello

Innanzitutto, crea il modello. CloudFormation

1. Nella Modello sezione, scegli l'icona di copia nella scheda JSON o YAML per copiare il contenuto del modello.
2. Copia i contenuti del modello in un nuovo file.
3. Salva il file localmente.

Creazione dello stack

Quindi, usa il modello che hai salvato per eseguire il provisioning di uno stack. CloudFormation

1. Apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Nella pagina Stack, scegli Con nuove risorse (standard) nel menu Crea stack.
3. Specifica il modello:
    a. In Prerequisito, seleziona Scegli un modello esistente.
    b. In Specifica modello, seleziona Carica un file di modello.
    c. Seleziona Scegli file, quindi passa al file del modello e selezionalo.
    d. Scegli Next (Successivo).
4. Specifica i seguenti dettagli dello stack:
    a. Inserisci un nome dello stack.
    b. Puoi lasciare questo campo vuoto per IdentityProviderName generare automaticamente un nome in base al nome dello stack o inserire un nome personalizzato per il tuo IdP SAML. I nomi personalizzati devono contenere solo caratteri alfanumerici, punti, trattini bassi e trattini.

c. Per IdentityProviderSAMLMetadataDocument, devi formattare il file XML di metadati SAML come una singola riga prima di incollarlo in questo campo. Ciò è necessario perché la CloudFormation console richiede che il contenuto XML venga formattato come una singola riga quando viene passato attraverso i parametri della console.

Usa il seguente comando Python per riformattare il tuo file XML:

```
python3 -c "import sys, re; content=open(sys.argv[1]).read(); print(re.sub(r'>
\s+<', '><', content.replace('\n', '').replace('\r', '').strip()))" saml-
metadata.xml
```

> **Note**
>
> Il documento di metadati SAML dell'IdP deve essere formattato come una singola riga per poter essere utilizzato come parametro nella console. Il comando Python rimuove le interruzioni di riga e gli spazi bianchi aggiuntivi per creare il formato richiesto, mantenendo però intatti tutti i contenuti e la struttura originali.

Copia l'output del comando Python e incollalo nel IdentityProviderSAMLMetadataDocument campo.

Esempio di documento di metadati SAML formattato (abbreviato):

```
<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
 entityID="https://portal.sso.example.com/saml/assertion/
CompanyIdP"><md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"><md:KeyDescriptor
 use="signing"><ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/
xmldsig#"><ds:X509Data><ds:X509Certificate>MIIDXTCCAkWgAwIBAgIJAJC1HiIAZAiIMA0GCSqGSIb3
ds:X509Certificate></ds:X509Data></ds:KeyInfo></
md:KeyDescriptor><md:SingleLogoutService
 Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
 Location="https://portal.sso.example.com/saml/logout/
CompanyIdP"/><md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent</md:NameIDFormat><md:SingleSignOnService
 Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://
```

```
portal.sso.example.com/saml/assertion/CompanyIdP"/></md:IDPSSODescriptor></
md:EntityDescriptor>
```

d. Per gli altri parametri, accetta i valori predefiniti o inseriscine altri in base ai tuoi requisiti:

- IdentityProviderAddPrivateKey- Chiave privata opzionale per decrittografare le asserzioni SAML
- IdentityProviderAssertionEncryptionMode- Facoltativo, imposta la modalità di crittografia per le asserzioni SAML (consentita, obbligatoria o vuota)

e. Scegli Next (Successivo).

5. Configura le opzioni dello stack:

a. In Opzioni di errore dello stack, scegli Elimina tutte le nuove risorse create.

> **Note**
>
> Scegliendo questa opzione si evita di ricevere addebiti per risorse la cui policy di eliminazione specifica che devono essere mantenute anche se la creazione dello stack non riesce.

b. Accetta tutti gli altri valori predefiniti.

c. In Capacità, seleziona la casella per confermare che ciò CloudFormation potrebbe creare risorse IAM nel tuo account.

d. Scegli Next (Successivo).

6. Rivedi i dettagli dello stack e scegli Invia.

CloudFormation crea lo stack. Una volta completata la creazione dello stack, le risorse dello stack sono pronte per l'uso. Puoi utilizzare la scheda Risorse nella pagina dei dettagli dello stack per visualizzare le risorse che sono state fornite nel tuo account.

Lo stack produrrà i seguenti valori, che puoi visualizzare nella scheda Output:

- ProviderARN: l'ARN dell'IdP SAML creato (ad esempio, `arn:aws:iam::123456789012:saml-provider/CompanyIdP`). Avrai bisogno di questo ARN per creare ruoli che considerano attendibile questo provider.
- ProviderName: il nome dell'IdP SAML creato (ad esempio, `CompanyIdP` se hai specificato un nome personalizzato o `my-saml-stack-saml-provider` se hai utilizzato il nome predefinito).

Questi output vengono anche esportati, consentendone l'importazione da altri stack CloudFormation utilizzando la funzione `Fn::ImportValue`.

Verifica dell'IdP SAML

Dopo aver creato l'IdP SAML, puoi verificarne la configurazione e annotare il relativo ARN da utilizzare con i ruoli federati.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Identity providers (Provider di identità).

   L'elenco dovrebbe mostrare l'IdP SAML appena creato.
3. Scegli il nome dell'IdP per visualizzarne i dettagli.

   Nella pagina dei dettagli dell'IdP, puoi vedere il documento di metadati SAML e altri dettagli di configurazione.
4. Prendi nota del Provider ARN visualizzato nella pagina dei dettagli.

   Avrai bisogno di questo ARN per creare ruoli IAM federati che ritengono attendibile questo IdP.
5. Rivedi il documento di metadati per assicurarti che corrisponda a quanto fornito dal tuo IdP esterno.

Il tuo IdP SAML è ora pronto per essere utilizzato dai ruoli IAM federati. Puoi creare ruoli che considerino attendibile questo IdP per consentire agli utenti autenticati del tuo IdP esterno di assumere tali ruoli e accedere alle risorse. AWS

Pulizia: eliminazione delle risorse

Come passaggio finale, eliminerai lo stack e le risorse in esso contenute.

1. Apri la console. CloudFormation
2. Nella pagina Stack, scegli lo stack creato dal modello, poi scegli Elimina, quindi conferma con Elimina.

   CloudFormation avvia l'eliminazione dello stack e di tutte le risorse che include.

CloudFormation dettagli del modello

Resources

Il CloudFormation modello per questo tutorial creerà la seguente risorsa nel tuo account:

`AWS::IAM::SAMLProvider`: un IdP SAML che stabilisce la fiducia tra AWS e il tuo IdP esterno.

Configurazione

Il modello include i seguenti parametri configurabili:

- IdentityProviderName- Il nome del tuo IdP SAML (lascia vuoto il campo per il nome generato automaticamente)

  Esempio: `CompanyIdP` o `EnterpriseSSO`
- IdentityProviderSAMLMetadataDocument- Il documento di metadati SAML del tuo IdP esterno (formattato come riga singola)
- IdentityProviderAddPrivateKey- Chiave privata opzionale per decrittografare le asserzioni SAML
- IdentityProviderAssertionEncryptionMode- Facoltativo, imposta la modalità di crittografia per le asserzioni SAML

CloudFormation modello

Salva il seguente codice JSON o YAML come file separato da utilizzare come CloudFormation modello per questo tutorial.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "[AWSDocs] IAM: tutorial_saml-idp",
  "Parameters": {
    "IdentityProviderName": {
      "Type": "String",
      "Description": "Name of the SAML Identity Provider (leave empty for auto-
generated name like '{StackName}-{UniqueId}')",
      "Default": "",
      "AllowedPattern": "^$|^[a-zA-Z0-9._-]+$",
```

```
      "ConstraintDescription": "Must be empty or contain only alphanumeric
characters, periods, underscores, and hyphens"
    },
    "IdentityProviderSAMLMetadataDocument": {
      "Type": "String",
      "Description": "SAML metadata document from identity provider"
    },
    "IdentityProviderAddPrivateKey": {
      "Type": "String",
      "Description": "Optional private key for decrypting SAML assertions. The
private key must be a .pem file that uses AES-GCM or AES-CBC encryption algorithm
to decrypt SAML assertions.",
      "Default": ""
    },
    "IdentityProviderAssertionEncryptionMode": {
      "Type": "String",
      "Description": "Optional, sets encryption mode for SAML assertions",
      "Default": "",
      "AllowedValues": ["", "Allowed", "Required"]
    }
  },
  "Conditions": {
    "HasPrivateKey": {"Fn::Not": [{"Fn::Equals": [{"Ref":
"IdentityProviderAddPrivateKey"}, ""]}]},
    "HasEncryptionMode": {"Fn::Not": [{"Fn::Equals": [{"Ref":
"IdentityProviderAssertionEncryptionMode"}, ""]}]},
    "HasCustomName": {"Fn::Not": [{"Fn::Equals": [{"Ref": "IdentityProviderName"},
""]}]}
  },
  "Resources": {
    "SAMLProvider": {
      "Type": "AWS::IAM::SAMLProvider",
      "Properties": {
        "Name": {"Fn::If": ["HasCustomName", {"Ref": "IdentityProviderName"},
{"Ref": "AWS::NoValue"}]},
        "SamlMetadataDocument": {"Ref": "IdentityProviderSAMLMetadataDocument"},
        "Tags": [
          {
            "Key": "Name",
            "Value": {"Fn::If": ["HasCustomName", {"Ref": "IdentityProviderName"},
{"Fn::Sub": "${AWS::StackName}-saml-provider"}]}
          }
        ],
```

```
        "AddPrivateKey": {"Fn::If": ["HasPrivateKey", {"Ref":
 "IdentityProviderAddPrivateKey"}, {"Ref": "AWS::NoValue"}]},
        "AssertionEncryptionMode": {"Fn::If": ["HasEncryptionMode", {"Ref":
 "IdentityProviderAssertionEncryptionMode"}, {"Ref": "AWS::NoValue"}]}
      }
    }
  },
  "Outputs": {
    "ProviderARN": {
      "Description": "ARN of the created SAML Identity Provider",
      "Value": {"Ref": "SAMLProvider"},
      "Export": {
        "Name": {"Fn::Sub": "${AWS::StackName}-ProviderARN"}
      }
    },
    "ProviderName": {
      "Description": "Name of the SAML Identity Provider",
      "Value": {"Fn::If": ["HasCustomName", {"Ref": "IdentityProviderName"},
 {"Fn::Sub": "${AWS::StackName}-saml-provider"}]},
      "Export": {
        "Name": {"Fn::Sub": "${AWS::StackName}-ProviderName"}
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Description: '[AWSDocs] IAM: tutorial_saml-idp'

Parameters:
  IdentityProviderName:
    Type: String
    Description: Name of the SAML Identity Provider (leave empty for auto-generated
 name like '{StackName}-{UniqueId}')
    Default: ""
    AllowedPattern: '^$|^[a-zA-Z0-9._-]+$'
    ConstraintDescription: 'Must be empty or contain only alphanumeric characters,
 periods, underscores, and hyphens'

  IdentityProviderSAMLMetadataDocument:
    Type: String
```

```
    Description: SAML metadata document from identity provider

  IdentityProviderAddPrivateKey:
    Type: String
    Description: Optional private key for decrypting SAML assertions. The private
 key must be a .pem file that uses AES-GCM or AES-CBC encryption algorithm to
 decrypt SAML assertions.
    Default: ""

  IdentityProviderAssertionEncryptionMode:
    Type: String
    Description: Optional, sets encryption mode for SAML assertions
    Default: ""
    AllowedValues:
      - ""
      - "Allowed"
      - "Required"

Conditions:
  HasPrivateKey: !Not [!Equals [!Ref IdentityProviderAddPrivateKey, ""]]
  HasEncryptionMode: !Not [!Equals [!Ref IdentityProviderAssertionEncryptionMode,
 ""]]
  HasCustomName: !Not [!Equals [!Ref IdentityProviderName, ""]]

Resources:
  SAMLProvider:
    Type: 'AWS::IAM::SAMLProvider'
    Properties:
      Name: !If
        - HasCustomName
        - !Ref IdentityProviderName
        - !Ref AWS::NoValue
      SamlMetadataDocument: !Ref IdentityProviderSAMLMetadataDocument
      Tags:
        - Key: Name
          Value: !If
            - HasCustomName
            - !Ref IdentityProviderName
            - !Sub '${AWS::StackName}-saml-provider'
      AddPrivateKey: !If
        - HasPrivateKey
        - !Ref IdentityProviderAddPrivateKey
        - !Ref AWS::NoValue
      AssertionEncryptionMode: !If
```

```
        - HasEncryptionMode
        - !Ref IdentityProviderAssertionEncryptionMode
        - !Ref AWS::NoValue

Outputs:
  ProviderARN:
    Description: 'ARN of the created SAML Identity Provider'
    Value: !Ref SAMLProvider
    Export:
      Name: !Sub '${AWS::StackName}-ProviderARN'

  ProviderName:
    Description: 'Name of the SAML Identity Provider'
    Value: !If
      - HasCustomName
      - !Ref IdentityProviderName
      - !Sub '${AWS::StackName}-saml-provider'
    Export:
      Name: !Sub '${AWS::StackName}-ProviderName'
```

Tutorial IAM: usa un CloudFormation modello per creare un ruolo IAM federato SAML

Se disponi di un provider di identità SAML (IdP) esistente configurato nel AWS tuo account, puoi creare ruoli IAM federati che si fidano di quell'IdP. Questo tutorial mostra come utilizzare un CloudFormation modello per creare un ruolo IAM federato SAML che può essere assunto dagli utenti autenticati tramite il tuo IdP esterno.

Il modello crea un ruolo IAM federato con una policy di attendibilità che consente all'IdP SAML di assumere tale ruolo. Gli utenti autenticati dal tuo IdP esterno possono assumere questo ruolo per accedere alle risorse AWS in base alle autorizzazioni associate al ruolo.

La risorsa implementata include i seguenti elementi:

- Un ruolo IAM federato che ritiene attendibile il tuo IdP SAML esistente.
- Policy gestite configurabili che possono essere associate al ruolo per concedere autorizzazioni specifiche.
- Impostazioni facoltative del limite delle autorizzazioni e della durata della sessione.

Prerequisiti

Questo tutorial presuppone che tu abbia a disposizione quanto segue:

- Un IdP SAML esistente configurato nel AWS tuo account. Se non ne possiedi già uno, puoi crearlo utilizzando il tutorial Tutorial IAM: usa un CloudFormation modello per creare un SAML Identity Provider (IdP).
- L'ARN del tuo IdP SAML, che dovrai specificare come parametro durante la creazione dello stack.
- Python 3.6 o versione successiva installato sul computer locale per eseguire il comando Python utilizzato in questo tutorial per formattare il file XML dei metadati SAML del tuo IdP.

Crea un ruolo federato SAML utilizzando CloudFormation

Per creare il ruolo federato SAML, creerai un CloudFormation modello e lo utilizzerai per creare uno stack contenente il ruolo.

Creazione del modello

Per prima cosa, crea il modello. CloudFormation

1. Nella sezione Modello, fai clic sull'icona di copia nella scheda JSON o YAML per copiare i contenuti del modello.
2. Copia i contenuti del modello in un nuovo file.
3. Salva il file localmente.

Creazione dello stack

Quindi, usa il modello che hai salvato per effettuare il provisioning di uno CloudFormation stack.

1. Apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Nella pagina Stack, scegli Con nuove risorse (standard) nel menu Crea stack.
3. Specifica il modello:
    a. In Prerequisito, seleziona Scegli un modello esistente.
    b. In Specifica modello, seleziona Carica un file di modello.
    c. Seleziona Scegli file, quindi passa al file del modello e selezionalo.

d. Scegli Next (Successivo).

4. Specifica i seguenti dettagli dello stack:

   a. Inserisci un nome dello stack.

   b. Per SAMLProviderARN, inserisci l'ARN del tuo IdP SAML esistente. Dovrebbe essere nel formato `arn:aws:iam::123456789012:saml-provider/YourProviderName`.

      Ad esempio: `arn:aws:iam::123456789012:saml-provider/CompanyIdP`

      > **Note**
      >
      > Se hai creato il tuo IdP SAML utilizzando Tutorial IAM: usa un CloudFormation modello per creare un SAML Identity Provider (IdP) il tutorial, puoi trovare l'ARN del provider nella scheda Output di quello stack. CloudFormation

   c. Infatti RoleName, puoi lasciare vuoto questo campo per generare automaticamente un nome basato sul nome dello stack o inserire un nome personalizzato per il ruolo IAM.

      Esempio: `SAML-Developer-Access` o `SAML-ReadOnly-Role`

   d. Per gli altri parametri, accetta i valori predefiniti o inseriscine altri in base ai tuoi requisiti:

      - RoleSessionDuration- Durata massima della sessione in secondi (3600-43200, impostazione predefinita 7200)

        Esempio: `14400` (4 ore)

      - RolePermissionsBoundary- ARN opzionale di una politica sui limiti delle autorizzazioni

        Ad esempio: `arn:aws:iam::123456789012:policy/DeveloperBoundary`

      - RolePath- Percorso per il ruolo IAM (l'impostazione predefinita è/)

        Ad esempio: `/saml-roles/`

      - ManagedPolicy1-5- ARN opzionali con un massimo di 5 policy gestite da allegare

        Esempio per ManagedPolicy 1: `arn:aws:iam::aws:policy/ReadOnlyAccess`

        Esempio per ManagedPolicy 2: `arn:aws:iam::123456789012:policy/CustomPolicy`

   e. Scegli Next (Successivo).

5. Configura le opzioni dello stack:

   a. In Opzioni di errore dello stack, scegli Elimina tutte le nuove risorse create.

   > **Note**
   >
   > Scegliendo questa opzione si evita di ricevere addebiti per risorse la cui policy di eliminazione specifica che devono essere mantenute anche se la creazione dello stack non riesce.

   b. Accetta tutti gli altri valori predefiniti.

   c. In Capacità, seleziona la casella per confermare che CloudFormation potresti creare risorse IAM nel tuo account.

   d. Scegli Next (Successivo).

6. Rivedi i dettagli dello stack e scegli Invia.

CloudFormation crea lo stack. Una volta completata la creazione dello stack, le relative risorse sono pronte per l'uso. Puoi utilizzare la scheda Risorse nella pagina dei dettagli dello stack per visualizzare le risorse che sono state fornite nel tuo account.

Lo stack produrrà il seguente valore, che puoi visualizzare nella scheda Output:

- RoleARN: l'ARN del ruolo IAM creato (ad esempio, `arn:aws:iam::123456789012:role/SAML-Developer-Access` o `arn:aws:iam::123456789012:role/stack-name-a1b2c3d4` se si utilizza un nome generato automaticamente).

Avrai bisogno di questo ARN del ruolo quando configuri il tuo IdP per inviare gli attributi SAML appropriati per l'assunzione del ruolo.

Prova del ruolo federato SAML

Una volta creato il ruolo federato SAML, puoi verificarne la configurazione e testare l'impostazione della federazione.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Trova e scegli il ruolo federato appena creato.

Se hai fornito un nome di ruolo personalizzato, cerca quel nome. Se hai lasciato il RoleName parametro vuoto, il ruolo avrà un nome generato automaticamente basato sul nome dello stack e un identificatore univoco.

4. Scegli la scheda Relazioni di attendibilità per rivedere la policy di attendibilità.

   La policy di attendibilità dovrebbe mostrare che il tuo IdP SAML è affidabile per assumere questo ruolo a condizione che il pubblico SAML (`SAML:aud`) corrisponda a `https://signin.aws.amazon.com/saml`.

5. Scegli la scheda Autorizzazioni per esaminare le policy collegate.

   Puoi vedere tutte le policy gestite collegate al ruolo durante la creazione.

6. Prendi nota dell'ARN del ruolo visualizzato nella pagina di riepilogo del ruolo.

   Questo ARN ti servirà per configurare il tuo IdP esterno e consentire agli utenti di assumere questo ruolo.

Il tuo ruolo federato SAML è ora pronto per essere utilizzato. Configura il tuo IdP esterno per includere l'ARN di questo ruolo nelle asserzioni SAML e gli utenti autenticati potranno assumere questo ruolo per accedere alle risorse. AWS

Pulizia: eliminazione delle risorse

Come passaggio finale, eliminerai lo stack e le risorse in esso contenute.

1. Apri la CloudFormation console.
2. Nella pagina Stack, scegli lo stack creato dal modello, poi scegli Elimina, quindi conferma con Elimina.

   CloudFormation avvia l'eliminazione dello stack e di tutte le risorse che include.

CloudFormation dettagli del modello

Resources

Il CloudFormation modello di questo tutorial creerà la seguente risorsa nel tuo account:

- `AWS::IAM::Role`: un ruolo IAM federato che può essere assunto dagli utenti autenticati tramite il tuo IdP SAML.

Configurazione

Il modello include i seguenti parametri configurabili:

- RoleName- Nome del ruolo IAM (lasciare vuoto il campo per il nome generato automaticamente)
- SAMLProviderARN: ARN dell'IdP SAML (obbligatorio)
- RoleSessionDuration- Durata massima della sessione in secondi (3600-43200, impostazione predefinita 7200)
- RolePermissionsBoundary- ARN opzionale della politica dei limiti delle autorizzazioni
- RolePath- Percorso per il ruolo IAM (default/)
- ManagedPolicy1-5- ARN opzionali con un massimo di 5 policy gestite da allegare

CloudFormation modello

Salva il seguente codice JSON o YAML come file separato da utilizzare come CloudFormation modello per questo tutorial.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "[AWSDocs] IAM: tutorial_saml-federated-role",
  "Parameters": {
    "RoleName": {
      "Type": "String",
      "Description": "Name of the IAM Role (leave empty for auto-generated name like
 '{StackName}-{UniqueId}')",
      "Default": "",
      "AllowedPattern": "^$|^[\\w+=,.@-]{1,64}$",
      "ConstraintDescription": "Must be empty or 1-64 characters and can contain
 alphanumeric characters and +=,.@-"
    },
    "SAMLProviderARN": {
      "Type": "String",
      "Description": "ARN of the SAML Identity Provider",
      "AllowedPattern": "^arn:aws:iam::\\d{12}:saml-provider/[a-zA-Z0-9._-]+$",
```

```
      "ConstraintDescription": "Must be a valid SAML provider ARN"
    },
    "RoleSessionDuration": {
      "Type": "Number",
      "Description": "The maximum session duration (in seconds) that you want to set
for the specified role (3600-43200)",
      "MinValue": 3600,
      "MaxValue": 43200,
      "Default": 7200
    },
    "RolePermissionsBoundary": {
      "Type": "String",
      "Description": "Optional ARN of the permissions boundary policy (leave empty
for none)",
      "Default": ""
    },
    "RolePath": {
      "Type": "String",
      "Description": "Path for the IAM role (must start and end with /)",
      "Default": "/",
      "AllowedPattern": "^\/.*\/$|^\/$",
      "ConstraintDescription": "Role path must start and end with forward slash (/)"
    },
    "RoleManagedPolicy1": {
      "Type": "String",
      "Description": "Optional managed policy ARN 1",
      "Default": ""
    },
    "RoleManagedPolicy2": {
      "Type": "String",
      "Description": "Optional managed policy ARN 2",
      "Default": ""
    },
    "RoleManagedPolicy3": {
      "Type": "String",
      "Description": "Optional managed policy ARN 3",
      "Default": ""
    },
    "RoleManagedPolicy4": {
      "Type": "String",
      "Description": "Optional managed policy ARN 4",
      "Default": ""
    },
    "RoleManagedPolicy5": {
```

```
      "Type": "String",
      "Description": "Optional managed policy ARN 5",
      "Default": ""
    }
  },
  "Conditions": {
    "HasCustomRoleName": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleName"}, ""]}]},
    "HasPermissionsBoundary": {"Fn::Not": [{"Fn::Equals": [{"Ref":
"RolePermissionsBoundary"}, ""]}]},
    "HasPolicy1": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy1"},
""]}]},
    "HasPolicy2": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy2"},
""]}]},
    "HasPolicy3": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy3"},
""]}]},
    "HasPolicy4": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy4"},
""]}]},
    "HasPolicy5": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy5"}, ""]}]}
  },
  "Resources": {
    "SAMLFederatedRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": {"Fn::If": ["HasCustomRoleName", {"Ref": "RoleName"}, {"Ref":
"AWS::NoValue"}]},
        "Description": "IAM role with SAML provider trust",
        "MaxSessionDuration": {"Ref": "RoleSessionDuration"},
        "PermissionsBoundary": {"Fn::If": ["HasPermissionsBoundary", {"Ref":
"RolePermissionsBoundary"}, {"Ref": "AWS::NoValue"}]},
        "Path": {"Ref": "RolePath"},
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Federated": {"Ref": "SAMLProviderARN"}
              },
              "Action": "sts:AssumeRoleWithSAML",
              "Condition": {
                "StringEquals": {
                  "SAML:aud": "https://signin.aws.amazon.com/saml"
                }
              }
```

```
            }
          ]
        },
        "ManagedPolicyArns": {
          "Fn::Split": [
            ",",
            {
              "Fn::Join": [
                ",",
                [
                  {"Fn::If": ["HasPolicy1", {"Ref": "RoleManagedPolicy1"}, {"Ref":
 "AWS::NoValue"}]},
                  {"Fn::If": ["HasPolicy2", {"Ref": "RoleManagedPolicy2"}, {"Ref":
 "AWS::NoValue"}]},
                  {"Fn::If": ["HasPolicy3", {"Ref": "RoleManagedPolicy3"}, {"Ref":
 "AWS::NoValue"}]},
                  {"Fn::If": ["HasPolicy4", {"Ref": "RoleManagedPolicy4"}, {"Ref":
 "AWS::NoValue"}]},
                  {"Fn::If": ["HasPolicy5", {"Ref": "RoleManagedPolicy5"}, {"Ref":
 "AWS::NoValue"}]}
                ]
              ]
            }
          ]
        }
      }
    }
  },
  "Outputs": {
    "RoleARN": {
      "Description": "ARN of the created IAM Role",
      "Value": {"Fn::GetAtt": ["SAMLFederatedRole", "Arn"]},
      "Export": {
        "Name": {"Fn::Sub": "${AWS::StackName}-RoleARN"}
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Description: '[AWSDocs] IAM: tutorial_saml-federated-role'
```

```
Parameters:
  RoleName:
    Type: String
    Description: 'Name of the IAM Role (leave empty for auto-generated name like
 ''{StackName}-{UniqueId}'')'
    Default: ""
    AllowedPattern: '^$|^[\w+=,.@-]{1,64}$'
    ConstraintDescription: 'Must be empty or 1-64 characters and can contain
 alphanumeric characters and +=,.@-'

  SAMLProviderARN:
    Type: String
    Description: 'ARN of the SAML Identity Provider'
    AllowedPattern: '^arn:aws:iam::\d{12}:saml-provider/[a-zA-Z0-9._-]+$'
    ConstraintDescription: 'Must be a valid SAML provider ARN'

  RoleSessionDuration:
    Type: Number
    Description: 'The maximum session duration (in seconds) that you want to set for
 the specified role (3600-43200)'
    MinValue: 3600
    MaxValue: 43200
    Default: 7200

  RolePermissionsBoundary:
    Type: String
    Description: Optional ARN of the permissions boundary policy (leave empty for
 none)
    Default: ""

  RolePath:
    Type: String
    Description: 'Path for the IAM role (must start and end with /)'
    Default: "/"
    AllowedPattern: '^\/.*\/$|^\/$'
    ConstraintDescription: 'Role path must start and end with forward slash (/)'

  RoleManagedPolicy1:
    Type: String
    Description: Optional managed policy ARN 1
    Default: ""
  RoleManagedPolicy2:
    Type: String
```

```
    Description: Optional managed policy ARN 2
    Default: ""
  RoleManagedPolicy3:
    Type: String
    Description: Optional managed policy ARN 3
    Default: ""
  RoleManagedPolicy4:
    Type: String
    Description: Optional managed policy ARN 4
    Default: ""
  RoleManagedPolicy5:
    Type: String
    Description: Optional managed policy ARN 5
    Default: ""

Conditions:
  HasCustomRoleName: !Not [!Equals [!Ref RoleName, ""]]
  HasPermissionsBoundary: !Not [!Equals [!Ref RolePermissionsBoundary, ""]]
  HasPolicy1: !Not [!Equals [!Ref RoleManagedPolicy1, ""]]
  HasPolicy2: !Not [!Equals [!Ref RoleManagedPolicy2, ""]]
  HasPolicy3: !Not [!Equals [!Ref RoleManagedPolicy3, ""]]
  HasPolicy4: !Not [!Equals [!Ref RoleManagedPolicy4, ""]]
  HasPolicy5: !Not [!Equals [!Ref RoleManagedPolicy5, ""]]

Resources:
  SAMLFederatedRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: !If
        - HasCustomRoleName
        - !Ref RoleName
        - !Ref AWS::NoValue
      Description: 'IAM role with SAML provider trust'
      MaxSessionDuration: !Ref RoleSessionDuration
      PermissionsBoundary: !If
        - HasPermissionsBoundary
        - !Ref RolePermissionsBoundary
        - !Ref AWS::NoValue
      Path: !Ref RolePath
      AssumeRolePolicyDocument:
        Version: '2012-10-17       '
        Statement:
          - Effect: Allow
            Principal:
```

```
              Federated: !Ref SAMLProviderARN
            Action: 'sts:AssumeRoleWithSAML'
            Condition:
              StringEquals:
                'SAML:aud': 'https://signin.aws.amazon.com/saml'
      ManagedPolicyArns:
        !Split
          - ','
          - !Join
            - ','
            - - !If [HasPolicy1, !Ref RoleManagedPolicy1, !Ref 'AWS::NoValue']
              - !If [HasPolicy2, !Ref RoleManagedPolicy2, !Ref 'AWS::NoValue']
              - !If [HasPolicy3, !Ref RoleManagedPolicy3, !Ref 'AWS::NoValue']
              - !If [HasPolicy4, !Ref RoleManagedPolicy4, !Ref 'AWS::NoValue']
              - !If [HasPolicy5, !Ref RoleManagedPolicy5, !Ref 'AWS::NoValue']

Outputs:
  RoleARN:
    Description: 'ARN of the created IAM Role'
    Value: !GetAtt SAMLFederatedRole.Arn
    Export:
      Name: !Sub '${AWS::StackName}-RoleARN'
```

Tutorial IAM: usa un CloudFormation modello per creare un ruolo IAM federato SAML Identity Provider (IdP) e SAML

Per acquisire familiarità con la federazione SAML e le sue funzionalità, utilizzerai un CloudFormation modello per configurare un SAML Identity Provider (IdP) e il ruolo IAM federato associato. Questo tutorial mostra come creare entrambe le risorse in un unico stack.

Il modello crea un IdP SAML che può essere utilizzato per l'accesso federato alle AWS risorse, insieme a un ruolo IAM che si fida del provider SAML. Gli utenti autenticati dal tuo IdP esterno possono assumere questo ruolo per accedere alle risorse. AWS

Le risorse implementate includono i seguenti elementi:

- Un IdP SAML configurato con il documento di metadati dell'IdP.
- Un ruolo IAM federato che ritiene attendibile l'IdP SAML e che può essere assunto dagli utenti autenticati.

- Policy gestite configurabili che possono essere associate al ruolo per concedere autorizzazioni specifiche.

Prerequisiti

Questo tutorial presuppone che tu abbia a disposizione quanto segue:

- Python 3.6 o versione successiva installato sul computer locale per eseguire il comando Python utilizzato in questo tutorial per formattare il file XML dei metadati SAML del tuo IdP.
- Un documento di metadati SAML del tuo IdP esterno salvato come file XML.

Crea un IdP e un ruolo SAML utilizzando CloudFormation

Per creare l'IdP SAML e il ruolo federato, creerai un CloudFormation modello e lo utilizzerai per creare uno stack contenente entrambe le risorse.

Creazione del modello

Innanzitutto, crea il modello. CloudFormation

1. Nella sezione Modello, fai clic sull'icona di copia nella scheda JSON o YAML per copiare i contenuti del modello.
2. Copia i contenuti del modello in un nuovo file.
3. Salva il file localmente.

Creazione dello stack

Quindi, usa il modello che hai salvato per eseguire il provisioning di uno CloudFormation stack.

1. Apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Nella pagina Stack, scegli Con nuove risorse (standard) nel menu Crea stack.
3. Specifica il modello:
    a. In Prerequisito, seleziona Scegli un modello esistente.
    b. In Specifica modello, seleziona Carica un file di modello.
    c. Seleziona Scegli file, quindi passa al file del modello e selezionalo.

d. Scegli Next (Successivo).

4. Specifica i seguenti dettagli dello stack:

   a. Inserisci un nome dello stack.

   b. Puoi lasciare questo campo vuoto per IdentityProviderName generare automaticamente un nome in base al nome dello stack o inserire un nome personalizzato per il tuo IdP SAML.

      Esempio: `CompanyIdP` o `EnterpriseSSO`

   c. Per IdentityProviderSAMLMetadataDocument, devi formattare il file XML di metadati SAML come una singola riga prima di incollarlo in questo campo. Ciò è necessario perché la CloudFormation console richiede che il contenuto XML venga formattato come una singola riga quando viene passato attraverso i parametri della console.

      Usa il seguente comando Python per riformattare il tuo file XML:

```
python3 -c "import sys, re; content=open(sys.argv[1]).read(); print(re.sub(r'>
\s+<', '><', content.replace('\n', '').replace('\r', '').strip()))" saml-
metadata.xml
```

> **Note**
>
> Il documento di metadati SAML dell'IdP deve essere formattato come una singola riga per poter essere utilizzato come parametro nella console. Il comando Python rimuove le interruzioni di riga e gli spazi bianchi aggiuntivi per creare il formato richiesto, mantenendo però intatti tutti i contenuti e la struttura originali.

      Copia l'output del comando Python e incollalo nel IdentityProviderSAMLMetadataDocument campo.

      Esempio di documento di metadati SAML formattato (abbreviato):

```
<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
 entityID="https://portal.sso.example.com/saml/assertion/
CompanyIdP"><md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"><md:KeyDescriptor
 use="signing"><ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/
xmldsig#"><ds:X509Data><ds:X509Certificate>MIIDXTCCAkWgAwIBAgIJAJC1HiIAZAiIMA0GCSqGSIb3
```

```
ds:X509Certificate></ds:X509Data></ds:KeyInfo></
md:KeyDescriptor><md:SingleLogoutService
 Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
 Location="https://portal.sso.example.com/saml/logout/
CompanyIdP"/><md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent</md:NameIDFormat><md:SingleSignOnService
 Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://
portal.sso.example.com/saml/assertion/CompanyIdP"/></md:IDPSSODescriptor></
md:EntityDescriptor>
```

d. Puoi lasciare questo campo vuoto per RoleName generare automaticamente un nome in base al nome dello stack o inserire un nome personalizzato per il ruolo IAM federato.

   Esempio: `SAML-Developer-Access` o `SAML-ReadOnly-Role`

e. Per gli altri parametri, accetta i valori predefiniti o inseriscine altri in base ai tuoi requisiti:

   - IdentityProviderAddPrivateKey- Chiave privata opzionale per decrittografare le asserzioni SAML
   - IdentityProviderAssertionEncryptionMode- Modalità di crittografia per le asserzioni SAML

     Valori di esempio: `Allowed`, `Required` o lasciare vuoto per non utilizzare alcuna crittografia
   - RoleSessionDuration- Durata massima della sessione in secondi (3600-43200, impostazione predefinita 7200)

     Esempio: `14400` (4 ore)
   - RolePermissionsBoundary- ARN opzionale di una politica di limitazione delle autorizzazioni

     Ad esempio: `arn:aws:iam::123456789012:policy/DeveloperBoundary`
   - RolePath- Percorso per il ruolo IAM (l'impostazione predefinita è/)

     Ad esempio: `/saml-roles/`
   - RoleManagedPolicy1-5- ARN opzionali con un massimo di 5 policy gestite da allegare

     Esempio per RoleManagedPolicy1: `arn:aws:iam::aws:policy/ReadOnlyAccess`

     Esempio per RoleManagedPolicy2: `arn:aws:iam::123456789012:policy/CustomPolicy`

f. Scegli Next (Successivo).

5. Configura le opzioni dello stack:

   a. In Opzioni di errore dello stack, scegli Elimina tutte le nuove risorse create.

   > **Note**
   >
   > Scegliendo questa opzione si evita di ricevere addebiti per risorse la cui policy di eliminazione specifica che devono essere mantenute anche se la creazione dello stack non riesce.

   b. Accetta tutti gli altri valori predefiniti.

   c. In Capacità, seleziona la casella per confermare che CloudFormation potrebbe creare risorse IAM nel tuo account.

   d. Scegli Next (Successivo).

6. Rivedi i dettagli dello stack e scegli Invia.

CloudFormation crea lo stack. Una volta completata la creazione dello stack, le relative risorse sono pronte per l'uso. Puoi utilizzare la scheda Risorse nella pagina dei dettagli dello stack per visualizzare le risorse che sono state fornite nel tuo account.

Lo stack produrrà i seguenti valori, che puoi visualizzare nella scheda Output:

- RoleARN: l'ARN del ruolo IAM creato (ad esempio, `arn:aws:iam::123456789012:role/SAML-Developer-Access` o `arn:aws:iam::123456789012:role/stack-name-a1b2c3d4` se si utilizza un nome generato automaticamente).
- IdentityProviderARN: l'ARN dell'IdP SAML creato (ad esempio,). `arn:aws:iam::123456789012:saml-provider/CompanyIdP`

Avrai bisogno di entrambi questi ARN durante la configurazione del tuo IdP per inviare gli attributi SAML appropriati per l'assunzione del ruolo.

Test della federazione SAML

Una volta che sono stati creati il ruolo federato e l'IdP SAML, puoi testare l'impostazione della federazione.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione, scegli Identity providers (Provider di identità).

   L'elenco dovrebbe mostrare l'IdP SAML appena creato.

3. Scegli il nome dell'IdP per visualizzarne i dettagli.

   Nella pagina dei dettagli dell'IdP, puoi vedere il documento di metadati SAML e altri dettagli di configurazione.

4. Nel riquadro di navigazione, seleziona Ruoli.

5. Trova e scegli il ruolo federato appena creato.

   Nella pagina dei dettagli del ruolo, puoi vedere la policy di attendibilità che consente all'IdP SAML di assumere questo ruolo.

6. Scegli la scheda Relazioni di attendibilità per rivedere la policy di attendibilità.

   La policy di attendibilità dovrebbe mostrare che l'IdP SAML è affidabile per assumere questo ruolo a condizione che il pubblico SAML (`SAML:aud`) corrisponda a `https://signin.aws.amazon.com/saml`.

Pulizia: eliminazione delle risorse

Come passaggio finale, eliminerai lo stack e le risorse in esso contenute.

1. Apri la console. CloudFormation

2. Nella pagina Stack, scegli lo stack creato dal modello, poi scegli Elimina, quindi conferma con Elimina.

   CloudFormation avvia l'eliminazione dello stack e di tutte le risorse che include.

CloudFormation dettagli del modello

Resources

Il CloudFormation modello per questo tutorial creerà le seguenti risorse nel tuo account:

- `AWS::IAM::SAMLProvider`: un IdP SAML che stabilisce la fiducia tra AWS e il tuo IdP esterno.
- `AWS::IAM::Role`: un ruolo IAM federato che può essere assunto dagli utenti autenticati tramite l'IdP SAML.

Configurazione

Il modello include i seguenti parametri configurabili:

- IdentityProviderName- Nome dell'IdP SAML (lascia vuoto il campo per il nome generato automaticamente)
- IdentityProviderSAMLMetadataDocument- Documento di metadati SAML fornito dal tuo IdP (obbligatorio)
- IdentityProviderAddPrivateKey- Chiave privata opzionale per decrittografare le asserzioni SAML
- IdentityProviderAssertionEncryptionMode- Modalità di crittografia per le asserzioni SAML
- RoleName- Nome del ruolo IAM (lasciare vuoto il campo per il nome generato automaticamente)
- RolePath- Percorso per il ruolo IAM (predefinito/)
- RolePermissionsBoundary- Politica opzionale relativa ai limiti dell'ARN delle autorizzazioni
- RoleSessionDuration- Durata massima della sessione in secondi (3600-43200, impostazione predefinita 7200)
- RoleManagedPolicy1-5- ARN opzionali con un massimo di 5 policy gestite da allegare

CloudFormation modello

Salva il seguente codice JSON o YAML come file separato da utilizzare come CloudFormation modello per questo tutorial.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "[AWSDocs] IAM: tutorial_saml-idp-and-federated-role",
  "Parameters": {
    "IdentityProviderName": {
      "Type": "String",
      "Description": "Name of the SAML Identity Provider (leave empty for auto-
generated name like '{StackName}-{UniqueId}')",
      "Default": "",
      "AllowedPattern": "^$|^[a-zA-Z0-9._-]+$",
      "ConstraintDescription": "Must be empty or contain only alphanumeric
 characters, periods, underscores, and hyphens"
    },
    "IdentityProviderSAMLMetadataDocument": {
```

```
      "Type": "String",
      "Description": "SAML metadata document from identity provider"
    },
    "IdentityProviderAddPrivateKey": {
      "Type": "String",
      "Description": "Optional private key for decrypting SAML assertions. The
private key must be a .pem file that uses AES-GCM or AES-CBC encryption algorithm
to decrypt SAML assertions.",
      "Default": ""
    },
    "IdentityProviderAssertionEncryptionMode": {
      "Type": "String",
      "Description": "Optional, sets encryption mode for SAML assertions",
      "Default": "",
      "AllowedValues": ["", "Allowed", "Required"]
    },
    "RoleName": {
      "Type": "String",
      "Description": "Name of the IAM Role (leave empty for auto-generated name like
'{StackName}-{UniqueId}')",
      "Default": "",
      "AllowedPattern": "^$|^[\\w+=,.@-]{1,64}$",
      "ConstraintDescription": "Must be empty or 1-64 characters and can contain
alphanumeric characters and +=,.@-"
    },
    "RolePath": {
      "Type": "String",
      "Description": "Path for the IAM Role",
      "AllowedPattern": "(^\\/$)|(^\\/.*\\/$)",
      "Default": "/"
    },
    "RolePermissionsBoundary": {
      "Type": "String",
      "Description": "Optional ARN of the permissions boundary policy (leave empty
for none)",
      "Default": ""
    },
    "RoleSessionDuration": {
      "Description": "The maximum session duration (in seconds) that you want to set
for the specified role (3600-43200)",
      "Type": "Number",
      "MinValue": 3600,
      "MaxValue": 43200,
      "Default": 7200
```

```
    },
    "RoleManagedPolicy1": {
      "Type": "String",
      "Description": "Optional managed policy ARN 1",
      "Default": ""
    },
    "RoleManagedPolicy2": {
      "Type": "String",
      "Description": "Optional managed policy ARN 2",
      "Default": ""
    },
    "RoleManagedPolicy3": {
      "Type": "String",
      "Description": "Optional managed policy ARN 3",
      "Default": ""
    },
    "RoleManagedPolicy4": {
      "Type": "String",
      "Description": "Optional managed policy ARN 4",
      "Default": ""
    },
    "RoleManagedPolicy5": {
      "Type": "String",
      "Description": "Optional managed policy ARN 5",
      "Default": ""
    }
  },
  "Conditions": {
    "HasCustomProviderName": {"Fn::Not": [{"Fn::Equals": [{"Ref":
"IdentityProviderName"}, ""]}]},
    "HasCustomRoleName": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleName"}, ""]}]},
    "HasPermissionsBoundary": {"Fn::Not": [{"Fn::Equals": [{"Ref":
"RolePermissionsBoundary"}, ""]}]},
    "HasPolicy1": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy1"},
""]}]},
    "HasPolicy2": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy2"},
""]}]},
    "HasPolicy3": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy3"},
""]}]},
    "HasPolicy4": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy4"},
""]}]},
    "HasPolicy5": {"Fn::Not": [{"Fn::Equals": [{"Ref": "RoleManagedPolicy5"},
""]}]},
```

```
    "HasPrivateKey": {"Fn::Not": [{"Fn::Equals": [{"Ref":
"IdentityProviderAddPrivateKey"}, ""]}]},
    "HasAssertionEncryptionMode": {"Fn::Not": [{"Fn::Equals": [{"Ref":
"IdentityProviderAssertionEncryptionMode"}, ""]}]}
  },
  "Resources": {
    "SAMLProvider": {
      "Type": "AWS::IAM::SAMLProvider",
      "Properties": {
        "Name": {"Fn::If": ["HasCustomProviderName", {"Ref":
"IdentityProviderName"}, {"Ref": "AWS::NoValue"}]},
        "SamlMetadataDocument": {"Ref": "IdentityProviderSAMLMetadataDocument"},
        "AddPrivateKey": {"Fn::If": ["HasPrivateKey", {"Ref":
"IdentityProviderAddPrivateKey"}, {"Ref": "AWS::NoValue"}]},
        "AssertionEncryptionMode": {"Fn::If": ["HasAssertionEncryptionMode", {"Ref":
"IdentityProviderAssertionEncryptionMode"}, {"Ref": "AWS::NoValue"}]}
      }
    },
    "SAMLFederatedRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": {"Fn::If": ["HasCustomRoleName", {"Ref": "RoleName"}, {"Ref":
"AWS::NoValue"}]},
        "Path": {"Ref": "RolePath"},
        "Description": "SAML federated IAM role for SSO access with specified
permissions",
        "MaxSessionDuration": {"Ref": "RoleSessionDuration"},
        "PermissionsBoundary": {"Fn::If": ["HasPermissionsBoundary", {"Ref":
"RolePermissionsBoundary"}, {"Ref": "AWS::NoValue"}]},
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Federated": {"Ref": "SAMLProvider"}
              },
              "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity",
                "sts:TagSession"
              ],
              "Condition": {
                "StringEquals": {
```

```
                "SAML:aud": "https://signin.aws.amazon.com/saml"
              }
            }
          }
        ]
      },
      "ManagedPolicyArns": {
        "Fn::Split": [
          ",",
          {
            "Fn::Join": [
              ",",
              [
                {"Fn::If": ["HasPolicy1", {"Ref": "RoleManagedPolicy1"}, {"Ref":
"AWS::NoValue"}]},
                {"Fn::If": ["HasPolicy2", {"Ref": "RoleManagedPolicy2"}, {"Ref":
"AWS::NoValue"}]},
                {"Fn::If": ["HasPolicy3", {"Ref": "RoleManagedPolicy3"}, {"Ref":
"AWS::NoValue"}]},
                {"Fn::If": ["HasPolicy4", {"Ref": "RoleManagedPolicy4"}, {"Ref":
"AWS::NoValue"}]},
                {"Fn::If": ["HasPolicy5", {"Ref": "RoleManagedPolicy5"}, {"Ref":
"AWS::NoValue"}]}
              ]
            ]
          }
        ]
      }
    }
  }
 },
 "Outputs": {
   "RoleARN": {
     "Description": "ARN of the created IAM Role",
     "Value": {"Fn::GetAtt": ["SAMLFederatedRole", "Arn"]},
     "Export": {
       "Name": {"Fn::Sub": "${AWS::StackName}-RoleARN"}
     }
   },
   "IdentityProviderARN": {
     "Description": "ARN of the created SAML Identity Provider",
     "Value": {"Ref": "SAMLProvider"},
     "Export": {
       "Name": {"Fn::Sub": "${AWS::StackName}-IdentityProviderARN"}
```

```
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Description: '[AWSDocs] IAM: tutorial_saml-idp-and-federated-role'

Parameters:
  IdentityProviderName:
    Type: String
    Description: Name of the SAML Identity Provider (leave empty for auto-generated
 name like '{StackName}-{UniqueId}')
    Default: ""
    AllowedPattern: '^$|^[a-zA-Z0-9._-]+$'
    ConstraintDescription: Must be empty or contain only alphanumeric characters,
 periods, underscores, and hyphens

  IdentityProviderSAMLMetadataDocument:
    Type: String
    Description: SAML metadata document from identity provider

  IdentityProviderAddPrivateKey:
    Type: String
    Description: Optional private key for decrypting SAML assertions. The private
 key must be a .pem file that uses AES-GCM or AES-CBC encryption algorithm to
 decrypt SAML assertions.
    Default: ""

  IdentityProviderAssertionEncryptionMode:
    Type: String
    Description: Optional, sets encryption mode for SAML assertions
    Default: ""
    AllowedValues:
      - ""
      - "Allowed"
      - "Required"

  RoleName:
    Type: String
```

```
    Description: Name of the IAM Role (leave empty for auto-generated name like
'{StackName}-{UniqueId}')
    Default: ""
    AllowedPattern: '^$|^[\w+=,.@-]{1,64}$'
    ConstraintDescription: "Must be empty or 1-64 characters and can contain
alphanumeric characters and +=,.@-"

  RolePath:
    Type: String
    Description: Path for the IAM Role
    AllowedPattern: (^\/$)|(^\/.*\/$)
    Default: "/"

  RolePermissionsBoundary:
    Type: String
    Description: Optional ARN of the permissions boundary policy (leave empty for
none)
    Default: ""

  RoleSessionDuration:
    Description: The maximum session duration (in seconds) that you want to set for
the specified role (3600-43200)
    Type: Number
    MinValue: 3600
    MaxValue: 43200
    Default: 7200

  RoleManagedPolicy1:
    Type: String
    Description: Optional managed policy ARN 1
    Default: ""
  RoleManagedPolicy2:
    Type: String
    Description: Optional managed policy ARN 2
    Default: ""
  RoleManagedPolicy3:
    Type: String
    Description: Optional managed policy ARN 3
    Default: ""
  RoleManagedPolicy4:
    Type: String
    Description: Optional managed policy ARN 4
    Default: ""
  RoleManagedPolicy5:
```

```
    Type: String
    Description: Optional managed policy ARN 5
    Default: ""

Conditions:
  HasCustomProviderName: !Not [!Equals [!Ref IdentityProviderName, ""]]
  HasCustomRoleName: !Not [!Equals [!Ref RoleName, ""]]
  HasPermissionsBoundary: !Not [!Equals [!Ref RolePermissionsBoundary, ""]]
  HasPolicy1: !Not [!Equals [!Ref RoleManagedPolicy1, ""]]
  HasPolicy2: !Not [!Equals [!Ref RoleManagedPolicy2, ""]]
  HasPolicy3: !Not [!Equals [!Ref RoleManagedPolicy3, ""]]
  HasPolicy4: !Not [!Equals [!Ref RoleManagedPolicy4, ""]]
  HasPolicy5: !Not [!Equals [!Ref RoleManagedPolicy5, ""]]
  HasPrivateKey: !Not [!Equals [!Ref IdentityProviderAddPrivateKey, ""]]
  HasAssertionEncryptionMode: !Not [!Equals [!Ref
 IdentityProviderAssertionEncryptionMode, ""]]

Resources:
  SAMLProvider:
    Type: AWS::IAM::SAMLProvider
    Properties:
      Name: !If
        - HasCustomProviderName
        - !Ref IdentityProviderName
        - !Ref AWS::NoValue
      SamlMetadataDocument: !Ref IdentityProviderSAMLMetadataDocument
      AddPrivateKey: !If
        - HasPrivateKey
        - !Ref IdentityProviderAddPrivateKey
        - !Ref AWS::NoValue
      AssertionEncryptionMode: !If
        - HasAssertionEncryptionMode
        - !Ref IdentityProviderAssertionEncryptionMode
        - !Ref AWS::NoValue

  SAMLFederatedRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: !If
        - HasCustomRoleName
        - !Ref RoleName
        - !Ref AWS::NoValue
      Path: !Ref RolePath
```

```
      Description: "SAML federated IAM role for SSO access with specified
 permissions"
      MaxSessionDuration: !Ref RoleSessionDuration
      PermissionsBoundary: !If
        - HasPermissionsBoundary
        - !Ref RolePermissionsBoundary
        - !Ref AWS::NoValue
      AssumeRolePolicyDocument:
        Version: '2012-10-17       '
        Statement:
          - Effect: Allow
            Principal:
              Federated: !Ref SAMLProvider
            Action:
              - 'sts:AssumeRole'
              - 'sts:SetSourceIdentity'
              - 'sts:TagSession'
            Condition:
              StringEquals:
                'SAML:aud': 'https://signin.aws.amazon.com/saml'
      ManagedPolicyArns: !Split
        - ','
        - !Join
          - ','
          - - !If [HasPolicy1, !Ref RoleManagedPolicy1, !Ref "AWS::NoValue"]
            - !If [HasPolicy2, !Ref RoleManagedPolicy2, !Ref "AWS::NoValue"]
            - !If [HasPolicy3, !Ref RoleManagedPolicy3, !Ref "AWS::NoValue"]
            - !If [HasPolicy4, !Ref RoleManagedPolicy4, !Ref "AWS::NoValue"]
            - !If [HasPolicy5, !Ref RoleManagedPolicy5, !Ref "AWS::NoValue"]

Outputs:
  RoleARN:
    Description: ARN of the created IAM Role
    Value: !GetAtt SAMLFederatedRole.Arn
    Export:
      Name: !Sub '${AWS::StackName}-RoleARN'

  IdentityProviderARN:
    Description: ARN of the created SAML Identity Provider
    Value: !Ref SAMLProvider
    Export:
      Name: !Sub '${AWS::StackName}-IdentityProviderARN'
```

Identità IAM

Un'identità IAM può essere associata a una o più policy, che determinano quali azioni un'identità è autorizzata a eseguire, su quali AWS risorse e in quali condizioni. Le identità IAM includono utenti IAM, ruoli IAM e gruppi IAM. Un'entità IAM è un tipo di identità che rappresenta un utente umano o un carico di lavoro programmatico che può essere autenticato e quindi autorizzato a eseguire operazioni in Account AWS. Le entità IAM includono utenti IAM e ruoli IAM. Per le definizioni dei termini di uso comune, consulta Termini.

Puoi federare identità esistenti da un provider di identità esterno. Queste identità assumeranno ruoli IAM per accedere alle AWS risorse. Per ulteriori informazioni, consulta the section called "Fornitori di identità e federazione in AWS".

Puoi anche utilizzarle Centro identità AWS IAM per creare e gestire identità e accesso alle AWS risorse. I set di autorizzazioni del Centro identità IAM creano automaticamente i ruoli IAM necessari per fornire l'accesso alle risorse. Per ulteriori informazioni, consulta Cos'è il Centro identità IAM?.

Puoi utilizzare l'account access manager , una funzionalità IAM che consente di assegnare ruoli IAM esistenti con il set completo di funzionalità dei ruoli IAM agli utenti e ai gruppi di IAM Identity Center. Puoi usarlo insieme ai set di autorizzazioni o da solo.

Utente root dell'account AWS È un Account AWS principio che viene creato quando Account AWS viene stabilito il tuo. L'utente root ha accesso a tutti i AWS servizi e le risorse dell'account. Per ulteriori informazioni, consulta the section called "Utente root IAM".

> **Note**
>
> - Segui le best practice di sicurezza in IAM quando lavori con le identità IAM.
> - Segui le best practice per l'utente root per il tuo Account AWS quando lavori con l'utente root.
> - Se riscontri problemi di accesso, consulta Accedi alla Console di gestione AWS.

Utente root IAM

Quando ne crei per la prima volta Account AWS, inizi con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root. Per ulteriori informazioni, consulta la sezione Utente root dell'account AWS.

Utenti IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Per ulteriori informazioni, consulta Utenti IAM.

Gruppi di utenti IAM

Un gruppo di utenti IAM è un'identità che specifica un insieme di utenti IAM. Per ulteriori informazioni, consulta Gruppi di utenti.

Ruoli IAM

Un ruolo IAM è un'identità all'interno dell'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per ulteriori informazioni, consulta Ruoli IAM.

Utente root dell'account AWS

Quando crei per la prima volta un account Amazon Web Services (AWS), inizi con un'identità di accesso singolo che ha accesso completo a tutti i AWS servizi e le risorse dell'account. Questa identità è chiamata utente root dell' AWS account. L'indirizzo e-mail e la password che hai usato per creare il tuo Account AWS sono le credenziali che usi per accedere come utente root.

- Ricorri all'utente root solo per eseguire le attività che richiedono le autorizzazioni a livello root. Per un elenco completo delle attività che richiedono il tuo accesso come utente root, consulta la pagina Attività che richiedono credenziali dell'utente root.
- Segui le best practice per gli utenti root per il tuo Account AWS.
- Se riscontri problemi di accesso, consulta Accedi alla Console di gestione AWS.

> **Important**
>
> Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane e di seguire le best practice dell'utente root per il tuo Account AWS. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono il tuo accesso come utente root, consulta la pagina Attività che richiedono credenziali dell'utente root.

Sebbene l'MFA venga applicata per impostazione predefinita per gli utenti root, è comunque necessaria un'azione da parte del cliente per aggiungere l'MFA durante la creazione iniziale dell'account o quando viene richiesto durante l'accesso. Per ulteriori informazioni sull'utilizzo dell'MFA per proteggere l'utente root, consulta Multi-factor autenticazione per Utente root dell'account AWS.

Oltre all'accesso alla console, puoi gestire l'accesso programmatico per l'utente root. Puoi farlo Creare chiavi di accesso per l'utente root e Eliminare le chiavi di accesso per l'utente root quando non sono più necessari.

Gestire centralmente l'accesso root per gli account membri

Per aiutarti a gestire le credenziali su larga scala, puoi proteggere centralmente l'accesso alle credenziali degli utenti root per gli account membri in AWS Organizations. Quando abiliti AWS Organizations, unisci tutti i tuoi AWS account in un'organizzazione per la gestione centralizzata. La centralizzazione dell'accesso root consente di rimuovere le credenziali dell'utente root e completare le seguenti attività con privilegi sugli account membri.

Rimuovere le credenziali dell'utente root per gli account membri

Dopo aver centralizzato l'accesso root per gli account membri, puoi scegliere di eliminare le credenziali dell'utente root dagli account membri in Organizations. Puoi rimuovere la password dell'utente root, le chiavi di accesso, i certificati per la firma e disattivare l'autenticazione a più fattori (MFA). Per impostazione predefinita, i nuovi account creati in Organizations non hanno credenziali dell'utente root. Gli account membri non possono accedere al proprio utente root o eseguirne il recupero della password a meno che non sia abilitato il recupero dell'account.

Esegui attività con privilegi che richiedono credenziali dell'utente root.

Alcune attività possono essere eseguite solo quando si effettua l'accesso come utente root di un account. Alcune di queste Attività che richiedono credenziali dell'utente root possono essere

eseguite dall'account di gestione o dall'amministratore delegato di IAM. Per ulteriori informazioni su come eseguire azioni con privilegi sugli account membri, consulta Eseguire un'attività con privilegi.

Abilitare il ripristino dell'account dell'utente root

Se è necessario recuperare le credenziali dell'utente root per un account membro, l'account di gestione di Organizations o l'amministratore delegato può eseguire l'attività con privilegi Consenti il recupero della password. La persona con accesso alla casella di posta elettronica dell'utente root per l'account membro può reimpostare la password dell'utente root per recuperare le credenziali dell'utente root. Si consiglia di eliminare le credenziali dell'utente root dopo aver completato l'operazione che richiede l'accesso all'utente root.

Centralizzare l'accesso root per gli account dei membri

Le credenziali utente root sono le credenziali iniziali assegnate a ciascun utente Account AWS che ha accesso completo a tutti i AWS servizi e le risorse dell'account. Quando abiliti AWS Organizations, unisci tutti i tuoi AWS account in un'organizzazione per la gestione centralizzata. Ogni account membro ha il proprio utente root con autorizzazioni predefinite per eseguire qualsiasi azione nell'account membro. Ti consigliamo di proteggere centralmente le credenziali dell'utente root di Account AWS managed using AWS Organizations per impedirne il ripristino e l'accesso su larga scala.

Dopo aver centralizzato l'accesso root, puoi scegliere di eliminare le credenziali dell'utente root dagli account membri dell'organizzazione. Puoi rimuovere la password dell'utente root, le chiavi di accesso, i certificati per la firma e disattivare l'autenticazione a più fattori (MFA). Per impostazione predefinita, i nuovi account creati in AWS Organizations non hanno credenziali dell'utente root. Gli account dei membri non possono accedere al proprio utente root o eseguirne il recupero della password.

> **Note**
>
> Sebbene alcune Attività che richiedono credenziali dell'utente root possano essere eseguite dall'account di gestione o dall'amministratore delegato per IAM, alcune attività possono essere eseguite solo quando si effettua l'accesso come utente root di un account.
> Se è necessario recuperare le credenziali dell'utente root per un account membro al fine di eseguire una di queste attività, segui la procedura indicata in Eseguire un'attività con privilegi e seleziona Consenti il recupero della password. La persona che ha accesso alla casella

di posta elettronica dell'utente root per l'account membro può quindi seguire i passaggi per reimpostare la password dell'utente root e accedere come utente root dell'account membro. Ti consigliamo di eliminare le credenziali dell'utente root una volta completata l'attività che richiede l'accesso all'utente root.

Prerequisiti

Prima di centralizzare l'accesso root, è necessario disporre di un account configurato con le seguenti impostazioni:

- Bisogna possedere le seguenti autorizzazioni IAM:
  - `iam:GetAccessKeyLastUsed`
  - `iam:GetAccountSummary`
  - `iam:GetLoginProfile`
  - `iam:GetUser`
  - `iam:ListAccessKeys`
  - `iam:ListMFADevices`
  - `iam:ListSigningCertificates`
  - `sts:AssumeRoot`

> **Note**
>
> Per verificare lo stato delle credenziali di utente root di un account membro, puoi utilizzare la policy IAMAuditRootUserCredentials AWS gestita per limitare le autorizzazioni quando esegui un'attività privilegiata su un account AWS Organizations membro o utilizzare qualsiasi policy con accesso a. `iam:GetAccountSummary`
> Per generare il report informativo sulle credenziali dell'utente root, per le altre policy è sufficiente che l'operazione `iam:GetAccountSummary` produca lo stesso risultato. Inoltre, è possibile elencare oppure ottenere informazioni sulle credenziali dei singoli utenti root, compreso:
>
> - Se è presente una password per l'utente root
> - Se è presente una chiave di accesso dell'utente root e quando è stata utilizzata l'ultima volta
> - Se l'utente root ha certificati di firma associati

- I dispositivi MFA associati all'utente root
- L'elenco dello stato delle credenziali consolidate dell'utente root

- È necessario gestire i propri Account AWS dati di accesso. AWS Organizations
- Per abilitare questa funzionalità nella tua organizzazione, devi disporre delle seguenti autorizzazioni:
  - `iam:EnableOrganizationsRootCredentialsManagement`
  - `iam:EnableOrganizationsRootSessions`
  - `iam:ListOrganizationsFeatures`
  - `organizations:EnableAwsServiceAccess`
  - `organizations:ListAccountsForParent`
  - `organizations:RegisterDelegatedAdministrator`
- Per garantire una funzionalità ottimale della console, si consiglia di abilitare le seguenti autorizzazioni aggiuntive:
  - `organizations:DescribeAccount`
  - `organizations:DescribeOrganization`
  - `organizations:ListAWSServiceAccessForOrganization`
  - `organizations:ListDelegatedAdministrators`
  - `organizations:ListOrganizationalUnitsForParent`
  - `organizations:ListParents`
  - `organizations:ListTagsForResource`

Abilitazione dell'accesso root centralizzato (console)

Per abilitare questa funzionalità per gli account dei membri nel Console di gestione AWS

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegli Gestione degli accessi root, quindi seleziona Abilita.

> **Note**
>
> Se vedi che la gestione degli accessi root è disabilitata, abilita l'accesso affidabile per AWS Identity and Access Management l'accesso AWS Organizations. Per i dettagli, consulta AWS IAM e AWS Organizations nella Guida per l'utente di AWS Organizations .

3. Nella sezione Funzionalità da abilitare, scegli le funzionalità da abilitare.

   - Seleziona Gestione delle credenziali root per consentire all'account di gestione e all'amministratore delegato di IAM di eliminare le credenziali utente root per gli account membri. È necessario abilitare le azioni root con privilegi negli account membri per consentire agli account membri di recuperare le credenziali dell'utente root dopo che sono state eliminate.
   - Seleziona Azioni root con privilegi negli account membri per consentire all'account di gestione e all'amministratore delegato di IAM di eseguire determinate attività che richiedono le credenziali dell'utente root.

4. (Facoltativo) Inserisci l'ID account dell'amministratore delegato autorizzato a gestire l'accesso degli utenti root e ad eseguire azioni con privilegi sugli account membri. Consigliamo un account destinato a scopi di sicurezza o gestione.
5. Scegli Abilita .

Abilitazione dell'accesso root centralizzato (AWS CLI)

Per abilitare l'accesso root centralizzato da AWS Command Line Interface (AWS CLI)

1. Se non hai già abilitato l'accesso affidabile per AWS Identity and Access Management in AWS Organizations, usa il seguente comando: aws organisations enable-aws-service-access.
2. Seleziona Gestione delle credenziali root per consentire all'account di gestione e all'amministratore delegato di IAM di eliminare le credenziali utente root per gli account membri.
3. Utilizza il seguente comando per consentire all'account di gestione e all'amministratore delegato di eseguire determinate attività che richiedono le credenziali dell'utente root: aws iam enable-organizations-root-sessions.
4. (Facoltativo) Usa il seguente comando per registrare un amministratore delegato: aws organisations register-delegated-administrator.

   L'esempio seguente assegna l'account 111111111111 come amministratore delegato per il servizio IAM.

```
aws organizations register-delegated-administrator
--service-principal iam.amazonaws.com
--account-id 111111111111
```

Abilitazione dell'accesso root centralizzato (AWS API)

Per abilitare l'accesso root centralizzato da AWS "Hello, World!"

1. Se non hai già abilitato l'accesso affidabile per AWS Identity and Access Management in AWS Organizations, usa il seguente comando: EnableAWSServiceAccess.
2. Usa il seguente comando per consentire all'account di gestione e all'amministratore delegato di eliminare le credenziali dell'utente root per gli account membri:. EnableOrganizationsRootCredentialsManagement
3. Utilizza il seguente comando per consentire all'account di gestione e all'amministratore delegato di eseguire determinate attività che richiedono le credenziali dell'utente root:. EnableOrganizationsRootSessions
4. (Facoltativo) Usa il seguente comando per registrare un amministratore delegato:. RegisterDelegatedAdministrator

Fasi successive

Dopo aver protetto centralmente le credenziali con privilegi per gli account membri dell'organizzazione, consulta Eseguire un'attività con privilegi per eseguire le azioni con privilegi su un account membro.

Esegui un'attività privilegiata su un AWS Organizations account membro

L'account di AWS Organizations gestione o un account amministratore delegato per IAM può eseguire alcune attività privilegiate sugli account membri che altrimenti richiederebbero le credenziali dell'utente root. Con l'accesso root centralizzato, queste attività vengono eseguite tramite sessioni privilegiate a breve termine. Queste sessioni forniscono credenziali temporanee destinate a specifiche azioni privilegiate, senza richiedere l'accesso dell'utente root sull'account membro.

Una volta avviata una sessione con privilegi, puoi eliminare una policy del bucket Amazon S3 configurata in modo errato, eliminare la policy di una coda Amazon SQS non configurata

correttamente, eliminare le credenziali dell'utente root per un account membro e riabilitare le credenziali dell'utente root per un account membro.

> **Note**
>
> Per utilizzare l'accesso root centralizzato, devi accedere tramite un account di gestione o un account di amministratore delegato come utente o ruolo IAM con l'autorizzazione esplicitamente concessa. `sts:AssumeRoot` Non è possibile utilizzare le credenziali dell'utente root per chiamare. `sts:AssumeRoot`

Prerequisiti

Prima di poter avviare una sessione con privilegi, è necessario disporre delle seguenti impostazioni:

- Devi aver abilitato l'accesso root centralizzato nella tua organizzazione. Per le operazioni per abilitare questa funzionalità, consulta Centralizzare l'accesso root per gli account dei membri.
- Il tuo account di gestione o l'account amministratore delegato deve disporre delle seguenti autorizzazioni: `sts:AssumeRoot`

Esecuzione di un'azione con privilegi su un account membro (console)

Per avviare una sessione per un'azione privilegiata in un account membro nel Console di gestione AWS

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegli Gestione dell'accesso root.
3. Seleziona un nome dall'elenco degli account membri e scegli Esegui azioni con privilegi.
4. Scegli l'azione con privilegi che desideri eseguire nell'account membro.
   - Seleziona Elimina policy del bucket Amazon S3 per rimuovere una policy del bucket mal configurata che impedisce a tutti i principali di accedere al bucket Amazon S3.
     a. Scegli Sfoglia S3 per selezionare un nome dai bucket di proprietà dell'account membro e seleziona Scegli.
     b. Scegli Elimina policy del bucket.

c. Utilizza la console Amazon S3 per correggere la policy del bucket dopo aver eliminato la policy configurata non correttamente. Per informazioni, consulta Aggiunta di una policy del bucket utilizzando la console Amazon S3 nella Guida per l'utente di Amazon S3.

- Seleziona Elimina la policy Amazon SQS per eliminare una policy basata sulle risorse Amazon Simple Queue Service che rifiuta a tutti i principali l'accesso a una coda Amazon SQS.

  a. Inserisci il nome della coda in Nome della coda SQS e seleziona Elimina la policy SQS.

  b. Utilizza la console Amazon SQS per correggere la policy della coda dopo aver eliminato la policy configurata non correttamente. Per ulteriori informazioni, consulta Configurazione di una policy di accesso in Amazon SQS nella Guida per gli sviluppatori di Amazon SQS.

- Seleziona Elimina le credenziali root per rimuovere l'accesso root da un account membro. L'eliminazione delle credenziali dell'utente root rimuove la password dell'utente root, le chiavi di accesso e i certificati di firma, e disattiva l'autenticazione a più fattori (MFA) per l'account membro.

  - Scegli Elimina le credenziali root.

- Seleziona Consenti il recupero della password per recuperare le credenziali dell'utente root per un account membro.

  Questa opzione è disponibile solo quando l'account membro non ha le credenziali dell'utente root.

  a. Scegli Consenti il recupero della password.

  b. Dopo aver eseguito questa azione con privilegi, la persona con accesso alla casella di posta elettronica dell'utente root per l'account membro può reimpostare la password dell'utente root e accedere all'utente root dell'account membro.

Effettuare un'azione privilegiata su un account membro (AWS CLI)

Per avviare una sessione di azione privilegiata in un account membro dal AWS Command Line Interface

1. Usa il seguente comando per assumere una sessione dell'utente root: aws sts assume-root.

> **Note**
>
> L'endpoint globale non è supportato per `sts:AssumeRoot`. È necessario inviare questa richiesta a un AWS STS endpoint regionale. Per ulteriori informazioni, consulta Manage (Gestione) AWS STS in un Regione AWS.

Quando si avvia una sessione dell'utente root con privilegi per un account membro, è necessario definire l'`task-policy-arn` per limitare l'ambito della sessione all'azione con privilegi da eseguire durante la sessione. È possibile utilizzare una delle seguenti policy gestite da AWS per definire l'ambito delle azioni di sessione con privilegi.

- IAMAuditRootUserCredentials
- IAMCreateRootUserPassword
- IAMDeleteRootUserCredentials
- S3UnlockBucketPolicy
- SQSUnlockQueuePolicy

Per limitare le azioni che un account di gestione o un amministratore delegato può eseguire durante una sessione utente root privilegiata, è possibile utilizzare la chiave AWS STS condizionale sts:. TaskPolicyArn

Nel seguente esempio, l'amministratore delegato presuppone che root elimini le credenziali dell'utente root per l'ID dell'account membro. *111122223333*

```
aws sts assume-root \
  --target-principal 111122223333 \
  --task-policy-arn arn=arn:aws:iam::aws:policy/root-task/
IAMDeleteRootUserCredentials \
  --duration-seconds 900
```

2. Utilizza `SessionToken`, `AccessKeyId` e `SecretAccessKey` dalla risposta per eseguire operazioni con privilegi nell'account membro. È possibile omettere il nome utente e la password nella richiesta per impostare come impostazione predefinita l'account utente.
   - Controlla lo stato delle credenziali dell'utente root. Usa i seguenti comandi per controllare lo stato delle credenziali dell'utente root per un account membro.

  - get-user
  - get-login-profile
  - list-access-keys
  - list-signing-certificates
  - list-mfa-devices
  - get-access-key-last-used
- Elimina le credenziali dell'utente root. Per eliminare l'accesso root utilizza i comandi riportati di seguito. È possibile rimuovere la password dell'utente root, le chiavi di accesso, i certificati di firma e disattivare l'autenticazione a più fattori (MFA) per rimuovere l'accesso e il recupero dell'utente root.
  - delete-login-profile
  - delete-access-key
  - delete-signing-certificate
  - deactivate-mfa-device
- Elimina la policy del bucket Amazon S3. Utilizza i seguenti comandi per leggere, modificare ed eliminare una policy del bucket configurata in modo errato che impedisce a tutti i principali di accedere al bucket Amazon S3.
  - list-buckets
  - get-bucket-policy
  - put-bucket-policy
  - delete-bucket-policy
- Elimina la policy di Amazon SQS. Utilizza i seguenti comandi per visualizzare ed eliminare una policy basata sulle risorse Amazon Simple Queue Service che nega a tutti i principali l'accesso a una coda Amazon SQS.
  - list-queues
  - get-queue-url
  - get-queue-attributes
  - set-queue-attributes
- Consenti il recupero della password. Usa i seguenti comandi per visualizzare il nome utente e recuperare le credenziali dell'utente root per un account membro.

- get-login-profile
- create-login-profile

Esecuzione di un'azione privilegiata su un account membro (AWS API)

Per avviare una sessione per un'azione privilegiata in un account membro dal AWS "Hello, World!"

1. Usa il seguente comando per assumere una sessione utente root: AssumeRoot.

> **Note**
>
> L'endpoint globale non è supportato per AssumeRoot. È necessario inviare questa richiesta a un AWS STS endpoint regionale. Per ulteriori informazioni, consulta Manage (Gestione) AWS STS in un Regione AWS.

Quando si avvia una sessione dell'utente root con privilegi per un account membro, è necessario definire l'`TaskPolicyArn` per limitare l'ambito della sessione all'azione con privilegi da eseguire durante la sessione. È possibile utilizzare una delle seguenti politiche AWS gestite per definire l'ambito delle azioni di sessione privilegiate.

- IAMAuditRootUserCredentials
- IAMCreateRootUserPassword
- IAMDeleteRootUserCredentials
- S3UnlockBucketPolicy
- SQSUnlockQueuePolicy

Per limitare le azioni che un account di gestione o un amministratore delegato può eseguire durante una sessione utente root privilegiata, è possibile utilizzare la chiave AWS STS condizionale sts: TaskPolicyArn.

Nel seguente esempio, l'amministratore delegato presume che root legga, modifichi ed elimini una policy basata su risorse non configurata correttamente per un bucket Amazon S3 per l'ID dell'account membro. *111122223333*

```
https://sts.us-east-2.amazonaws.com/
```

```
?Version=2011-06-15
&Action=AssumeRoot
&TargetPrincipal=111122223333
&PolicyArns.arn=arn:aws:iam::aws:policy/root-task/S3UnlockBucketPolicy
&DurationSeconds=900
```

2. Utilizza `SessionToken`, `AccessKeyId` e `SecretAccessKey` dalla risposta per eseguire operazioni con privilegi nell'account membro. È possibile omettere il nome utente e la password nella richiesta per impostare come impostazione predefinita l'account utente.

   - Controlla lo stato delle credenziali dell'utente root. Usa i seguenti comandi per controllare lo stato delle credenziali dell'utente root per un account membro.

     - GetUser
     - GetLoginProfile
     - ListAccessKeys
     - ListSigningCertificates
     - ListMFADevices
     - GetAccessKeyLastUsed

   - Elimina le credenziali dell'utente root. Per eliminare l'accesso root utilizza i comandi riportati di seguito. È possibile rimuovere la password dell'utente root, le chiavi di accesso, i certificati di firma e disattivare l'autenticazione a più fattori (MFA) per rimuovere l'accesso e il recupero dell'utente root.

     - DeleteLoginProfile
     - DeleteAccessKey
     - DeleteSigningCertificate
     - DeactivateMfaDevice

   - Elimina la policy del bucket Amazon S3. Utilizza i seguenti comandi per leggere, modificare ed eliminare una policy del bucket configurata in modo errato che impedisce a tutti i principali di accedere al bucket Amazon S3.

     - ListBuckets
     - GetBucketPolicy
     - PutBucketPolicy
     - DeleteBucketPolicy

- Elimina la policy di Amazon SQS. Utilizza i seguenti comandi per visualizzare ed eliminare una policy basata sulle risorse Amazon Simple Queue Service che nega a tutti i principali l'accesso a una coda Amazon SQS.
  - ListQueues
  - GetQueueUrl
  - GetQueueAttributes
  - SetQueueAttributes
- Consenti il recupero della password. Usa i seguenti comandi per visualizzare il nome utente e recuperare le credenziali dell'utente root per un account membro.
  - GetLoginProfile
  - CreateLoginProfile

Multi-factor autenticazione per Utente root dell'account AWS

> **Important**
>
> AWS consiglia di utilizzare una passkey o una chiave di sicurezza per l'MFA AWS, laddove possibile, poiché sono più resistenti agli attacchi come il phishing. Per ulteriori informazioni, consulta Passkey e chiavi di sicurezza.

Multi-factor l'autenticazione (MFA) è un meccanismo semplice ed efficace per migliorare la sicurezza. Il primo fattore, la password, è un segreto che viene memorizzato, noto anche come fattore di conoscenza. Altri fattori possono essere fattori di possesso (qualcosa che possiedi, come una chiave di sicurezza) o fattori intrinseci (qualcosa che sei, come una scansione biometrica). Per una maggiore sicurezza, consigliamo vivamente di configurare l'autenticazione a più fattori (MFA) per proteggere le risorse. AWS

> **Note**
>
> Tutti i Account AWS tipi (account autonomi, di gestione e membri) richiedono la configurazione dell'MFA per l'utente root. Gli utenti devono registrare l'MFA entro 35 giorni dal primo tentativo di accesso per accedere a Console di gestione AWS se l'MFA non è già abilitato.

Puoi abilitare l'MFA per gli utenti IAM e per gli Utente root dell'account AWS utenti IAM. Quando abiliti MFA per l'utente root, questa impostazione influisce solo sulle credenziali dell'utente root. Per ulteriori informazioni su come abilitare l'MFA per gli utenti IAM, consulta AWS Multi-factor autenticazione in IAM.

> **Note**
>
> Account AWS managed using AWS Organizations potrebbe avere la possibilità di gestire centralmente l'accesso root per gli account dei membri per impedire il ripristino delle credenziali e l'accesso su larga scala. Se questa opzione è abilitata, è possibile eliminare le credenziali dell'utente root dagli account membri, incluse password e MFA, impedendo efficacemente l'accesso come utente root, il recupero della password o la configurazione dell'MFA. In alternativa, se preferisci mantenere i metodi di accesso basati su password, proteggi il tuo account registrando l'MFA per una maggiore protezione.

Prima di abilitare MFA per il tuo utente root, rivedi e aggiorna le impostazioni dell'account e le informazioni di contatto per verificare di disporre dell'accesso a e-mail e numero di telefono. Se il dispositivo MFA viene smarrito, rubato o non funziona, è comunque possibile accedere come utente root verificando la propria identità utilizzando tale e-mail e il numero di telefono. Per ulteriori informazioni sull'accesso utilizzando fattori di autenticazione alternativi, consultare Recuperare un'identità protetta da MFA in IAM. Per disabilitare questa funzionalità, contatta Supporto AWS.

AWS supporta i seguenti tipi di MFA per l'utente root:

- Passkey e chiavi di sicurezza
- Applicazioni di autenticazione virtuale
- Token TOTP hardware

Passkey e chiavi di sicurezza

AWS Identity and Access Management supporta le chiavi di accesso e le chiavi di sicurezza per l'MFA. In base agli standard FIDO, le password utilizzano la crittografia a chiave pubblica per fornire un'autenticazione forte e resistente al phishing e più sicura delle password. AWS supporta due tipi di password: password associate al dispositivo (chiavi di sicurezza) e chiavi di accesso sincronizzate.

- Chiavi di sicurezza: si tratta di dispositivi fisici, ad esempio YubiKey, utilizzati come secondo fattore di autenticazione. Una singola chiave di sicurezza può supportare più account utente root e utenti IAM.
- Passkey sincronizzate: come secondo fattore utilizzano gestori di credenziali di provider come Google, Apple, account Microsoft e servizi di terze parti come 1Password, Dashlane e Bitwarden come secondo fattore.

Puoi utilizzare gli autenticatori biometrici integrati, come Touch ID su Apple MacBooks, per sbloccare il gestore delle credenziali e accedere a. AWS Le passkey vengono create con il provider scelto utilizzando l'impronta digitale, il viso o il PIN del dispositivo. Puoi anche utilizzare una passkey di autenticazione tra dispositivi (CDA) da un dispositivo, ad esempio un dispositivo mobile o una chiave di sicurezza hardware, per accedere su un altro dispositivo, ad esempio un laptop. Per ulteriori informazioni, consulta Cross-device authentication (CDA).

Puoi sincronizzare le password tra i tuoi dispositivi per facilitare l'accesso, migliorando l'usabilità e la recuperabilità. AWS Per ulteriori informazioni sull'abilitazione delle passkey e delle chiavi di sicurezza, consulta Abilitare una passkey o una chiave di sicurezza per l'utente root (console).

FIDO Alliance mantiene un elenco di tutti i prodotti certificati FIDO compatibili con le specifiche FIDO.

Applicazioni di autenticazione virtuale

Un'applicazione di autenticazione virtuale che viene eseguita su un telefono o altro dispositivo e simula un dispositivo fisico. Le app di autenticazione virtuale implementano l'algoritmo TOTP (password monouso) e supportano più token su un singolo dispositivo. L'utente deve immettere un codice valido dal dispositivo quando richiesto durante la procedura di accesso. Ogni token assegnato a un utente deve essere univoco. Per autenticarsi, un utente non può digitare un codice dal token di un altro utente.

È consigliabile utilizzare un dispositivo MFA virtuale nell'attesa dell'approvazione di un acquisto hardware o della consegna del dispositivo hardware. Per un elenco di alcune app supportate che puoi utilizzare come dispositivi MFA virtuali, vedi Autenticazione (MFA). Multi-Factor Per istruzioni sulla configurazione di un dispositivo MFA virtuale con AWS, consulta. Abilitare un dispositivo MFA virtuale per l'utente root (console)

Token TOTP hardware

Un dispositivo hardware che genera un codice numerico a sei cifre basato sull'algoritmo con password monouso. L'utente deve immettere un codice valido dal dispositivo su una seconda pagina

Web durante la procedura di accesso. Ogni dispositivo MFA assegnato a un utente deve essere univoco. Per essere autenticati, gli utenti non possono digitare un codice generato dal dispositivo di un altro utente. Per informazioni sui dispositivi hardware MFA supportati, vedere Multi-Factor Autenticazione (MFA). Per le istruzioni sulla configurazione di un token TOTP hardware con AWS, consulta Abilitazione di un token TOTP hardware per l'utente root dell' (console).

Se desideri utilizzare un dispositivo MFA fisico, ti consigliamo di utilizzare le chiavi di sicurezza FIDO come alternativa ai dispositivi TOTP hardware. Le chiavi di sicurezza FIDO offrono i vantaggi di non richiedere alcuna batteria, resistono al phishing e supportano più utenti root e IAM su un unico dispositivo per una maggiore sicurezza.

Argomenti

- Abilitare una passkey o una chiave di sicurezza per l'utente root (console)
- Abilitare un dispositivo MFA virtuale per l'utente root (console)
- Abilitazione di un token TOTP hardware per l'utente root dell' (console)

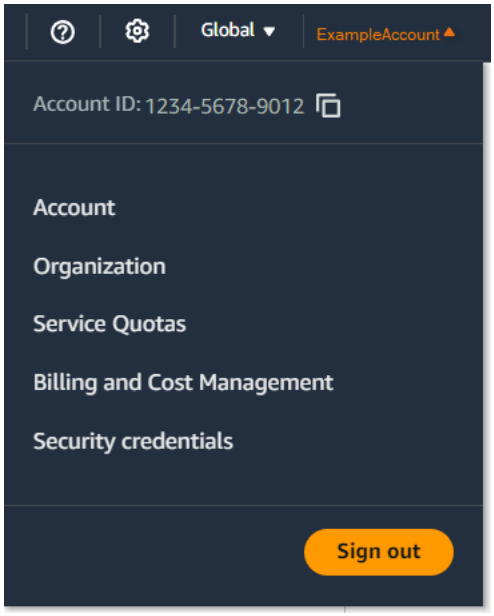
Abilitare una passkey o una chiave di sicurezza per l'utente root (console)

Puoi configurare e abilitare una passkey per il tuo utente root Console di gestione AWS solo dall'API, non dall' AWS API AWS CLI o.

Per abilitare una passkey o una chiave di sicurezza per l'utente root (console)

1. Apri la Console di gestione AWS e accedi utilizzando le credenziali dell'utente root.

   Per istruzioni, consulta Accedere Console di gestione AWS come utente root nella Guida per l'Accedi ad AWS utente.
2. Scegli il nome dell'account sul lato destro della barra di navigazione, quindi seleziona Credenziali di sicurezza.



3. Nella pagina Le mie credenziali di sicurezza dell'utente root, in Multi-factor Autenticazione (MFA), scegli Assegna dispositivo MFA.
4. Nella pagina Nome del dispositivo MFA, inserisci un nome per il dispositivo, scegli Passkey o chiave di sicurezza, quindi scegli Avanti.
5. In Configura dispositivo, configura la tua passkey. Crea una passkey con dati biometrici come il viso o l'impronta digitale, con un pin del dispositivo oppure inserendo la chiave di sicurezza FIDO nella porta USB del computer e toccandola.
6. Segui le istruzioni del tuo browser per scegliere un provider di passkey o selezionare dove vuoi archiviare la passkey da utilizzare su tutti i tuoi dispositivi.
7. Scegli Continua.

Ora hai registrato la tua passkey per utilizzarla con. AWS La prossima volta che utilizzi le credenziali dell'utente root per effettuare l'accesso, dovrai autenticarti con la passkey per completare la procedura di accesso.

Per assistenza con la risoluzione dei problemi della chiave di sicurezza FIDO in , consulta Risoluzione dei problemi relativi alle passkey e alle chiavi di sicurezza FIDO.

Abilitare un dispositivo MFA virtuale per l'utente root (console)

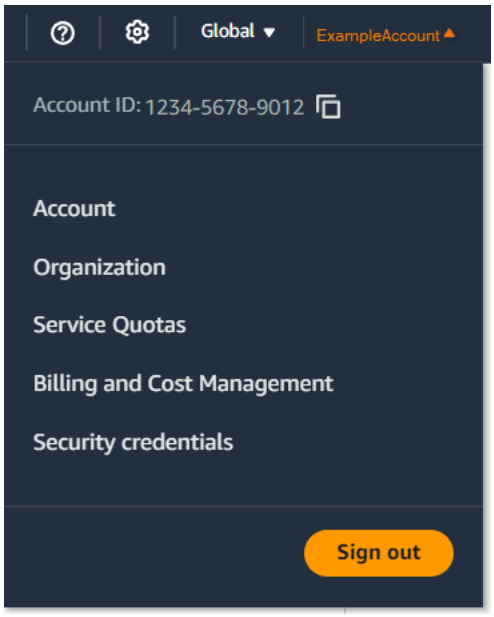
È possibile utilizzare il Console di gestione AWS per configurare e abilitare un dispositivo MFA virtuale per l'utente root. Per abilitare i dispositivi MFA per Account AWS, è necessario accedere AWS utilizzando le credenziali dell'utente root.

Come configurare e abilitare un dispositivo MFA virtuale da utilizzare con l'utente root (console)

1. Apri la console di gestione AWS e accedi utilizzando le credenziali dell'utente root.

   Per istruzioni, consulta Accedere Console di gestione AWS come utente root nella Guida per l'Accedi ad AWS utente.

2. Selezionare il nome dell'account sul lato destro della barra di navigazione e scegliere Security Credentials (Credenziali di sicurezza).



3. Nella sezione Multi-Factor Autenticazione (MFA), scegli Assegna dispositivo MFA.
4. Nella procedura guidata, digita un nome per il dispositivo, scegli l'app Authenticator e quindi scegli Next (Avanti).

   IAM genera e visualizza le informazioni di configurazione per il dispositivo MFA virtuale, tra cui il codice grafico QR. Il grafico è una rappresentazione della chiave di configurazione segreta che è disponibile per l'inserimento manuale su dispositivi che non supportano i codici QR.

5. Aprire l'app MFA virtuale sul dispositivo.

   Se l'app MFA virtuale supporta più account o dispositivi MFA virtuali, selezionare l'opzione che consente di creare un nuovo account o dispositivo MFA virtuale.

6. Il modo più semplice per configurare l'app è di utilizzare l'app per scannerizzare il codice QR. Se non è possibile scansionare il codice, è possibile digitare le informazioni di configurazione manualmente. Il codice QR e la chiave di configurazione segreta generati da IAM sono collegati al tuo Account AWS e non possono essere utilizzati con un altro account. Possono tuttavia

essere riutilizzati per configurare un nuovo dispositivo MFA per l'account nel caso in cui si perda l'accesso al dispositivo MFA originale.

- Per utilizzare il codice QR per la configurazione del dispositivo MFA virtuale, scegliere Show QR code (Mostra codice QR) nella procedura guidata. Quindi, seguire le istruzioni nell'app relative alla scansione del codice. Ad esempio, potrebbe essere necessario scegliere l'icona della fotocamera o un comando come Scan account barcode (Scannerizza codice a barre account) e utilizzare la fotocamera del dispositivo per eseguire la scansione del codice QR.
- Nella procedura guidata Set up device (Configura dispositivo), scegli Show secret key (Mostra chiave segreta) e digita la chiave segreta nell'app MFA.

> **Important**
>
> Effettuare un backup sicuro del codice QR o della chiave segreta di configurazione o assicurarsi che più dispositivi MFA virtuali siano abilitati per il proprio account. Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi MFA attualmente supportati con i tuoi utenti Utente root dell'account AWS e quelli IAM. Un dispositivo MFA virtuale potrebbe non essere più disponibili, ad esempio, se si perde lo smartphone, in cui il dispositivo MFA virtuale è ospitato. In tal caso e se non riesci ad accedere al tuo account senza dispositivi MFA aggiuntivi collegati all'utente né tramite Ripristino di un dispositivo MFA per l'utente root, non potrai accedere al tuo account e dovrai contattare l'assistenza clienti per rimuovere la protezione MFA per l'account.

Il dispositivo avvia la generazione di numeri a sei cifre.

7. Nella procedura guidata, nella casella MFA Code 1 (Codice MFA 1), digita la password monouso visualizzata nel dispositivo MFA virtuale. Attendere fino a un massimo di 30 secondi prima che il dispositivo generi una nuova password una tantum. Quindi, digitare la seconda password monouso nella casella Codice MFA 2. Scegli Aggiungi MFA.

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, il dispositivo MFA si associa correttamente con l'utente ma il dispositivo MFA non viene sincronizzato. Ciò accade

perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo è pronto per l'uso con. AWS Per ulteriori informazioni sull'utilizzo di MFA con la Console di gestione AWS, consulta Accesso abilitato con MFA.

Abilitazione di un token TOTP hardware per l'utente root dell' (console)

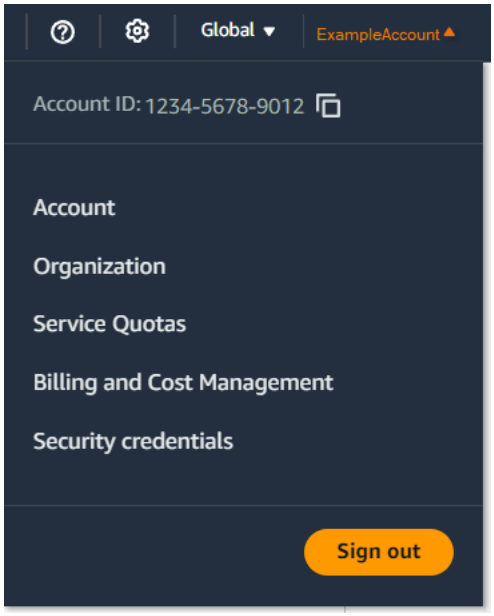
Puoi configurare e abilitare un dispositivo MFA fisico per il tuo utente root Console di gestione AWS solo dall'API, non dall'API AWS CLI o AWS .

> **Note**
>
> È possibile che il testo visualizzato sia differente, ad esempio Sign in using MFA (Accesso con un dispositivo MFA) e Troubleshoot your authentication device (Risoluzione dei problemi del dispositivo di autenticazione). Tuttavia, le funzionalità sono identiche. In entrambi i casi, se non fosse possibile verificare l'indirizzo e-mail e il numero di telefono dell'account utilizzando fattori alternativi dell'autenticazione, contatta Supporto AWS per eliminare l'impostazione MFA.

Per abilitare un token TOTP hardware per l'utente root (console)

1. Apri la console di gestione AWS e accedi utilizzando le credenziali dell'utente root.

   Per istruzioni, consulta Accedere Console di gestione AWS come utente root nella Guida per l'Accedi ad AWS utente.
2. Scegli il nome dell'account sul lato destro della barra di navigazione, quindi seleziona Credenziali di sicurezza.



3. Espandi la Multi-factor sezione di autenticazione (MFA).
4. Scegli Assegna dispositivo MFA.
5. Nella procedura guidata, digitate il nome del dispositivo, scegliete il token Hardware TOTP e quindi scegliete Avanti.
6. Nella casella Serial number (Numero di serie) digitare il numero di serie riportato sulla parte posteriore del dispositivo MFA.
7. Nella casella MFA code 1 (Codice MFA 1) digitare il numero di sei cifre visualizzato nel dispositivo MFA. Per visualizzare il numero, potrebbe essere necessario premere il pulsante sul lato anteriore del dispositivo.



8. Attendere 30 secondi per consentire al dispositivo di aggiornare il codice, quindi digitare il nuovo numero a sei cifre nella casella MFA code 2 (Codice MFA 2). Per visualizzare il secondo numero, potrebbe essere necessario premere nuovamente il pulsante sul lato anteriore del dispositivo.
9. Scegli Aggiungi MFA. Il dispositivo MFA è ora associato all'account Account AWS.

> **Important**
>
> La richiesta deve essere inviata immediatamente dopo la generazione dei codici di autenticazione. Se dopo avere generato i codici attendi troppo a lungo prima di inviare la richiesta, il dispositivo MFA si associerà correttamente con l'utente, ma perderà

la sincronizzazione. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Al successivo utilizzo delle credenziali dell'utente root per effettuare l'accesso, dovrai immettere un codice dal dispositivo MFA.

Cambia la password per Utente root dell'account AWS

Puoi modificare l'indirizzo e-mail e la password in Credenziali di sicurezza o nella pagina Account. Puoi anche scegliere Password dimenticata? nella pagina di AWS accesso per reimpostare la password.

Per modificare la password dell'utente root, devi accedere come utente IAM Utente root dell'account AWS e non come utente IAM. Per ulteriori informazioni su come reimpostare una password dell'utente root dimenticata, consulta Reimpostare una password dell'utente root persa o dimenticata.

Per proteggere la password, consigliamo di seguire queste best practice:

- Cambia periodicamente la password.
- Mantieni la password privata, perché chiunque la conosca può accedere al tuo account.
- Usa una password diversa da AWS quella che usi su altri siti.
- Evitare password che sono facili da indovinare. Queste includono password, come `secret`, `password`, `amazon` o `123456`. Sono inclusi anche i dati come una parole comuni, il tuo nome, l'indirizzo e-mail o altre informazioni personali che possono essere ottenute facilmente.

Important

Account AWS gestito utilizzando AWS Organizations potrebbe essere abilitato l'accesso root centralizzato per gli account dei membri. Questi account membri non dispongono di credenziali dell'utente root, non possono accedere come utente root e non possono recuperare la password dell'utente root. Contatta l'amministratore se devi eseguire un'operazione che richiede le credenziali dell'utente root.

Console di gestione AWS

Come modificare la password per l'utente root

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - È necessario accedere come utente Account AWS root, operazione che non richiede autorizzazioni AWS Identity and Access Management (IAM) aggiuntive. Non è possibile eseguire questi passaggi come utente o ruolo IAM.

1. Apri la Console di gestione AWS e accedi utilizzando le credenziali dell'utente root.

   Per istruzioni, consulta Accedi Console di gestione AWS come utente root nella Guida per l'Accedi ad AWS utente.
2. Nell'angolo in alto a destra della console, scegli il nome o il numero dell'account, quindi seleziona Credenziali di sicurezza.
3. Nella pagina Account, accanto a AImpostazioni account, scegli Modifica. Ti viene richiesto di effettuare nuovamente l'autenticazione per motivi di sicurezza.

   > **Note**
   >
   > Se non vedi l'opzione Modifica, è probabile che tu non abbia effettuato l'accesso come utente root dell'account. Non è possibile modificare le impostazioni dell'account dopo aver effettuato l'accesso come utente o ruolo IAM.
4. Nella pagina Aggiorna le impostazioni dell'account, in Password, scegli Modifica.
5. Nella pagina Aggiorna la password, compila i campi Password corrente, Nuova password e Conferma nuova password.

   > **Important**
   >
   > Scegli una password sicura. Anche se è possibile impostare una policy per le password dell'account per gli utenti IAM, tale policy non si applica all'utente root.

AWS richiede che la password soddisfi le seguenti condizioni:

- Deve avere un minimo di 8 caratteri e un massimo di 128 caratteri.
- Deve includere almeno tre dei seguenti tipi di caratteri: maiuscole, minuscole, numeri e i simboli ! @ # $ % ^ & * () <> [] {} | _+-=.
- Non deve essere identica al tuo Account AWS nome o indirizzo email.

6. Scegli Save changes (Salva modifiche).

AWS CLI or AWS SDK

Questa attività non è supportata in AWS CLI o da un'operazione API di uno degli AWS SDK. È possibile eseguire questa attività solo utilizzando. Console di gestione AWS

Reimpostare una password dell'utente root persa o dimenticata

Quando hai creato il tuo Account AWS, hai fornito un indirizzo email e una password. Queste sono le tue Utente root dell'account AWS credenziali. Se dimentichi la password dell'utente root, puoi reimpostarla dalla Console di gestione AWS.

Account AWS managed using AWS Organizations potrebbe avere l'accesso root centralizzato abilitato per gli account dei membri. Questi account membri non dispongono di credenziali dell'utente root, non possono accedere come utente root e non possono recuperare la password dell'utente root. Contatta l'amministratore se devi eseguire un'operazione che richiede le credenziali dell'utente root.

Important

Hai problemi ad accedere a AWS? Assicurati di essere nella pagina di accesso AWS corretta per il tuo tipo di utente. Se sei il Utente root dell'account AWS (proprietario dell'account), puoi accedere AWS utilizzando le credenziali che hai impostato quando hai creato il Account AWS. Se sei un utente IAM, l'amministratore dell'account può fornirti le credenziali che puoi utilizzare per accedere ad AWS. Se hai bisogno di richiedere assistenza, non utilizzare il link di feedback in questa pagina, poiché il modulo viene ricevuto dal team di AWS documentazione, no Supporto. Invece, nella pagina Contattaci, scegli Ancora impossibile accedere al tuo AWS account e quindi scegli una delle opzioni di supporto disponibili.

Per reimpostare la password dell'utente root

1. Apri la console di gestione AWS e accedi utilizzando le credenziali dell'utente root.

   Per istruzioni, consulta Accedi Console di gestione AWS come utente root nella Guida per l'Accedi ad AWS utente.

   > **Note**
   >
   > Se è stato effettuato l'accesso alla Console di gestione AWS con le credenziali dell'utente IAM, per reimpostare la password dell'utente root è necessario prima disconnettersi. Se vedi la pagina di accesso utente IAM specifica per l'account, scegli di Sign-in utilizzare le credenziali dell'account root nella parte inferiore della pagina. Se necessario, fornisci l'indirizzo e-mail dell'account e scegli Next (Avanti) per accedere alla pagina Root user sign in (Accesso utente root).

2. Selezionare Forgot your password? (Password dimenticata?).

   > **Note**
   >
   > Se sei un utente IAM, questa opzione non è disponibile. L'opzione Password dimenticata? è disponibile solo per l'account utente root. Gli utenti IAM devono chiedere al proprio amministratore di reimpostare una password dimenticata. Per maggiori informazioni, consulta Ho dimenticato la password utente IAM per il mio account. AWS Se accedi tramite il portale di AWS accesso, consulta Reimpostazione della password utente dell'IAM Identity Center.

3. Fornire l'indirizzo e-mail associato all'account. Fornire quindi il testo CAPTCHA e selezionare Continue (Continua).

4. Controlla l'email associata alla tua Account AWS per ricevere un messaggio da Amazon Web Services. L'e-mail proviene da un indirizzo che termina con `@verify.signin.aws`. Seguire le istruzioni nel messaggio. Se il messaggio e-mail non viene visualizzato nel proprio account, controllare la cartella spam. Se non hai più accesso all'email, consulta Non ho accesso all'email del mio AWS account nella Guida per l'Accedi ad AWS utente.

Creare chiavi di accesso per l'utente root

> **Warning**
>
> Consigliamo fortemente di non creare coppie di chiavi di accesso per l'utente root. Poiché solo alcune attività richiedono l'utente root e in genere tali attività vengono eseguite raramente, si consiglia di accedere a per Console di gestione AWS eseguire le attività dell'utente root. Prima di creare le chiavi di accesso, esamina le alternative alle chiavi di accesso a lungo termine.

Sebbene non lo consigliamo, puoi creare chiavi di accesso per l'utente root in modo da poter eseguire comandi in AWS Command Line Interface (AWS CLI) o utilizzare le operazioni API da uno degli AWS SDK utilizzando le credenziali dell'utente root. Quando crei una chiave di accesso, crei l'ID chiave di accesso e la chiave di accesso segreta come set. Durante la creazione della chiave di AWS accesso, hai l'opportunità di visualizzare e scaricare la parte della chiave di accesso segreta della chiave di accesso. Se non la scarichi o la perdi, puoi eliminare la chiave di accesso e quindi crearne una nuova. È possibile creare chiavi di accesso utente root con la console o AWS CLI l' AWS API.

Lo stato di una nuova chiave di accesso è attiva, il che significa che puoi usarla per le chiamate API e CLI. Inoltre, è possibile assegnare fino a due chiavi di accesso all'utente root.

Le chiavi di accesso non utilizzate dovrebbero essere disattivate. Dopo che una chiave di accesso è inattiva, non puoi utilizzarla per le chiamate API. Le chiavi inattive contano comunque per il limite. Puoi creare o eliminare una chiave di accesso in qualsiasi momento. Tuttavia, una volta eliminata, viene persa per sempre e non può essere recuperata.

Console di gestione AWS

Per creare una chiave di accesso per Utente root dell'account AWS

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:

- È necessario accedere come utente Account AWS root, operazione che non richiede autorizzazioni AWS Identity and Access Management (IAM) aggiuntive. Non è possibile eseguire questi passaggi come utente o ruolo IAM.

1. Apri la Console di gestione AWS e accedi utilizzando le credenziali dell'utente root.

   Per istruzioni, consulta Accedi Console di gestione AWS come utente root nella Guida per l'Accedi ad AWS utente.
2. Nell'angolo in alto a destra della console, seleziona il nome o il numero dell'account, quindi scegli Credenziali di sicurezza.
3. Nella sezione Chiavi di accesso, scegliere Crea chiave di accesso. Se questa opzione non è disponibile, è già stato raggiunto il numero massimo di chiavi di accesso. È necessario eliminare una delle chiavi di accesso esistenti prima di poter creare una nuova chiave. Per ulteriori informazioni, consulta la pagina Quote degli oggetti IAM.
4. Nella pagina Alternative alle chiavi di accesso dell'utente root, consulta le raccomandazioni di sicurezza. Per continuare, seleziona la casella di controllo, quindi scegli Crea chiave di accesso.
5. Nella pagina Recupera la chiave d'accesso, viene visualizzato l'ID della Chiave di accesso.
6. In Chiave di accesso segreta, seleziona Mostra, quindi copia l'ID della chiave di accesso e della chiave segreta dalla finestra del browser e incollale in un luogo sicuro. In alternativa, puoi selezionare Scarica file .csv: eseguirai il download di un file denominato `rootkey.csv` che contiene l'ID chiave di accesso e la chiave segreta. Salvare il file da qualche parte al sicuro.
7. Seleziona Fatto. Quando non hai più bisogno della chiave di accesso, ti consigliamo di eliminarla o almeno di valutare se disattivarla, in modo che nessuno possa usarla in modo improprio.

AWS CLI & SDKs

Per creare una chiave di accesso per l'utente root

> **Note**
>
> Per eseguire il seguente comando od operazione API come utente root, è necessario disporre già di una coppia di chiavi di accesso attive. Se non disponi delle chiavi di

accesso, crea la prima chiave di accesso utilizzando la Console di gestione AWS. Quindi, puoi utilizzare le credenziali della prima chiave di accesso con AWS CLI per creare la seconda chiave di accesso o per eliminare una chiave di accesso.

- AWS CLI: aws iam create-access-key

  Example

```
$ aws iam create-access-key
{
    "AccessKey": {
        "UserName": "MyUserName",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "Status": "Active",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "CreateDate": "2021-04-08T19:30:16+00:00"
    }
}
```

- AWS API: CreateAccessKey nell'API Reference di IAM.

Eliminare le chiavi di accesso per l'utente root

È possibile utilizzare l' Console di gestione AWS, the AWS CLI o l' AWS API per eliminare le chiavi di accesso dell'utente root.

Console di gestione AWS

Per eliminare una chiave di accesso per l'utente root

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - È necessario accedere come utente Account AWS root, il che non richiede autorizzazioni aggiuntive AWS Identity and Access Management (IAM). Non è possibile eseguire questi passaggi come utente o ruolo IAM.

1. Apri la Console di gestione AWS e accedi utilizzando le credenziali dell'utente root.

   Per istruzioni, consulta Accedere Console di gestione AWS come utente root nella Guida per l'Accedi ad AWS utente.
2. Nell'angolo in alto a destra della console, seleziona il nome o il numero dell'account, quindi scegli Credenziali di sicurezza.
3. Nella sezione Chiavi di accesso, individua la chiave di accesso che desideri eliminare, quindi scegli Operazioni e poi Elimina.

   > **Note**
   >
   > In alternativa, puoi disattivare una chiave di accesso anziché eliminarla definitivamente. In questo modo potrai riutilizzarla in futuro senza dover modificare l'ID chiave o la chiave segreta. Sebbene la chiave sia inattiva, qualsiasi tentativo di utilizzarla nelle richieste all' AWS API fallisce e viene negato l'errore di accesso.
4. Nella finestra di dialogo Elimina <ID chiave di accesso>, scegli Disattiva, inserisci l'ID della chiave di accesso per confermare che desideri eliminarla, quindi scegli Elimina.

AWS CLI & SDKs

Per eliminare una chiave di accesso per l'utente root

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - È necessario accedere come utente Account AWS root, operazione che non richiede autorizzazioni aggiuntive AWS Identity and Access Management (IAM). Non è possibile eseguire questi passaggi come utente o ruolo IAM.

- AWS CLI: era iam delete-access-key

  Example

```
$ aws iam delete-access-key \
```

```
    --access-key-id AKIAIOSFODNN7EXAMPLE
```

Questo comando non produce alcun output se ha esito positivo.

- AWS API: DeleteAccessKey

Attività che richiedono credenziali dell'utente root

Si consiglia di configurare un utente amministrativo Centro identità AWS IAM per eseguire le attività quotidiane e accedere alle AWS risorse. Tuttavia, è possibile eseguire le attività elencate di seguito solo se si effettua l'accesso come utente root di un account.

Per semplificare la gestione delle credenziali degli utenti root privilegiati tra gli account dei membri in AWS Organizations, puoi abilitare l'accesso root centralizzato per aiutarti a proteggere centralmente l'accesso altamente privilegiato al tuo. Account AWSGestire centralmente l'accesso root per gli account membriconsente di rimuovere e impedire in modo centralizzato il ripristino a lungo termine delle credenziali degli utenti root, migliorando la sicurezza dell'account nell'organizzazione. Dopo aver abilitato questa funzionalità, è possibile completare le seguenti attività con privilegi sugli account membri.

- Rimuovi le credenziali dell'utente root dell'account membro per impedire il recupero dell'account dell'utente root. Puoi anche consentire il recupero della password per recuperare le credenziali dell'utente root per un account membro.
- Rimuovi una policy del bucket configurata non correttamente che impedisce a tutti i principali di accedere al bucket Amazon S3.
- Elimina una policy basata sulle risorse Amazon Simple Queue Service che nega a tutti i principali l'accesso a una coda Amazon SQS.

Attività di gestione degli account

- Modifica le impostazioni del tuo Account AWS . Le versioni autonome Account AWS che non fanno parte di AWS Organizations richiedono le credenziali root per aggiornare l'indirizzo e-mail, la password dell'utente root e le chiavi di accesso dell'utente root. Altre impostazioni dell'account, come il nome dell'account, le informazioni di contatto, i contatti alternativi, la preferenza della valuta di pagamento e Regioni AWS, non richiedono credenziali dell'utente root.

> **Note**
>
> AWS Organizations, con tutte le funzionalità abilitate, può essere utilizzato per gestire centralmente le impostazioni degli account membro dall'account di gestione e dagli account amministratore delegato. Gli utenti IAM autorizzati o i ruoli IAM sia nell'account di gestione che negli account di amministrazione delegati possono chiudere gli account dei membri e aggiornare gli indirizzi e-mail principali, i nomi degli account, le informazioni di contatto, i contatti alternativi e Regioni AWS gli account dei membri.

- Chiudi il tuo. Account AWS Le versioni autonome Account AWS che non fanno parte di AWS Organizations richiedono le credenziali di root per chiudere l'account. Con AWS Organizations, puoi chiudere gli account dei membri centralmente dall'account di gestione e dagli account di amministratore delegato.
- Riapri il tuo. Account AWS Se hai chiuso personalmente un conto autonomo e non presenta alcun saldo in sospeso, puoi riaprirlo durante il periodo successivo alla chiusura. È necessario accedere come utente root per farlo.
- Ripristina le autorizzazioni dell'utente IAM. Se l'unico amministratore IAM revoca accidentalmente le autorizzazioni, sarà possibile effettuare l'accesso come utente root per modificare le policy e ripristinare le autorizzazioni.

Attività di fatturazione

- Attivare l'accesso IAM alla console di gestione fatturazione e costi.
- Alcune attività di fatturazione sono limitate all'utente root. Per ulteriori informazioni, consulta Managing an Account AWS in AWS Billing User Guide.
- Visualizzare alcune fatture fiscali. Un utente IAM con il portale aws: l'ViewBillingautorizzazione può visualizzare e scaricare le fatture IVA dall' AWS Europa, ma non da AWS Inc. o Amazon Internet Services Private Limited (AISPL).

AWS GovCloud (US) Processi

- Registrazione per AWS GovCloud (US).
- Richiedi le chiavi di AWS GovCloud (US) accesso utente root dell'account da. Supporto AWS

Attività di Amazon EC2

- Registrati come venditore nel marketplace di istanze riservate.

AWS KMS Operazione

- Nel caso in cui una AWS Key Management Service chiave diventi ingestibile, un amministratore può recuperarla contattandola Supporto; tuttavia, Supporto risponde al numero di telefono principale dell'utente root per l'autorizzazione confermando il ticket OTP.

Attività di Amazon Simple Storage Service

- Configura un bucket Amazon S3 per abilitare l'autenticazione a più fattori (MFA).
- Modifica o elimina una policy del bucket Amazon S3 che rifiuta tutti i principali.

  Puoi utilizzare azioni con privilegi per sbloccare un bucket Amazon S3 con una policy di bucket configurata non correttamente. Per informazioni dettagliate, vedi Esegui un'attività privilegiata su un AWS Organizations account membro.

Attività di Amazon Simple Queue Service

- Modifica o elimina una policy basata sulle risorse Amazon SQS che rifiuta tutti i principali.

  Puoi utilizzare azioni con privilegi per sbloccare una coda Amazon SQS con una policy basata sulle risorse configurata non correttamente. Per informazioni dettagliate, vedi Esegui un'attività privilegiata su un AWS Organizations account membro.

Risorse aggiuntive

Per ulteriori informazioni sull'utente AWS root, consulta le seguenti risorse:

- Per assistenza con i problemi relativi agli utenti root, consulta Risoluzione dei problemi con l'utente root.
- Per gestire centralmente gli indirizzi e-mail degli utenti root in AWS Organizations, vedere Aggiornamento dell'indirizzo e-mail dell'utente root per un account membro nella Guida per l'AWS Organizations utente.

I seguenti articoli forniscono ulteriori informazioni sull'utilizzo dell'utente root.

- Quali sono le migliori pratiche per proteggere le mie Account AWS e le relative risorse?
- Come posso creare una regola di EventBridge evento per informarmi che è stato utilizzato il mio utente root?
- Monitora e invia notifiche sulle Utente root dell'account AWS attività
- Monitoraggio dell'attività dell'utente root IAM

Utenti IAM

> **Important**
>
> Le best practice IAM consigliano di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee invece di utilizzare utenti IAM con credenziali a lungo termine. Ti consigliamo di utilizzare gli utenti IAM solo per casi d'uso specifici non supportati dagli utenti federati.
> Se stai utilizzando la nostra nuova AWS esperienza e desideri fornire agli utenti umani l'accesso ai tuoi progetti o alle tue AWS impostazioni, devi aggiungerli al tuo team. Nella nostra nuova esperienza, non puoi creare utenti IAM con accesso alla console. Per maggiori informazioni, consulta Invita i membri del team.

Un utente IAM è un'entità che viene creata nell' Account AWS. L'utente IAM rappresenta la persona o il carico di lavoro utilizzato dall'utente IAM per interagire con le risorse AWS . Un utente IAM dispone di un nome e di credenziali.

Un utente IAM con le autorizzazioni di amministratore non è la stessa cosa dell' Utente root dell'account AWS. Per ulteriori informazioni sull'utilizzo dell'utente root, consulta Utente root dell'account AWS.

In che modo AWS identifica un utente IAM

Quando crei un utente IAM, IAM crea questi metodi per identificare quell'utente:

- Un "nome semplice" per l'utente IAM, che è il nome specificato quando hai creato l'utente IAM, ad esempio `Richard` o `Anaya`. Questi sono i nomi che vedi nella Console di gestione AWS. Poiché i nomi utente IAM compaiono nel nome della risorsa Amazon (ARN), non è consigliabile includere

informazioni di identificazione personale nel nome IAM. Fai riferimento a Requisiti del nome IAM per i requisiti e le restrizioni per i nomi IAM.

- Un nome della risorsa Amazon (ARN) per l'utente IAM. L'ARN viene utilizzato quando è necessario identificare in modo univoco l'utente IAM in tutti i settori. AWS Ad esempio, puoi usare un ARN per specificare l'utente IAM come `Principal` in una policy IAM per un bucket Amazon S3. Un ARN per un utente IAM potrebbe essere simile al seguente:

  `arn:aws:iam::account-ID-without-hyphens:user/Richard`

- Un identificatore univoco per l'utente IAM. Questo ID viene restituito solo quando utilizzi l'API, Tools for Windows PowerShell o AWS CLI per creare l'utente IAM; non viene visualizzato nella console.

Per ulteriori informazioni su questi identificatori, consulta Identificatori IAM.

Utenti IAM e credenziali

Puoi accedere AWS in diversi modi a seconda delle credenziali dell'utente IAM:

- Password console: una password che l'utente IAM può inserire per accedere a sessioni interattive come la Console di gestione AWS. La disabilitazione della password (accesso alla console) per un utente IAM impedisce a quest'ultimo di accedere Console di gestione AWS utilizzando le proprie credenziali di accesso. Non modifica le autorizzazioni né impedisce l'accesso alla console utilizzando un ruolo assunto. Gli utenti IAM con accesso alla console abilitato possono anche utilizzare queste stesse credenziali per autenticarsi AWS CLI e accedere all'SDK utilizzando il comando. `aws login` AWS CLI Questi utenti dovranno disporre delle autorizzazioni. SignInLocalDevelopmentAccess Per ulteriori dettagli, vedere Credenziali di autenticazione e accesso AWS CLI nella Guida per AWS Command Line Interface l'utente.
- Tasti di accesso: utilizzati per effettuare chiamate programmatiche a AWS. Tuttavia, ci sono alternative più sicure da considerare prima di creare le chiavi di accesso per gli utenti IAM. Per ulteriori informazioni, consulta Considerazioni e alternative per le chiavi di accesso a lungo termine nella Riferimenti generali di AWS. Se l'utente IAM dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l' AWS CLI accesso tramite Tools for Windows PowerShell, AWS API o l'applicazione mobile AWS Console.
- Chiavi SSH da utilizzare con CodeCommit: una chiave pubblica SSH nel formato OpenSSH che può essere utilizzata per eseguire l'autenticazione con CodeCommit.
- Certificati server : SSL/TLS certificati che puoi utilizzare per autenticarti con alcuni AWS servizi. Ti consigliamo di utilizzare AWS Certificate Manager (ACM) per fornire, gestire e distribuire i

certificati del server. Utilizza IAM solo quando è necessario il supporto alle connessioni HTTPS in una regione che non è supportata da ACM. Per informazioni sulle regioni che supportano ACM, consulta Endpoint e quote di AWS Certificate Manager nella Riferimenti generali di AWS.

È possibile scegliere le credenziali che meglio si adattano all'utente IAM. Quando utilizzi la Console di gestione AWS per creare un utente IAM, devi scegliere di includere almeno una password o delle chiavi di accesso alla console. Per impostazione predefinita, un nuovo utente IAM creato utilizzando l' AWS API AWS CLI or non ha credenziali di alcun tipo. Il tipo di credenziali dell'utente IAM da creare dipende dal caso d'uso.

Hai le seguenti opzioni per amministrare le password, le chiavi di accesso e i dispositivi con l'autenticazione a più fattori (MFA):

- Gestione di password per gli utenti IAM. Crea e modifica le password che consentono l'accesso alla Console di gestione AWS. Imposta una policy per la password, così da implementare un minimo di complessità per la password. Consenti agli utenti IAM di cambiare le loro password.
- Gestione delle chiavi di accesso per gli utenti IAM. Crea e aggiorna le chiavi di accesso per l'accesso programmatico alle risorse nel tuo account.
- Abilita l'utente IAM all'autenticazione a più fattori (MFA). Come best practice, ti consigliamo di richiedere l'autenticazione a più fattori per tutti gli utenti IAM nel tuo account. Con l'MFA, gli utenti IAM devono fornire due forme di identificazione. Innanzitutto, forniscono le credenziali che fanno parte dell'identità utente (una password o una chiave di accesso). Inoltre, forniscono un codice numerico temporaneo generato su un dispositivo hardware o da un'applicazione su uno smartphone o un tablet.
- Trovare password e chiavi di accesso non utilizzate. Chiunque abbia una password o una chiave di accesso per il tuo account o un utente IAM nel tuo account ha accesso alle tue AWS risorse. La best practice di sicurezza consiste nel rimuovere le password e le chiavi di accesso quando gli utenti IAM non ne hanno più bisogno.
- Download di un report delle credenziali per l'account. È possibile generare e scaricare un report delle credenziali che riporta tutti gli utenti IAM presenti nell'account e lo stato delle loro diverse credenziali, tra cui password, chiavi di accesso e dispositivi MFA. Per le password e le chiavi di accesso, il report sulle credenziali mostra se la password o la chiave di accesso siano state utilizzate di recente.

Utenti e autorizzazioni IAM

Per impostazione predefinita, un nuovo utente IAM non ha le autorizzazioni per svolgere alcuna operazione. Non sono autorizzati a eseguire alcuna AWS operazione o ad accedere ad alcuna AWS risorsa. Un vantaggio di avere singoli utenti IAM è quello di poter assegnare le autorizzazioni individualmente a ogni utente. Potresti assegnare autorizzazioni amministrative ad alcuni utenti, che possono quindi amministrare AWS le tue risorse e persino creare e gestire altri utenti IAM. Nella maggior parte dei casi, tuttavia, è necessario limitare le autorizzazioni di un utente solo alle attività (AWS azioni o operazioni) e alle risorse necessarie per il lavoro.

Prendiamo a esempio un utente denominato Diego. Quando crei l'utente IAM `Diego`, crei una password per questo utente e colleghi le autorizzazioni all'utente, per permettergli di avviare una determinata istanza Amazon EC2 e leggere le informazioni (`GET`) da una tabella in un database Amazon RDS. Per le procedure su come creare gli utenti IAM e concedere loro le credenziali iniziali e le autorizzazioni, consulta Crea un utente IAM nel tuo Account AWS. Per le procedure su come modificare le autorizzazioni agli utenti esistenti, consulta Modificare le autorizzazioni per un utente IAM. Per le procedure su come cambiare la password dell'utente o le chiavi di accesso, consulta la pagina Password utente in AWS e Gestione delle chiavi di accesso per gli utenti IAM.

Puoi anche aggiungere un limite delle autorizzazioni agli utenti IAM. Un limite di autorizzazioni è una funzionalità avanzata che consente di utilizzare policy AWS gestite per limitare le autorizzazioni massime che una policy basata sull'identità può concedere a un utente o ruolo IAM. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Politiche e autorizzazioni in AWS Identity and Access Management.

Utenti IAM e account

Ogni utente IAM è associato a un solo Account AWS. Poiché gli utenti IAM sono definiti all'interno del tuo Account AWS, non è necessario che dispongano di un metodo di pagamento registrato. AWS Qualsiasi AWS attività eseguita dagli utenti IAM nel tuo account viene fatturata sul tuo account.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Utenti IAM come account di servizio

Un utente IAM è una risorsa in IAM con credenziali e autorizzazioni associate. Un utente IAM può rappresentare una persona o un'applicazione che utilizza le proprie credenziali per effettuare richieste

AWS . In genere questo si chiama account di servizio. Se nella tua applicazione scegli di utilizzare le credenziali a lungo termine di un utente IAM, non integrare le chiavi di accesso direttamente nel codice dell'applicazione. Gli AWS SDK e il AWS Command Line Interface consentono di inserire le chiavi di accesso in posizioni note in modo da non doverle conservare in codice. Per ulteriori informazioni, consulta Gestione corretta delle chiavi di accesso dell'utente IAM nella Riferimenti generali di AWS. Oppure, come best practice, puoi utilizzare le credenziali di sicurezza temporanee (ruoli IAM) al posto delle chiavi di accesso a lungo termine.

Dopo aver creato gli utenti IAM, puoi farlo Visualizzare gli utenti IAM nel tuo account e Ridenominare un utente IAM se necessario.

In che modo gli utenti IAM accedono a AWS

Per accedere Console di gestione AWS come utente IAM, devi fornire l'ID dell'account o l'alias dell'account oltre al nome utente e alla password. Quando l'amministratore ha creato l'utente IAM nella console, dovrebbero aver inviato le credenziali di accesso, ovvero nome utente e URL della pagina di accesso dell'account che include l'ID account o l'alias dell'account.

```
https://My_AWS_Account_ID.signin.aws.amazon.com/console/
```

> **Suggerimento**
>
> Per creare un segnalibro per la pagina di accesso al tuo account nel tuo browser Web, devi digitare manualmente l'URL della pagina di accesso per il tuo account nella voce segnalibro. Non utilizzare la funzione di segnalibro del tuo browser Web perché gli indirizzamenti possono oscurare l'URL della pagina di accesso.

Puoi effettuare l'accesso anche al seguente endpoint generale di accesso e digitare manualmente l'ID account o l'alias dell'account:

```
https://console.aws.amazon.com/
```

Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. La volta successiva che l'utente accede a qualsiasi pagina del Console di gestione AWS, la console utilizza il cookie per reindirizzare l'utente alla pagina di accesso dell'account.

Hai accesso solo alle AWS risorse specificate dall'amministratore nella policy allegata alla tua identità utente IAM. Per lavorare nella console, devi disporre delle autorizzazioni per eseguire le azioni eseguite dalla console, come elencare e creare AWS risorse. Per ulteriori informazioni, consultare Gestione degli accessi per AWS risorse e Esempi di policy basate su identità IAM.

> **Note**
>
> Se la tua organizzazione dispone di un suo sistema di identità, puoi pensare di creare un'opzione Single Sign-On (SSO). L'SSO consente agli utenti di accedere al Console di gestione AWS tuo account senza richiedere loro un'identità utente IAM. L'SSO elimina inoltre la necessità per gli utenti di accedere al sito dell'organizzazione e separatamente. AWS Per ulteriori informazioni, consulta Abilita l'accesso personalizzato del broker di identità al AWS console.

Registrazione dei dati di accesso CloudTrail

Se abiliti CloudTrail la registrazione degli eventi di accesso nei tuoi registri, devi sapere come CloudTrail sceglie dove registrare gli eventi.

- Se gli utenti accedono direttamente a una console, vengono reindirizzati a un endpoint di accesso globale o regionale, a seconda se la console del servizio selezionato supporti o meno le regioni. Ad esempio, la homepage della console principale supporta le regioni, perciò se effettui l'accesso al seguente URL:

  ```
  https://alias.signin.aws.amazon.com/console
  ```

  verrai reindirizzato a un endpoint di accesso regionale, ad esempio`https://us-east-2.signin.aws.amazon.com`, generando una voce di CloudTrail registro regionale nel registro della regione dell'utente:

  D'altra parte, la console Amazon S3 non supporta le regioni, perciò se si effettua l'accesso al seguente URL:

  ```
  https://alias.signin.aws.amazon.com/console/s3
  ```

  AWS ti reindirizza all'endpoint di accesso globale all'indirizzo`https://signin.aws.amazon.com`, generando una voce di registro globale. CloudTrail

- È possibile richiedere manualmente un determinato endpoint di accesso regionale tramite l'accesso alla homepage della console principale con attivazione regionale, utilizzando una sintassi di URL simile a quella riportata di seguito:

  ```
  https://alias.signin.aws.amazon.com/console?region=ap-southeast-1
  ```

  AWS ti reindirizza all'endpoint di accesso `ap-southeast-1` regionale e genera un evento di registro regionale. CloudTrail

Per ulteriori informazioni su CloudTrail e IAM, consulta Registrazione degli eventi IAM con. CloudTrail

Se gli utenti richiedono un accesso programmatico per funzionare con il tuo account, puoi creare una coppia di chiavi di accesso (un ID chiave di accesso e una chiave di accesso segreta) per ciascun utente, come descritto in . Tuttavia, ci sono alternative più sicure da considerare prima di creare le chiavi di accesso per gli utenti. Per ulteriori informazioni, consulta Considerazioni e alternative per le chiavi di accesso a lungo termine nella Riferimenti generali di AWS.

Risorse aggiuntive

Le seguenti risorse possono aiutarti a saperne di più sull' AWS accesso.

- La Guida per AWS Sign-In l'utente ti aiuta a comprendere i diversi modi in cui puoi accedere ad Amazon Web Services (AWS), a seconda del tipo di utente che sei.
- Puoi accedere contemporaneamente a un massimo di cinque identità diverse in un unico browser web nella Console di gestione AWS. Per maggiori dettagli, consulta Accesso a più account nella Guida introduttiva alla Console di gestione AWS .
- Per trovare l'ID dell'account o l'alias necessario per l'accesso, consulta Visualizzazione del tuo Account AWS ID e. Usare un alias per il tuo Account AWS ID

Accesso abilitato con MFA

Gli utenti configurati con dispositivi di autenticazione a più fattori (MFA) devono utilizzare i propri dispositivi MFA per accedere alla Console di gestione AWS. Dopo che l'utente ha inserito le proprie credenziali di accesso, AWS controlla l'account dell'utente per vedere se l'MFA è richiesta per quell'utente.

> **Important**
>
> Se si utilizzano le credenziali della chiave di accesso e della chiave segreta per Console di gestione AWS l'accesso diretto con la chiamata AWS STS `GetFederationToken`API, l'autenticazione MFA NON sarà richiesta. Per ulteriori informazioni, consulta Utilizzo delle chiavi di accesso e delle credenziali segrete per l'accesso alla console.

Gli argomenti seguenti forniscono informazioni su come gli utenti completano l'accesso quando è necessaria l'autenticazione MFA.

Argomenti

- Più dispositivi MFA abilitati
- Chiave di sicurezza FIDO
- Dispositivo MFA virtuale
- Token TOTP hardware

Più dispositivi MFA abilitati

Se un utente accede Console di gestione AWS come utente Account AWS root o utente IAM con più dispositivi MFA abilitati per quell'account, deve utilizzare un solo dispositivo MFA per accedere. Dopo l'autenticazione con la password dell'utente, l'utente seleziona il tipo di dispositivo MFA che desidera utilizzare per completare l'autenticazione. Quindi all'utente viene richiesto di autenticarsi con il tipo di dispositivo selezionato.

Chiave di sicurezza FIDO

Se l'autenticazione MFA è obbligatorio per l'utente, viene visualizzata una seconda pagina di accesso. L'utente deve toccare la chiave di sicurezza FIDO.

> **Note**
>
> Gli utenti di Chrome non devono scegliere alcuna delle opzioni disponibili nel pop-up di Google Chrome che chiede di verificare la tua identità con amazon.com. Limitati a toccare la chiave di sicurezza.

A differenza di altri dispositivi MFA, le chiavi di sicurezza FIDO sono sempre aggiornate. Se una chiave di sicurezza FIDO viene smarrita o danneggiata, gli amministratori possono disattivarla. Per ulteriori informazioni, consulta Disattivazione dei dispositivi MFA (console).

Per informazioni sui browser che supportano WebAuthn e sui dispositivi compatibili con FIDO che supportano, vedere. AWS Configurazioni supportate per l'uso delle passkey e delle chiavi di sicurezza

Dispositivo MFA virtuale

Se l'autenticazione MFA è obbligatorio per l'utente, viene visualizzata una seconda pagina di accesso. Nella casella MFA code (Codice MFA), l'utente deve immettere il codice numerico fornito dall'applicazione MFA.

Se il codice MFA è corretto, l'utente può accedere alla Console di gestione AWS. Se il codice non è corretto, l'utente può riprovare con un altro codice.

Un dispositivo MFA virtuale può andare fuori sincrono. Se un utente non riesce ad accedere Console di gestione AWS dopo diversi tentativi, all'utente viene richiesto di sincronizzare il dispositivo MFA virtuale. L'utente può seguire le istruzioni mostrate sullo schermo per sincronizzare il dispositivo MFA virtuale. Per informazioni su come sincronizzare un dispositivo per conto di un utente del tuo paese, consulta. Account AWSRisincronizzare i dispositivi MFA virtuali e hardware

Token TOTP hardware

Se l'autenticazione MFA è obbligatorio per l'utente, viene visualizzata una seconda pagina di accesso. Nella casella MFA code (Codice MFA), l'utente deve immettere il codice numerico fornito dal token TOTP hardware.

Se il codice MFA è corretto, l'utente può accedere alla Console di gestione AWS. Se il codice non è corretto, l'utente può riprovare con un altro codice.

Un token TOTP hardware può non essere sempre sincronizzato. Se un utente non riesce ad accedere Console di gestione AWS dopo diversi tentativi, all'utente viene richiesto di sincronizzare il dispositivo token MFA. L'utente può seguire le istruzioni visualizzate per sincronizzare il dispositivo token MFA. Per informazioni su come sincronizzare un dispositivo per conto di un utente del tuo, consulta. Account AWSRisincronizzare i dispositivi MFA virtuali e hardware

Crea un utente IAM nel tuo Account AWS

> **Important**
>
> Le best practice IAM consigliano di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee invece di utilizzare utenti IAM con credenziali a lungo termine. Ti consigliamo di utilizzare gli utenti IAM solo per casi d'uso specifici non supportati dagli utenti federati.
> Se stai utilizzando la nostra nuova AWS esperienza e desideri fornire agli utenti umani l'accesso ai tuoi progetti o alle tue AWS impostazioni, devi aggiungerli al tuo team. Nella nostra nuova esperienza, non puoi creare utenti IAM con accesso alla console. Per maggiori informazioni, consulta Invita i membri del team.

Il processo con cui si crea un utente IAM e gli si consente di eseguire attività è costituito dalle fasi seguenti:

1. Crea l'utente negli Console di gestione AWS, negli AWS CLI, Strumenti per Windows PowerShell o utilizzando un'operazione AWS API. Se crei l'utente in Console di gestione AWS, i passaggi 1—4 vengono gestiti automaticamente, in base alle tue scelte. Se crei gli utenti IAM in modo programmatico, allora devi eseguire ognuna di queste fasi singolarmente.
2. Creazione delle credenziali per l'utente, a seconda del tipo di accesso che l'utente richiede:
   - Abilita l'accesso alla console, opzionale : se l'utente deve accedere a Console di gestione AWS, crea una password per l'utente. La disabilitazione dell'accesso alla console per un utente gli impedisce di accedere alla Console di gestione AWS tramite il nome utente e la password. Non modifica le autorizzazioni né impedisce l'accesso alla console utilizzando un ruolo assunto.

   > **Tip**
   >
   > Crea solo le credenziali di cui l'utente necessita. Ad esempio, per un utente che richiede l'accesso solo tramite Console di gestione AWS, non creare chiavi di accesso.

3. Concedi all'utente le autorizzazioni per eseguire le attività richieste. Consigliamo di inserire gli utenti IAM in gruppi e gestire le autorizzazioni tramite le policy collegate a tali gruppi. Tuttavia, puoi concedere autorizzazioni anche collegando le policy di autorizzazione direttamente all'utente. Se usi la console per aggiungere l'utente, puoi copiare le autorizzazioni da un utente esistente al nuovo utente.

Puoi anche aggiungere un limite delle autorizzazioni per limitare le autorizzazioni dell'utente specificando una policy che definisce le autorizzazioni massime che l'utente può avere. I limiti delle autorizzazioni non concedono alcuna autorizzazione.

Per istruzioni sulla creazione di una policy di autorizzazione personalizzata da utilizzare per concedere le autorizzazioni o impostare un limite delle autorizzazioni, consulta Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente.

4. (Facoltativo) Aggiungere metadati all'utente collegando tag. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.
5. Fornisci all'utente le necessarie informazioni di accesso. Queste informazioni includono la password e l'URL della console per la pagina di accesso all'account in cui l'utente immette tali credenziali. Per ulteriori informazioni, consulta In che modo gli utenti IAM accedono a AWS.
6. (Facoltativo) Configura multi-factor authentication (MFA) per l'utente. MFA richiede che l'utente fornisca un codice utilizzabile una sola volta, ogni volta che accede alla Console di gestione AWS.
7. (Facoltativo) Fornisci agli utenti IAM le autorizzazioni per gestire le proprie credenziali di sicurezza. (Per impostazione predefinita, gli utenti IAM non dispongono delle autorizzazioni per gestire le proprie credenziali). Per ulteriori informazioni, consulta Consentire agli utenti IAM di cambiare le loro password.

> **Note**
>
> Se si utilizza la console per creare l'utente e si seleziona L'utente deve creare una nuova password all'accesso successivo (consigliato), l'utente dispone delle autorizzazioni necessarie.

Per informazioni sulle autorizzazioni di cui hai bisogno per creare un utente, consulta la pagina Autorizzazioni necessarie per accedere alle risorse IAM.

Per istruzioni sulla creazione di utenti IAM per casi d'uso specifici, consulta i seguenti argomenti:

- Creare un utente IAM per l'accesso di emergenza
- Creare un utente IAM per carichi di lavoro che non possono utilizzare i ruoli IAM

Visualizzare gli utenti IAM

Puoi elencare gli utenti IAM nel tuo Account AWS o in uno specifico gruppo IAM ed elencare tutti i gruppi IAM in cui si trova un utente. Per informazioni sulle autorizzazioni necessarie per elencare gli utenti, consulta Autorizzazioni necessarie per accedere alle risorse IAM.

Per visualizzare tutti gli utenti IAM nel tuo account

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.

La console mostra gli utenti IAM del tuo Account AWS.

AWS CLI

Esegui il comando seguente:

- `aws iam list-users`

API

Chiamare l'operazione seguente:

- `ListUsers`

Per elencare gli utenti IAM in un gruppo IAM

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Gruppi di utenti.

4. Scegli il nome del gruppo di utenti.

Gli utenti IAM che sono membri del gruppo sono elencati nella scheda Utenti.

AWS CLI

Esegui il comando seguente:

- `aws iam get-group`

API

Chiamare l'operazione seguente:

- `GetGroup`

Per elencare tutti i gruppi IAM in cui si trova un utente

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Nell'elenco degli Utenti, seleziona il nome dell'utente IAM.
5. Seleziona la scheda Gruppi per visualizzare l'elenco dei gruppi IAM che includono l'utente corrente.

AWS CLI

Esegui il comando seguente:

- `aws iam list-groups-for-user`

API

Chiamare l'operazione seguente:

- `ListGroupsForUser`

Fasi successive

Una volta che hai un elenco dei tuoi utenti IAM, puoi rinominare, eliminare o disattivare un utente IAM utilizzando le seguenti procedure.

- Ridenominare un utente IAM
- Rimuovere o disattivare un utente IAM

Ridenominare un utente IAM

> **Note**
>
> Come procedura consigliata, si consiglia di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee. Se segui le best practice, non gestisci utenti e gruppi IAM. Gli utenti e i gruppi sono invece gestiti all'esterno AWS e possono accedere alle AWS risorse come identità federata. Un'identità federata è un utente dell'elenco utenti aziendale, un provider di identità Web, il AWS Directory Service, la directory Identity Center o qualsiasi utente che accede AWS ai servizi utilizzando le credenziali fornite tramite un'origine di identità. Le identità federate utilizzano i gruppi definiti dal rispettivo gestore di identità. Se lo utilizzi Centro identità AWS IAM, consulta Gestisci le identità in IAM Identity Center nella Guida per l'Centro identità AWS IAM utente per informazioni sulla creazione di utenti e gruppi in IAM Identity Center.

Amazon Web Services offre vari strumenti per gestire gli utenti IAM nel proprio Account AWS. Puoi elencare gli utenti IAM nel tuo account o in un gruppo di utenti oppure elencare tutti i gruppi IAM di cui un utente è membro. È possibile rinominare o modificare il percorso di un utente IAM. Se desideri utilizzare le identità federate invece degli utenti IAM, puoi eliminare un utente IAM dall'account AWS o disattivarlo.

Per ulteriori informazioni sull'aggiunta, la modifica o la rimozione di policy gestite per un utente IAM, consulta Modificare le autorizzazioni per un utente IAM. Per informazioni sulla gestione di policy in linea per utenti IAM, consulta Aggiunta e rimozione di autorizzazioni per identità IAM, Modificare le policy IAM e Eliminare le policy IAM. Come best practice, utilizza le policy gestite anziché le policy in linea. Le policy gestite da AWS concedono autorizzazioni per numerosi casi d'uso comuni. Tieni

presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per l'uso da parte di tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo Policy gestite dal cliente specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta AWS policy gestite. Per ulteriori informazioni sulle politiche AWS gestite progettate per funzioni lavorative specifiche, consulta. AWS politiche gestite per le funzioni lavorative

Per ulteriori informazioni sulla convalida delle policy IAM, consulta Convalida delle policy IAM.

> **Tip**
>
> IAM Access Analyzer analizza i servizi e le azioni utilizzati dai tuoi ruoli IAM e quindi genera una policy dettagliata che puoi utilizzare. Dopo aver testato ogni policy generata, puoi distribuirla nell'ambiente di produzione. In questo modo si garantisce di concedere solo le autorizzazioni necessarie ai carichi di lavoro. Per ulteriori informazioni sulla generazione delle policy, consulta IAM Access Analyzer policy generation.

Per informazioni sulla gestione delle password utente IAM, consulta Gestione delle password per gli utenti IAM.

Ridenominazione di un utente IAM

Per modificare il nome o il percorso di un utente, è necessario utilizzare gli AWS CLI strumenti per Windows PowerShell o AWS l'API. Non è disponibile alcuna opzione nella console per rinominare un utente. Per informazioni sulle autorizzazioni necessarie per ridenominare un utente, consulta Autorizzazioni necessarie per accedere alle risorse IAM.

Quando si modifica il nome o il percorso di un utente, si verificano i seguenti eventi:

- Qualsiasi policy collegata all'utente viene mantenuta per l'utente con il nuovo nome.
- L'utente rimane negli stessi gruppi IAM con il nuovo nome.
- L'ID univoco dell'utente rimane invariato. Per ulteriori informazioni su unique IDs, consultaIdentificatori univoci.
- Qualsiasi policy relativa alle risorse o ai ruoli che fa riferimento all'utente come principale (l'utente a cui viene consentito l'accesso) viene automaticamente aggiornata per l'utilizzo del nuovo nome o percorso. Ad esempio, qualsiasi policy basata su code in Amazon SQS o basata sulle risorse in Amazon S3 viene aggiornata automaticamente per utilizzare il nuovo nome e percorso.

IAM non aggiorna automaticamente le policy che fanno riferimento all'utente come una risorsa per l'utilizzo del nuovo nome o percorso, è necessario un aggiornamento manuale. Ad esempio, si immagini che l'utente `Richard` disponga di una policy collegata che permette di gestire le credenziali di sicurezza dell'utente. Se un amministratore rinomina `Richard` in `Rich`, l'amministratore deve anche aggiornare questa policy per modificar la risorsa da:

```
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/Richard
```

a:

```
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/Rich
```

Ciò è valido anche se un amministratore cambia il percorso, l'amministratore deve aggiornare la policy in base al nuovo percorso per l'utente.

Per rinominare un utente

- AWS CLI: aws iam update-user
- AWS API: UpdateUser

Rimuovere o disattivare un utente IAM

Le migliori pratiche consigliano di rimuovere gli utenti IAM non utilizzati dal tuo Account AWS. Se desideri conservare le credenziali dell'utente IAM per uso futuro, invece di eliminarle dall'account è possibile disattivare l'accesso dell'utente. Per ulteriori informazioni, consulta Disattivazione di un utente IAM.

> **Warning**
>
> Una volta che un utente IAM e le relative chiavi di accesso vengono eliminate, non possono essere ripristinate o recuperate.

Prerequisito: visualizzazione dell'accesso dell'utente IAM

Prima di rimuovere un utente, esamina la sua attività recente a livello di servizio. Questo aiuta a prevenire la rimozione dell'accesso da parte di un principale (persona o applicazione) che lo utilizza.

Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Rimozione di un utente IAM (console)

Quando utilizzi Console di gestione AWS per rimuovere un utente IAM, IAM elimina automaticamente le seguenti informazioni associate:

- L'identificatore dell'utente IAM
- Qualsiasi appartenenza al gruppo, ovvero, l'utente IAM viene rimosso da qualsiasi gruppo di cui l'utente IAM era membro
- Qualsiasi password associata all'utente IAM
- Tutte le policy in linea integrate nell'utente IAM (le policy applicate a un utente IAM tramite le autorizzazioni del gruppo di utenti non sono interessate)

> **Note**
>
> IAM rimuove tutte le policy gestite collegate all'utente IAM quando si elimina l'utente, ma non elimina le policy gestite.

- Qualsiasi dispositivo MFA associato

Per rimuovere un utente IAM (console)

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel riquadro di navigazione seleziona Utenti, quindi seleziona la casella di controllo accanto al nome dell'utente IAM da eliminare.
4. Nella parte superiore della pagina, scegli Delete (Elimina).

> **Note**
>
> Se uno qualsiasi degli utenti dispone di chiavi di accesso attive, è necessario disattivarle prima di eliminare gli utenti. Per ulteriori informazioni, consulta Per disattivare una chiave di accesso per un utente IAM.

5. Nella casella di dialogo di conferma, inserisci il nome utente nel campo di inserimento testo per confermare l'eliminazione dell'utente. Scegli Elimina.

La console visualizza una notifica di stato che indica che l'utente IAM è stato eliminato.

Eliminazione di un utente IAM (AWS CLI)

A differenza di Console di gestione AWS, quando elimini un utente IAM con AWS CLI, devi eliminare manualmente gli elementi collegati all'utente IAM. Questa procedura illustra il processo.

Per eliminare un utente IAM dal tuo Account AWS (AWS CLI)

1. Eliminare la password dell'utente, se l'utente ne ha una.

   `aws iam delete-login-profile`
2. Eliminare le chiavi di accesso dell'utente, se disponibili.

   `aws iam list-access-keys` (per elencare le chiavi di accesso dell'utente) e `aws iam delete-access-key`
3. Eliminare il certificato di firma dell'utente. Si noti che quando si eliminano delle credenziali di sicurezza queste vengono viene eliminate per sempre e non possono più essere recuperate.

   `aws iam list-signing-certificates` (per elencare i certificati di firma dell'utente) e `aws iam delete-signing-certificate`
4. Eliminare la chiave pubblica SSH dell'utente, se disponibile.

   `aws iam list-ssh-public-keys` (per elencare le chiavi pubbliche SSH dell'utente) e `aws iam delete-ssh-public-key`
5. Eliminare le credenziali Git dell'utente.

`aws iam list-service-specific-credentials` (per elencare le credenziali git dell'utente) e `aws iam delete-service-specific-credential`

6. Disattivare il dispositivo Multi-Factor Authentication (MFA), se uno è disponibile.

   `aws iam list-mfa-devices` (per elencare i dispositivi MFA dell'utente), `aws iam deactivate-mfa-device` (per disattivare il dispositivo) e `aws iam delete-virtual-mfa-device` (per eliminare definitivamente un dispositivo MFA virtuale)

7. Eliminare le policy inline dell'utente.

   `aws iam list-user-policies` (per elencare le policy inline per l'utente) e `aws iam delete-user-policy` (per eliminare la policy)

8. Scollegare le policy gestite collegate all'utente.

   `aws iam list-attached-user-policies` (per elencare le policy gestite collegate all'utente) e `aws iam detach-user-policy` (per scollegare la policy)

9. Rimuovere l'utente da qualsiasi gruppo IAM.

   `aws iam list-groups-for-user` (per elencare i gruppi IAM a cui appartiene l'utente) e `aws iam remove-user-from-group`

10. Eliminare l'utente.

    `aws iam delete-user`

Disattivazione di un utente IAM

Potrebbe essere necessario disattivare un utente IAM mentre è temporaneamente lontano dall'azienda. Puoi bloccare l'accesso ad AWS pur lasciando attive le sue credenziali di utente IAM.

Per disattivare un utente, crea e collega una policy per negare all'utente l'accesso a AWS. Puoi ripristinare l'accesso dell'utente in un secondo momento.

Di seguito sono riportati due esempi di policy di diniego che puoi collegare a un utente per negargli l'accesso.

La seguente policy non include un limite di tempo. È necessario rimuovere la policy per ripristinare l'accesso dell'utente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*"
      }
   ]
}
```

La seguente policy include una condizione che avvia la policy il 24 dicembre 2024 alle 23:59 (UTC) e la termina il 28 febbraio 2025 alle 23:59 (UTC).

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
          "DateGreaterThan": {"aws:CurrentTime": "2024-12-24T23:59:59Z"},
          "DateLessThan": {"aws:CurrentTime": "2025-02-28T23:59:59Z"}
          }
       }
   ]
}
```

Controlla l'accesso degli utenti IAM a Console di gestione AWS

Gli utenti IAM con autorizzazione che accedono al tuo account Account AWS tramite il Console di gestione AWS possono accedere alle tue risorse. AWS L'elenco seguente mostra i modi in cui puoi

concedere agli utenti IAM l'accesso alle tue Account AWS risorse tramite Console di gestione AWS. Mostra anche come gli utenti IAM possono accedere ad altre funzionalità AWS dell'account tramite il AWS sito web.

> **Note**
>
> L'uso di IAM non comporta alcun costo.

Il Console di gestione AWS

È possibile creare una password per ciascun utente IAM che deve accedere alla Console di gestione AWS. Gli utenti accedono alla console tramite la pagina di Account AWS accesso abilitata a IAM. Per informazioni su come visualizzare la pagina di accesso, consulta Come accedere ad AWS nella Guida per l'utente di Accedi ad AWS . Per informazioni sulla creazione di password , consulta Password utente in AWS.

Puoi impedire a un utente IAM di accedere a Console di gestione AWS rimuovendo la sua password. Ciò impedisce loro di accedere Console di gestione AWS utilizzando le proprie credenziali di accesso. Non modifica le autorizzazioni né impedisce l'accesso alla console utilizzando un ruolo assunto. Se l'utente dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l' AWS CLI accesso tramite Tools for Windows PowerShell, AWS API o AWS Console Mobile Application.

AWS Le tue risorse, come istanze Amazon EC2, bucket Amazon S3 e così via

Anche se gli utenti IAM dispongono di password, hanno ancora bisogno dell'autorizzazione per accedere alle risorse AWS . Quando crei un utente IAM, questo non dispone di autorizzazioni per impostazione predefinita. Per assegnare agli utenti IAM le autorizzazioni necessarie, puoi associare loro delle policy. Se disponi di molti utenti IAM che eseguiranno le stesse attività con le stesse risorse, puoi assegnare tali utenti IAM a un gruppo. Quindi assegna le autorizzazioni a tale gruppo. Per informazioni sulla creazione di utenti e gruppi IAM, vedere Identità IAM. Per ulteriori informazioni sull'utilizzo di policy per impostare le autorizzazioni, vedere Gestione degli accessi per AWS risorse.

AWS Forum di discussione

Chiunque può leggere i post sui forum di discussione AWS. Gli utenti che desiderano pubblicare domande o commenti sul Forum di AWS discussione possono farlo utilizzando il proprio nome utente. La prima volta che un utente pubblica un post sul Forum di AWS discussione, gli viene

richiesto di inserire un soprannome e un indirizzo e-mail. Solo quell'utente può usare quel soprannome nei forum di AWS discussione.

Le tue informazioni Account AWS di fatturazione e utilizzo

Puoi consentire agli utenti di accedere alle tue informazioni di Account AWS fatturazione e utilizzo. Per ulteriori informazioni, consulta Controllo dell'accesso alle informazioni di fatturazione nella Guida per l'utente di AWS Billing .

Le informazioni del tuo Account AWS profilo

Gli utenti non possono accedere alle informazioni Account AWS del tuo profilo.

Le tue credenziali Account AWS di sicurezza

Gli utenti non possono accedere alle tue credenziali Account AWS di sicurezza.

> **Note**
>
> Le policy IAM controllano l'accesso indipendentemente dall'interfaccia. Ad esempio, è possibile fornire a un utente la password per accedere alla Console di gestione AWS. Le policy per tale utente (o per qualsiasi gruppo cui l'utente appartiene) determina ciò che l'utente può fare nella Console di gestione AWS. In alternativa, puoi fornire all'utente le chiavi di AWS accesso a cui effettuare AWS chiamate API. In questo caso, le policy controllano le operazioni che l'utente può richiamare tramite una libreria o un client che utilizza le chiavi di accesso per l'autenticazione.

Modificare le autorizzazioni per un utente IAM

Puoi modificare le autorizzazioni per un utente IAM del tuo paese Account AWS modificando l'appartenenza ai gruppi, copiando le autorizzazioni da un utente esistente, allegando le policy direttamente a un utente o impostando un limite di autorizzazioni. Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un utente. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

Per informazioni sulle autorizzazioni necessarie per modificare le autorizzazioni per un utente, consulta la pagina Autorizzazioni necessarie per accedere alle risorse IAM.

Argomenti

- Visualizzazione dell'accesso utente

- Generazione di una policy basata sull'attività di accesso di un utente
- Aggiunta di autorizzazioni a un utente (console)
- Modifica delle autorizzazioni per un utente (console)
- Per rimuovere una policy delle autorizzazioni da un utente (console)
- Per rimuovere il limite delle autorizzazioni da un utente (console)
- Aggiungere e rimuovere le autorizzazioni (AWS CLI o AWS API) di un utente

Visualizzazione dell'accesso utente

Prima di modificare le autorizzazioni per un utente, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Generazione di una policy basata sull'attività di accesso di un utente

Talvolta, è possibile concedere autorizzazioni a un'entità IAM (utente o ruolo) oltre a quelle richieste. Per ottimizzare le autorizzazioni concesse, puoi generare una policy IAM basata sull'attività di accesso per un'entità. IAM Access Analyzer esamina AWS CloudTrail i log e genera un modello di policy che contiene le autorizzazioni utilizzate dall'entità nell'intervallo di date specificato. È possibile utilizzare il modello per creare una policy gestita con autorizzazioni granulari e quindi collegarla al ruolo IAM. In questo modo, concedi solo le autorizzazioni necessarie all'utente o al ruolo per interagire con le AWS risorse per il tuo caso d'uso specifico. Per ulteriori informazioni, consulta Generazione di policy per Sistema di analisi degli accessi IAM.

Aggiunta di autorizzazioni a un utente (console)

Per aggiungere policy di autorizzazione a un utente, IAM offre tre diverse possibilità:

- Aggiungi l'utente IAM a un gruppo IAM: rende l'utente membro di un gruppo. Le policy del gruppo vengono collegate all'utente.
- Copia le autorizzazioni da un utente IAM esistente: copia tutte le appartenenze ai gruppi, le policy gestite collegate, le policy in linea e tutti i limiti delle autorizzazioni esistenti dall'utente di origine.
- Collega direttamente le policy all'utente IAM: collega una policy gestita direttamente all'utente. Per una gestione più semplice delle autorizzazioni, collega le policy a un gruppo e rendi quindi gli utenti IAM membri dei gruppi appropriati.

> **Important**
>
> Se l'utente ha un limite delle autorizzazioni, non sarà possibile aggiungere più autorizzazioni di quante ne consenta il limite delle autorizzazioni.

Per aggiungere le autorizzazioni aggiungendo l'utente IAM a un gruppo

L'aggiunta di un utente IAM a un gruppo IAM aggiorna immediatamente le autorizzazioni dell'utente con le autorizzazioni definite per il gruppo.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Nell'elenco degli Utenti, seleziona il nome dell'utente IAM.
5. Seleziona la scheda Gruppi per visualizzare l'elenco dei gruppi IAM che includono l'utente corrente.
6. Scegli Aggiungi utente ai gruppi.
7. Seleziona la casella di controllo per ciascun gruppo in cui si desidera includere l'utente. L'elenco mostra il nome di ciascun gruppo e le policy che l'utente riceve se diventa un membro di tale gruppo.
8. (Facoltativo) Puoi scegliere Crea gruppo per definire un nuovo gruppo. Ciò è utile se desideri aggiungere l'utente a un gruppo con policy collegate diverse rispetto ai gruppi esistenti:

    a. Nella nuova scheda, per User group name (Nome gruppo di utenti), digita un nome per il nuovo gruppo.

    > **Note**
    >
    > Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote. I nomi dei gruppi possono essere una combinazione di un massimo di 128 lettere, cifre e dei seguenti caratteri: più (+), uguale (=), virgola (,), punto (.), chiocciola (@) e trattino (-).

I nomi devono essere univoci all'interno di un account. Non si distinguono per caso. Ad esempio, non è possibile creare due gruppi chiamati TESTGROUP e testgroup.

b. Seleziona una o più caselle di controllo per le policy gestite da collegare al gruppo. È inoltre possibile creare una nuova policy gestita selezionando Create policy (Crea policy). In questo caso, tornare a questa scheda o finestra del browser quando la nuova policy è stata completata, selezionare Refresh (Aggiorna) e quindi selezionare la nuova policy da collegare al gruppo. Per ulteriori informazioni, consulta Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente.

c. Scegli Create user group (Crea gruppo di utenti).

d. Tornare alla scheda originale e aggiornare l'elenco di gruppi. Quindi seleziona la casella di controllo del nuovo gruppo.

9. Scegli Aggiungi utente ai gruppi.

La console visualizza un messaggio di stato che informa che l'utente è stato aggiunto ai gruppi specificati.

Per aggiungere le autorizzazioni tramite copia da un altro utente IAM

Se scegli di aggiungere autorizzazioni a un utente IAM copiando le autorizzazioni, IAM copia tutte le appartenenze ai gruppi, le policy gestite collegate, le policy in linea e tutti i limiti delle autorizzazioni esistenti dall'utente specificato e li applica immediatamente all'utente correntemente selezionato.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Nell'elenco degli Utenti, seleziona il nome dell'utente IAM.
5. Nella scheda Autorizzazioni, scegli Aggiungi autorizzazioni.
6. Nella pagina Aggiungi autorizzazioni, scegli Copia autorizzazioni. L'elenco mostra gli utenti IAM disponibili con le relative appartenenze ai gruppi e le policy collegate.

7. Selezionare il pulsante di opzione accanto all'utente che dispone delle autorizzazioni che si desidera copiare.
8. Seleziona Next (Successivo) per visualizzare l'elenco delle modifiche che devono essere apportate all'utente. Selezionare quindi Add permissions (Aggiungi autorizzazioni).

La console visualizza un messaggio di stato che ti informa che le autorizzazioni sono state copiate dall'utente IAM che hai specificato.

Per aggiungere le autorizzazioni collegando le policy direttamente all'utente IAM

Puoi collegare una policy gestita direttamente a un utente IAM. Le autorizzazioni aggiornate vengono applicate immediatamente.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Nell'elenco degli Utenti, seleziona il nome dell'utente IAM.
5. Nella scheda Autorizzazioni, scegli Aggiungi autorizzazioni.
6. Nella pagina Aggiungi autorizzazioni, seleziona Collega direttamente le policy. L'elenco Policy di autorizzazione mostra le policy disponibili insieme ai relativi tipi di policy e alle entità collegate.
7. Seleziona il pulsante di opzione accanto al nome della policy che desideri collegare.
8. Seleziona Next (Successivo) per visualizzare l'elenco delle modifiche che devono essere apportate all'utente. Selezionare quindi Add permissions (Aggiungi autorizzazioni).

La console visualizza un messaggio di stato che informa che la policy è stata aggiunta all'utente IAM specificato.

Per impostare il limite delle autorizzazioni per un utente IAM

Un limite di autorizzazioni è una funzionalità avanzata per la gestione delle autorizzazioni AWS che viene utilizzata per impostare le autorizzazioni massime che un utente IAM può avere. L'impostazione di un limite delle autorizzazioni limita immediatamente le autorizzazioni degli utenti IAM al limite, indipendentemente dalle altre autorizzazioni concesse.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Nell'elenco Utenti, scegli il nome dell'utente IAM di cui si desidera modificare il limite delle autorizzazioni.
5. Scegli la scheda Autorizzazioni. Se necessario, apri la sezione Permissions boundary (Limite delle autorizzazioni), quindi seleziona Set permissions boundary (Imposta limite delle autorizzazioni).
6. Nella pagina Imposta il limite delle autorizzazioni, in Policy delle autorizzazioni, seleziona la policy da utilizzare per il limite delle autorizzazioni.
7. Scegliere Set boundary (Imposta limite).

   La console visualizza un messaggio di stato che informa che il limite delle autorizzazioni è stato aggiunto.

Modifica delle autorizzazioni per un utente (console)

IAM consente di modificare le autorizzazioni associate a un utente nei modi seguenti:

- Modifica una policy di autorizzazione: è possibile modificare la policy in linea di un utente, la policy in linea del gruppo dell'utente o la policy gestita collegato all'utente direttamente o da un gruppo. Se l'utente ha un limite delle autorizzazioni, non è possibile fornire più autorizzazioni di quante ne consenta la policy utilizzata come limite delle autorizzazioni dell'utente.

- Modifica del limite delle autorizzazioni: modifica la policy utilizzata come limite delle autorizzazioni per l'utente. In questo modo è possibile ampliare o limitare il numero massimo di autorizzazioni che un utente può avere.

Modifica di una policy di autorizzazione collegata a un utente

La modifica delle autorizzazioni aggiorna immediatamente l'accesso dell'utente.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Nell'elenco Utenti, scegli il nome dell'utente IAM di cui si desidera modificare il limite delle autorizzazioni.
5. Scegli la scheda Autorizzazioni. Se necessario, apri la sezione Limite delle autorizzazioni.
6. Selezionare il nome della policy da modificare per visualizzare i relativi dettagli. Seleziona la scheda Entità collegate per visualizzare le altre entità (utenti, gruppi e ruoli IAM) che potrebbero essere interessate dalla modifica della policy.
7. Selezionare la scheda Permissions (Autorizzazioni) e rivedere le autorizzazioni concesse dalla policy. Per apportare modifiche alle autorizzazioni, scegli Modifica.
8. Modifica la policy e risolvi eventuali suggerimenti di convalida della policy. Per ulteriori informazioni, consulta Modificare le policy IAM.
9. Seleziona Successivo, esamina il riepilogo della policy, quindi scegli Salva le modifiche.

   La console visualizza un messaggio di stato che informa che la policy è stata aggiornata.

Per modificare il limite delle autorizzazioni per un utente

La modifica del limite delle autorizzazioni aggiorna immediatamente l'accesso dell'utente.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Nell'elenco Utenti, scegli il nome dell'utente IAM di cui si desidera modificare il limite delle autorizzazioni.
5. Scegli la scheda Autorizzazioni. Se necessario, aprire la sezione Permissions boundary (Limite delle autorizzazioni) e selezionare Change boundary (Modifica limite).
6. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
7. Scegliere Set boundary (Imposta limite).

La console visualizza un messaggio di stato che informa che il limite delle autorizzazioni è stato modificato.

Per rimuovere una policy delle autorizzazioni da un utente (console)

La rimozione di una policy di autorizzazioni aggiorna immediatamente l'accesso dell'utente.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Selezionare il nome dell'utente per cui rimuovere le policy di autorizzazioni.
5. Scegli la scheda Autorizzazioni.
6. Se desideri rimuovere le autorizzazioni rimuovendo una policy esistente, visualizza la colonna Collegato tramite per comprendere il modo in cui l'utente riceve la policy prima di selezionare Rimuovi per rimuoverla:

- Se la policy è applicata tramite l'appartenenza a un gruppo, seleziona X per rimuovere l'utente dal gruppo. Ricordare che potrebbero essere presenti più policy associate a un singolo gruppo. Se si rimuove un utente da un gruppo, l'utente perde l'accesso a tutte le policy che ricevute tramite l'appartenenza a tale gruppo.
- Se la policy è una policy gestita collegata direttamente all'utente, scegliendo Remove (Rimuovi) questa sarà scollegata dall'utente. Ciò non influisce sulla policy stessa o su qualsiasi altra entità a cui la policy potrebbe essere collegata.
- Se la policy è una policy in linea integrata, scegliendo Rimuovi la policy verrà rimossa da IAM. Le policy inline collegate direttamente a un utente sono presenti solo in tale utente.

Se la policy è stata concessa all'utente tramite l'appartenenza a un gruppo, la console visualizza un messaggio di stato che informa che l'utente IAM è stato rimosso dal gruppo IAM. Se la policy è collegata direttamente o in linea, il messaggio di stato informa che la policy è stata rimossa.

Per rimuovere il limite delle autorizzazioni da un utente (console)

La rimozione del limite delle autorizzazioni aggiorna immediatamente l'accesso dell'utente.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Nell'elenco Utenti, scegli il nome dell'utente IAM di cui si desidera rimuovere il limite delle autorizzazioni.
5. Scegli la scheda Autorizzazioni. Se necessario, apri la sezione Limite delle autorizzazioni.
6. Selezionare Change boundary (Modifica limite). Nella finestra di dialogo di conferma, scegli Rimuovi limite per confermare la rimozione del limite delle autorizzazioni.

La console visualizza un messaggio di stato che informa che il limite delle autorizzazioni è stato rimosso.

Aggiungere e rimuovere le autorizzazioni (AWS CLI o AWS API) di un utente

Per aggiungere o rimuovere le autorizzazioni a livello di codice, è necessario aggiungere o rimuovere i gruppi di appartenenza, collegare o distaccare le appartenenze ai gruppi, collegare o distaccare le policy gestite oppure aggiungere o eliminare le policy inline. Per ulteriori informazioni, consulta i seguenti argomenti:

- Modificare gli utenti nei gruppi IAM
- Aggiunta e rimozione di autorizzazioni per identità IAM

Password utente in AWS

Puoi gestire le password per gli utenti IAM del tuo account. Gli utenti IAM necessitano di password per accedere a. Console di gestione AWS Gli utenti non hanno bisogno di password per accedere alle AWS risorse in modo programmatico utilizzando Tools for Windows AWS CLI PowerShell, o. AWS SDKs APIs Per questi ambienti, hai la possibilità di assegnare chiavi di accesso agli utenti IAM. Tuttavia, ci sono altre alternative più sicure delle chiavi di accesso che ti consigliamo di prendere in considerazione per prime. Per ulteriori informazioni, consulta AWS credenziali di sicurezza.

> **Note**
>
> Se uno dei tuoi utenti IAM perde o dimentica la password, non puoi recuperarla da IAM. A seconda delle impostazioni, l'utente o l'amministratore devono creare una nuova password.

Indice

- Configurare una policy delle password di un account per gli utenti IAM
- Gestione delle password per gli utenti IAM
- Consentire agli utenti IAM di cambiare le loro password
- Come un utente IAM può modificare la propria password

Configurare una policy delle password di un account per gli utenti IAM

Puoi impostare una politica di password personalizzata Account AWS per specificare i requisiti di complessità e i periodi di rotazione obbligatori per le password degli utenti IAM. Se non imposti una

politica di password personalizzata, le password utente IAM devono soddisfare la politica di password predefinita AWS . Per ulteriori informazioni, consulta Opzioni di policy delle password personalizzata.

Argomenti

- Impostazione di una policy delle password
- Autorizzazioni necessarie per impostare una policy delle password
- Policy delle password predefinita
- Opzioni di policy delle password personalizzata
- Per impostare una policy delle password (console)
- Per modificare una policy delle password (console)
- Eliminazione di una policy delle password personalizzata (console)
- Impostazione di una policy sulle password (AWS CLI)
- Impostazione di una politica in materia di password (AWS API)

Impostazione di una policy delle password

La policy sulle password IAM non si applica alla Utente root dell'account AWS password o alle chiavi di accesso utente IAM. Se una password scade, l'utente IAM non può accedere Console di gestione AWS ma può continuare a utilizzare le proprie chiavi di accesso.

Quando si crea o si modifica una policy sulle password, la maggior parte delle impostazioni sulla policy delle password vengono applicate la prossima volta che gli utenti modificano le password. Tuttavia, alcune delle impostazioni vengono applicate immediatamente. Ad esempio:

- Quando si impostano i requisiti minimi di lunghezza e del tipo di caratteri, le impostazioni vengono applicate la volta successiva che gli utenti cambiano le password. Gli utenti non sono costretti a modificare le proprie password esistenti, anche se le password esistenti non rispettano i criteri della policy aggiornata sulle password.
- Quando si imposta un periodo di scadenza della password, il periodo di scadenza viene applicato immediatamente. Ad esempio, un periodo di scadenza della password viene impostato a 90 giorni. In tal caso, la password scade per tutti gli utenti IAM la cui password attuale è più vecchia di 90 giorni. Gli utenti sono tenuti a modificare la propria password all'accesso successivo.

Non è possibile creare una "policy di esclusione" per bloccare un utente fuori dall'account dopo un numero specificato di tentativi di accesso non riusciti. Per una maggiore sicurezza, si consiglia di

combinare una policy delle password complessa con l'autenticazione Multi-Factor Authentication (MFA). Per ulteriori informazioni sulla funzionalità MFA, consultare AWS Multi-factor autenticazione in IAM.

Autorizzazioni necessarie per impostare una policy delle password

È necessario configurare le autorizzazioni per consentire a un'entità IAM (utente o ruolo) di visualizzare o modificare la policy delle password dell'account. È possibile includere le seguenti operazioni di policy della password in una policy IAM:

- `iam:GetAccountPasswordPolicy`: consente all'entità di visualizzare la policy delle password per il proprio account
- `iam:DeleteAccountPasswordPolicy`: consente all'entità di eliminare la policy delle password personalizzata per il proprio account e ripristinare la policy delle password di default
- `iam:UpdateAccountPasswordPolicy`: consente all'entità di creare o modificare la policy delle password personalizzata per il proprio account

La policy seguente consente l'accesso completo per visualizzare e modificare la policy delle password dell'account. Per ulteriori informazioni su come creare una policy IAM usando il documento di policy JSON di esempio, consulta the section called "Creazione di policy utilizzando l'editor JSON".

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccessPasswordPolicy",
            "Effect": "Allow",
            "Action": [
                "iam:GetAccountPasswordPolicy",
                "iam:DeleteAccountPasswordPolicy",
                "iam:UpdateAccountPasswordPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Per informazioni sulle autorizzazioni necessarie per modificare la password da parte di un utente IAM, consulta Consentire agli utenti IAM di cambiare le loro password.

Policy delle password predefinita

Se un amministratore non imposta una politica di password personalizzata, le password degli utenti IAM devono soddisfare la politica di AWS password predefinita.

La policy delle password predefinita applica le seguenti condizioni:

- Deve avere una lunghezza minima di 8 caratteri e massima di 128 caratteri
- Deve includere almeno tre dei seguenti tipi di caratteri: lettere maiuscole, lettere minuscole, numeri e caratteri non alfanumerici (`! @ # $ % ^ & * ( ) _ + - = [ ] { } | '`)
- Non essere identico al tuo Account AWS nome o indirizzo email
- Password che non scadono mai

Opzioni di policy delle password personalizzata

Quando si configura una policy delle password personalizzata per l'account, è possibile specificare le seguenti condizioni:

- Lunghezza minima della password: è possibile specificare un minimo di 6 caratteri e un massimo di 128 caratteri.
- Complessità della password: puoi selezionare una delle seguenti caselle di controllo per definire la complessità delle password dell'utente IAM:
  - Richiedi almeno una lettera maiuscola dall'alfabeto latino (A-Z)
  - Richiedi almeno una lettera minuscola dall'alfabeto latino (a-z)
  - Richiedere almeno un numero
  - Richiedere almeno un carattere non alfanumerico `! @ # $ % ^ & * ( ) _ + - = [ ] { } | '`
- Turn on password expiration (Abilita scadenza della password): puoi selezionare e specificare un minimo di 1 e un massimo di 1.095 giorni di validità delle password utente IAM dopo che sono state impostate. Ad esempio, se specifichi una scadenza di 90 giorni, ciò influisce immediatamente su tutti gli utenti. Dopo la modifica, gli utenti con password impostata da oltre 90 giorni dovranno impostarne una nuova quando accedono alla console. Gli utenti con password vecchie di 75-89 giorni ricevono un Console di gestione AWS avviso sulla scadenza della password. Gli utenti IAM

possono modificare la password in qualsiasi momento se dispongono dell'autorizzazione. Quando impostano una nuova password, il periodo di scadenza per tale password ricomincia da capo. Un utente IAM può avere solo una password valida alla volta.

- La scadenza della password richiede la reimpostazione dell'amministratore: seleziona questa opzione per impedire agli utenti IAM di utilizzare il Console di gestione AWS per aggiornare le proprie password dopo la scadenza della password. Prima di selezionare questa opzione, verifica che nell' Account AWS sia presente più di un utente con le autorizzazioni amministrative per ripristinare le password degli utenti IAM. Gli amministratori che dispongono dell'autorizzazione `iam:UpdateLoginProfile` possono reimpostare le password degli utenti IAM. Gli utenti IAM che dispongono dell'autorizzazione `iam:ChangePassword` e di chiavi di accesso attive possono reimpostare autonomamente la propria password della console utente IAM a livello di programmazione. Se si deseleziona questa casella di controllo, gli utenti IAM con password scadute devono comunque impostare una nuova password prima di poter accedere alla Console di gestione AWS.
- Allow users to change their own password (Consenti agli utenti di modificare la propria password): puoi consentire a tutti gli utenti IAM nel tuo account di modificare autonomamente le proprie password. Ciò consente agli utenti di accedere all'operazione `iam:ChangePassword` solo per il proprio utente e all'operazione `iam:GetAccountPasswordPolicy`. Questa opzione non associa una policy di autorizzazione a ciascun utente. Piuttosto, IAM applica le autorizzazioni a livello di account per tutti gli utenti. In alternativa, è possibile consentire solo ad alcuni utenti di gestire in autonomia le proprie password. A tale scopo, deseleziona questa casella di controllo. Per ulteriori informazioni sull'utilizzo di policy per limitare chi può gestire le password, consultare Consentire agli utenti IAM di cambiare le loro password.
- Impedisci il riutilizzo di una password: puoi impedire che gli utenti IAM riutilizzino un determinato numero di password precedenti. È possibile specificare un numero minimo di 1 e un numero massimo di 24 password precedenti che non possono essere ripetute.

Per impostare una policy delle password (console)

Puoi utilizzare la Console di gestione AWS per creare, modificare o eliminare una politica di password personalizzata. Le modifiche alla policy delle password si applicano ai nuovi utenti IAM creati dopo questa modifica della policy e agli utenti IAM esistenti quando questi modificano le proprie password.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.

2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
4. Nella sezione Password policy (Policy delle password), scegli Edit (Modifica).
5. Scegli Custom (Personalizzato) per utilizzare una policy di password personalizzata.
6. Seleziona le opzioni che desideri applicare alla policy delle password e scegli Salva modifiche.
7. Conferma che desideri impostare la policy delle password personalizzata scegliendo Set custom (Imposta personalizzata).

La console visualizza un messaggio di stato che informa che i requisiti di password per gli utenti IAM sono stati aggiornati.

Per modificare una policy delle password (console)

È possibile utilizzare la Console di gestione AWS per creare, modificare o eliminare una politica personalizzata in materia di password. Le modifiche alla policy delle password si applicano ai nuovi utenti IAM creati dopo questa modifica della policy e agli utenti IAM esistenti quando questi modificano le proprie password.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
4. Nella sezione Password policy (Policy delle password), scegli Edit (Modifica).
5. Seleziona le opzioni che desideri applicare alla policy delle password e scegli Salva modifiche.
6. Conferma che desideri impostare la policy delle password personalizzata scegliendo Set custom (Imposta personalizzata).

La console visualizza un messaggio di stato che informa che i requisiti di password per gli utenti IAM sono stati aggiornati.

Eliminazione di una policy delle password personalizzata (console)

È possibile utilizzare la Console di gestione AWS per creare, modificare o eliminare una politica personalizzata in materia di password. Le modifiche alla policy delle password si applicano ai nuovi utenti IAM creati dopo questa modifica della policy e agli utenti IAM esistenti quando questi modificano le proprie password.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
4. Nella sezione Password policy (Policy delle password), scegli Edit (Modifica).
5. Scegli IAM default (Predefinito IAM) per eliminare la policy di password personalizzata, quindi scegli Save changes (Salva modifiche).
6. Conferma che desideri impostare la policy delle password predefinita IAM scegliendo Set default (Imposta predefinita).

   La console visualizza un messaggio di stato che informa che la policy delle password è impostata sul valore predefinito di IAM.

Impostazione di una policy sulle password (AWS CLI)

È possibile utilizzare il AWS Command Line Interface per impostare una politica in materia di password.

Per gestire la politica personalizzata in materia di password dell'account dal AWS CLI

Esegui i comandi seguenti:

- Per creare o modificare la policy delle password personalizzata: `aws iam update-account-password-policy`
- Per visualizzare la policy delle password: `aws iam get-account-password-policy`
- Per eliminare la policy delle password personalizzata: `aws iam delete-account-password-policy`

Impostazione di una politica in materia di password (AWS API)

È possibile utilizzare le operazioni AWS API per impostare una politica in materia di password.

Per gestire la politica personalizzata in materia di password dell'account dall' AWS API

Chiamare le operazioni seguenti:

- Per creare o modificare la policy delle password personalizzata: `UpdateAccountPasswordPolicy`
- Per visualizzare la policy delle password: `GetAccountPasswordPolicy`
- Per eliminare la policy delle password personalizzata: `DeleteAccountPasswordPolicy`

Gestione delle password per gli utenti IAM

Gli utenti IAM che utilizzano il Console di gestione AWS per lavorare con AWS le risorse devono disporre di una password per accedere. Puoi creare, modificare o eliminare una password di un utente IAM nel tuo account AWS .

Dopo aver assegnato una password a un utente, l'utente può accedere Console di gestione AWS utilizzando l'URL di accesso del tuo account, che ha il seguente aspetto:

```
https://12-digit-AWS-account-ID or alias.signin.aws.amazon.com/console
```

Per ulteriori informazioni su come gli utenti IAM accedono a Console di gestione AWS, consulta Come accedere alla AWS Guida per l'Accedi ad AWS utente.

Anche se gli utenti hanno le proprie password, hanno ancora bisogno delle autorizzazioni per accedere alle tue risorse AWS . Per impostazione predefinita, un utente non ha autorizzazioni. Per fornire agli utenti le autorizzazioni di cui hanno bisogno, assegna loro le policy o ai gruppi di appartenenza. Per ulteriori informazioni sulla creazione di utenti e gruppi, vedere Identità IAM. Per ulteriori informazioni sull'utilizzo di policy per impostare le autorizzazioni, vedere Modificare le autorizzazioni per un utente IAM.

Puoi concedere le autorizzazioni agli utenti per modificare le loro password. Per ulteriori informazioni, consulta Consentire agli utenti IAM di cambiare le loro password. Per informazioni su come gli utenti accedono alla pagina di accesso del tuo account, consulta Come accedere ad AWS nella Guida per l'utente di Accedi ad AWS .

Argomenti

- Creazione, modifica o eliminazione di una password dell'utente IAM (console)

Creazione, modifica o eliminazione di una password dell'utente IAM (console)

Puoi utilizzare il Console di gestione AWS per gestire le password dei tuoi utenti IAM.

Le esigenze di accesso degli utenti possono cambiare nel tempo. Potrebbe essere necessario consentire a un utente destinato all'accesso alla CLI di accedere alla console, modificare la password di un utente perché riceve l'e-mail con le proprie credenziali o eliminare un utente quando lascia l'organizzazione o non ha più bisogno AWS di accedervi.

Per creare una password per l'utente IAM (console)

Utilizza questa procedura per consentire a un utente di accedere alla console creando una password associata al nome utente.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere a nella Guida per AWS l'AWS Sign-In utente.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Seleziona il nome dell'utente sul quale desideri creare la password.
5. Scegli la scheda Security credentials (Credenziali di sicurezza), quindi in Console sign-in (Accesso alla console), scegli Enable console access (Abilita l'accesso alla console).
6. Nella finestra di dialogo Consenti accesso alla console, seleziona Reimposta password, quindi scegli se IAM deve generare una password o se deve essere creata una password personalizzata:

   - Per fare in modo che IAM generi una password, scegli Password autogenerata
   - Per creare una password personalizzata, scegliere Custom password (Password personalizzata) e digitare la password.

   > **Note**
   >
   > La password creata deve essere conforme alla policy delle password dell'account.

7. Per richiedere all'utente di creare una nuova password all'accesso, scegli L'utente deve creare una nuova password al prossimo accesso.
8. Per richiedere all'utente di utilizzare immediatamente la nuova password, seleziona Revoca sessioni di console attive. Ciò collega una policy in linea all'utente IAM che nega all'utente l'accesso alle risorse se le sue credenziali sono più vecchie del periodo specificato dalla policy.
9. Scegliere Reimposta password
10. La finestra di dialogo Password della console ti informa che è stata abilitata la nuova password dell'utente. Per visualizzare la password in modo da poterla condividere con l'utente, scegli Mostra nella finestra di dialogo Password della console. Seleziona Scarica il file .csv per scaricare un file con le credenziali dell'utente.

> **Important**
>
> Per motivi di sicurezza, non è possibile accedere alla password dopo aver completato questa fase, ma è possibile creare una nuova password in qualsiasi momento.

La console visualizza un messaggio di stato che informa che l'accesso alla console è stato abilitato.

Come cambiare la password per un utente IAM (console)

Utilizza questa procedura per aggiornare una password associata al nome utente.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere a nella Guida per AWS l'AWS Sign-In utente.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Seleziona il nome dell'utente di cui desideri modificare la password.
5. Scegli la scheda Security credentials (Credenziali di sicurezza), quindi in Console sign-in (Accesso alla console), scegli Manage console access (Gestisci l'accesso alla console).

6. Nella finestra di dialogo Gestisci l'accesso alla console, seleziona Reimposta password, quindi scegli se la password deve essere generata da IAM o se deve essere creata una password personalizzata:

   - Per fare in modo che IAM generi una password, scegli Password autogenerata
   - Per creare una password personalizzata, scegliere Custom password (Password personalizzata) e digitare la password.

   > **Note**
   >
   > La password creata deve essere conforme alla policy delle password dell'account.

7. Per richiedere all'utente di creare una nuova password all'accesso, scegli L'utente deve creare una nuova password al prossimo accesso.
8. Per richiedere all'utente di utilizzare immediatamente la nuova password, seleziona Revoca sessioni di console attive. Ciò collega una policy in linea all'utente IAM che nega all'utente l'accesso alle risorse se le sue credenziali sono più vecchie del periodo specificato dalla policy.
9. Scegliere Reimposta password
10. La finestra di dialogo Password della console ti informa che è stata abilitata la nuova password dell'utente. Per visualizzare la password in modo da poterla condividere con l'utente, scegli Mostra nella finestra di dialogo Password della console. Seleziona Scarica il file .csv per scaricare un file con le credenziali dell'utente.

    > **Important**
    >
    > Per motivi di sicurezza, non è possibile accedere alla password dopo aver completato questa fase, ma è possibile creare una nuova password in qualsiasi momento.

La console visualizza un messaggio di stato che informa che l'accesso alla console è stato aggiornato.

Come eliminare (disabilitare) una password dell'utente IAM (console)

Utilizza questa procedura per eliminare una password associata al nome utente rimuovendo l'accesso alla console per l'utente.

> **Important**
>
> Puoi impedire a un utente IAM di accedere a Console di gestione AWS rimuovendo la sua password. Ciò impedisce loro di accedere Console di gestione AWS utilizzando le proprie credenziali di accesso. Non modifica le autorizzazioni né impedisce l'accesso alla console utilizzando un ruolo assunto. Se l'utente dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l'accesso tramite l' AWS CLI AWS API Tools for Windows PowerShell o l'applicazione mobile AWS Console.

Console

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere a nella Guida per AWS l'AWS Sign-In utente.
2. Nella home page della console IAM, nel pannello di navigazione a sinistra, immetti la query nella casella di testo Cerca IAM.
3. Nel pannello di navigazione, seleziona Utenti.
4. Seleziona il nome dell'utente di cui desideri eliminare la password.
5. Scegli la scheda Security credentials (Credenziali di sicurezza), quindi in Console sign-in (Accesso alla console), scegli Manage console access (Gestisci l'accesso alla console).
6. Per richiedere all'utente di interrompere immediatamente l'uso della console, seleziona Revoca sessioni di console attive. Ciò collega una policy in linea all'utente IAM che nega all'utente l'accesso alle risorse se le sue credenziali sono più vecchie del periodo specificato dalla policy.
7. Scegli Disabilita accesso

   La console visualizza un messaggio di stato che informa che l'accesso alla console è stato disabilitato.

Creazione, modifica o eliminazione di una password utente IAM (AWS CLI)

Puoi usare il AWS CLI per gestire le password dei tuoi utenti IAM.

Per creare una password (AWS CLI)

1. (Facoltativo) Per determinare se un utente dispone di una password, eseguire questo comando: aws iam get-login-profile
2. Per creare una password, eseguire questo comando: aws iam create-login-profile

Per modificare la password di un utente (AWS CLI)

1. (Facoltativo) Per determinare se un utente dispone di una password, eseguire questo comando: aws iam get-login-profile
2. Per modificare una password, eseguire questo comando: aws iam update-login-profile

Per eliminare (disabilitare) la password di un utente (AWS CLI)

1. (Facoltativo) Per determinare se un utente dispone di una password, eseguire questo comando: aws iam get-login-profile
2. (Facoltativo) Per determinare quando una password è stata utilizzata per l'ultima volta, eseguire questo comando: aws iam get-user
3. Per eliminare una password, eseguire questo comando: aws iam delete-login-profile

> **Important**
>
> Quando elimini una password dell'utente, l'utente non può più accedere alla Console di gestione AWS. Se l'utente dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l'accesso tramite AWS CLI le chiamate di funzione Tools for Windows PowerShell o AWS API. Quando utilizzi Tools for Windows PowerShell o AWS API per eliminare un utente dal tuo Account AWS, devi prima eliminare la password utilizzando questa operazione. AWS CLI Per ulteriori informazioni, consulta Eliminazione di un utente IAM (AWS CLI).

Per revocare le sessioni attive della console di un utente prima di un orario specificato (AWS CLI)

1. Per incorporare una policy in linea che revochi le sessioni di console attive di un utente IAM prima di un orario specificato, usa la seguente policy in linea ed esegui questo comando: aws iam put-user-policy

Questa policy in linea nega tutte le autorizzazioni e include la chiave di condizione `era: TokenIssueTime`. Revoca le sessioni di console attive dell'utente prima del tempo specificato nell'elemento `Condition` della policy in linea. Sostituire il valore della chiave di condizione `aws:TokenIssueTime` con il proprio valore.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "DateLessThan": {
        "aws:TokenIssueTime": "2014-05-07T23:47:00Z"
      }
    }
  }
}
```

2. (Facoltativo) Per elencare i nomi delle policy in linea incorporate nell'utente IAM, esegui questo comando: aws iam list-user-policies
3. (Facoltativo) Per visualizzare la policy in linea specificata incorporata nell'utente IAM, esegui questo comando: aws iam list-user-policies

Creazione, modifica o eliminazione di una password utente IAM (AWS API)

Puoi utilizzare l' AWS API per gestire le password dei tuoi utenti IAM.

Per creare una password (AWS API)

1. (Facoltativo) Per determinare se un utente dispone di una password, chiama questa operazione: GetLoginProfile
2. Per creare una password, chiamate questa operazione: CreateLoginProfile

Per modificare la password di un utente (AWS API)

1. (Facoltativo) Per determinare se un utente dispone di una password, chiama questa operazione: GetLoginProfile
2. Per modificare una password, chiamate questa operazione: UpdateLoginProfile

Per eliminare (disabilitare) la password di un utente (AWS API)

1. (Facoltativo) Per determinare se un utente dispone di una password, esegui questo comando: GetLoginProfile
2. (Facoltativo) Per determinare l'ultima volta che una password è stata utilizzata, esegui questo comando: GetUser
3. Per eliminare una password, esegui questo comando: DeleteLoginProfile

> **Important**
>
> Quando elimini una password dell'utente, l'utente non può più accedere alla Console di gestione AWS. Se l'utente dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l'accesso tramite AWS CLI le chiamate di funzione Tools for Windows PowerShell o AWS API. Quando utilizzi Tools for Windows PowerShell o AWS API per eliminare un utente dal tuo Account AWS, devi prima eliminare la password utilizzando questa operazione. AWS CLI Per ulteriori informazioni, consulta Eliminazione di un utente IAM (AWS CLI).

Per revocare le sessioni attive della console di un utente prima di un orario specificato (AWS API)

1. Per incorporare una policy in linea che revoca le sessioni attive della console di un utente IAM prima di un orario specificato, utilizza la seguente policy in linea ed esegui questo comando: PutUserPolicy

   Questa policy in linea nega tutte le autorizzazioni e include la chiave di condizione `era: TokenIssueTime`. Revoca le sessioni di console attive dell'utente prima del tempo specificato nell'elemento `Condition` della policy in linea. Sostituire il valore della chiave di condizione `aws:TokenIssueTime` con il proprio valore.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "DateLessThan": {
        "aws:TokenIssueTime": "2014-05-07T23:47:00Z"
      }
    }
  }
}
```

2. (Facoltativo) Per elencare i nomi delle policy in linea incorporate nell'utente IAM, esegui questo comando: ListUserPolicies
3. (Facoltativo) Per visualizzare la policy in linea denominata incorporata nell'utente IAM, esegui questo comando: GetUserPolicy

Consentire agli utenti IAM di cambiare le loro password

> **Note**
>
> Per modificare le password, gli utenti con identità federate utilizzeranno il processo definito dal proprio gestore di identità. Come best practice, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee.

Puoi concedere agli utenti IAM l'autorizzazione per modificare le password di accesso alla Console di gestione AWS. Ci sono due modi per farlo:

- Consenti a tutti gli utenti IAM nell'account di cambiare le loro password.
- Consentire solo agli utenti IAM selezionati di cambiare le loro password. In questo scenario, è possibile disattivare l'opzione di modifica della password per tutti gli utenti e utilizzare una policy IAM per concedere autorizzazioni solo ad alcuni utenti. Questo approccio consente a questi utenti

di modificare le proprie password e, facoltativamente, altre credenziali, come le proprie chiavi di accesso.

> **Important**
>
> Consigliamo di impostare una policy delle password personalizzata che richieda agli utenti IAM di creare password complesse.

Per consentire a tutti gli utenti IAM di modificare le proprie password

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
3. Nella sezione Password policy (Policy delle password), scegli Edit (Modifica).
4. Scegli Custom (Personalizzato) per utilizzare una policy di password personalizzata.
5. Seleziona Allow users to change their own password (Consenti agli utenti di modificare la propria password), quindi scegli Save changes (Salva modifiche). Ciò consente a tutti gli utenti nell'account di accedere `iam:ChangePassword` all'operazione solo per il proprio utente e all'operazione `iam:GetAccountPasswordPolicy`.
6. Fornisci agli utenti le seguenti istruzioni per modificare le password: Come un utente IAM può modificare la propria password.

AWS CLI

Esegui il comando seguente:

- `aws iam update-account-password-policy`

API

Per aggiornare la politica relativa alla password dell'account, esegui la seguente operazione:

- `UpdateAccountPasswordPolicy`

Per consentire agli utenti IAM selezionati di modificare le proprie password

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
3. Nella sezione Impostazioni account, assicurati che l'opzione Consenti a tutti gli utenti di cambiare la propria password non sia selezionata. Se questa casella di controllo è selezionata, tutti gli utenti possono modificare le proprie password. (Consulta la procedura precedente.)
4. Crea gli utenti che dovrebbero essere autorizzati a modificare la propria password, se non esistono ancora. Per informazioni dettagliate, vedi Crea un utente IAM nel tuo Account AWS.
5. (Facoltativo) Crea un gruppo IAM per gli utenti che possono modificare le loro password e aggiungi gli utenti dalla fase precedente a tale gruppo. Per informazioni dettagliate, vedi Gruppi di utenti IAM.
6. Assegnare la policy seguente al gruppo. Per ulteriori informazioni, consulta Gestire le policy IAM.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:GetAccountPasswordPolicy",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:ChangePassword",
      "Resource": "arn:aws:iam::*:user/${aws:username}"
    }
  ]
}
```

Questa policy garantisce l'accesso all'ChangePasswordazione, che consente agli utenti di modificare solo le proprie password dalla console, dagli Strumenti per Windows PowerShell o dall'API. AWS CLI Consente inoltre l'accesso all'GetAccountPasswordPolicyazione, che consente all'utente di visualizzare la politica in materia di password corrente; questa autorizzazione è necessaria affinché l'utente possa visualizzare la politica sulla password dell'account nella pagina Modifica password. L'utente deve essere autorizzato a leggere l'attuale politica in materia di password per assicurarsi che la password modificata soddisfi i requisiti della politica.

7. Fornisci agli utenti le seguenti istruzioni per modificare le password: Come un utente IAM può modificare la propria password.

Ulteriori informazioni

Per ulteriori informazioni sulla gestione delle credenziali, consultare i seguenti argomenti:

- Consentire agli utenti IAM di cambiare le loro password
- Password utente in AWS
- Configurare una policy delle password di un account per gli utenti IAM
- Gestire le policy IAM
- Come un utente IAM può modificare la propria password

Come un utente IAM può modificare la propria password

Se ti è stata concessa l'autorizzazione a modificare la tua password utente IAM, puoi utilizzare una pagina speciale Console di gestione AWS per farlo. Puoi anche usare l' AWS API AWS CLI or.

Argomenti

- Autorizzazioni richieste
- Come gli utenti IAM possono cambiare le proprie password (console)
- In che modo gli utenti IAM modificano la propria password (AWS CLI or AWS API)

Autorizzazioni richieste

Per modificare la password del proprio utente IAM, è necessario disporre delle autorizzazioni dalla policy seguente: AWS: consente agli utenti IAM di modificare la password della console sulla pagina Credenziali di sicurezza.

Come gli utenti IAM possono cambiare le proprie password (console)

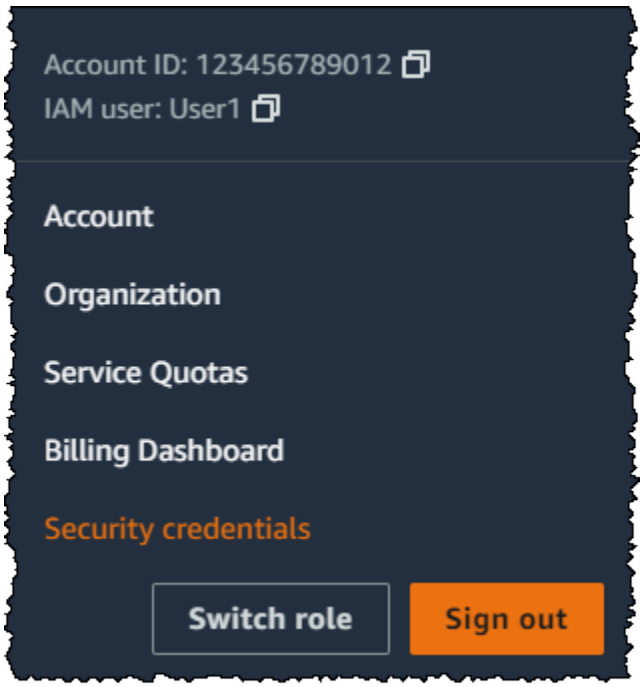
La procedura seguente descrive come gli utenti IAM possono utilizzare il Console di gestione AWS per modificare la propria password.

Come cambiare la propria password utente IAM (console)

1. Utilizza l' AWS ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM.

   > **Note**
   >
   > Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

   Per ottenere il tuo Account AWS ID, contatta l'amministratore.
2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella scheda Credenziali AWS IAM, seleziona Aggiorna password.
4. In Current password (Password corrente) digitare la password attuale. Digitare una nuova password per New password (Nuova password) e Confirm new password (Conferma nuova password). Quindi sceglie Aggiorna password.

> **Note**
>
> La nuova password deve soddisfare i requisiti della nuova policy delle password per l'account. Per ulteriori informazioni, consulta Configurare una policy delle password di un account per gli utenti IAM.

In che modo gli utenti IAM modificano la propria password (AWS CLI or AWS API)

La procedura seguente descrive come gli utenti IAM possono utilizzare l' AWS API AWS CLI or per modificare la propria password.

Per modificare la propria password IAM, utilizza i seguenti comandi:

- AWS CLI: `aws iam change-password`
- AWS API: `ChangePassword`

Gestione delle chiavi di accesso per gli utenti IAM

> **Important**
>
> Come best practice, utilizza credenziali di sicurezza temporanee (come i ruoli IAM) invece di creare credenziali a lungo termine come le chiavi di accesso. Prima di creare le chiavi di accesso, esamina le alternative alle chiavi di accesso a lungo termine.

Le chiavi di accesso sono credenziali a lungo termine per un utente IAM o l' Utente root dell'account AWS. Puoi utilizzare le chiavi di accesso per firmare le richieste programmatiche all' AWS API AWS CLI o (direttamente o utilizzando l' AWS SDK). Per ulteriori informazioni, consulta Accesso programmatico con AWS credenziali di sicurezza.

Le chiavi di accesso sono composte da due parti: un ID chiave di accesso (ad esempio `AKIAIOSFODNN7EXAMPLE`) e una chiave di accesso segreta (ad esempio, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). È necessario utilizzare sia l'ID chiave di accesso sia la chiave di accesso segreta insieme per autenticare le richieste dell'utente.

Se crei una coppia di chiavi di accesso, salva l'ID chiave di accesso e la chiave di accesso segreta in una posizione sicura. La chiave di accesso segreta può essere recuperata solo al momento della creazione. Se perdi la chiave di accesso segreta, è necessario eliminarla e crearne una nuova. Per ulteriori istruzioni, consulta Aggiornare le chiavi di accesso.

È possibile avere al massimo due chiavi di accesso per utente.

> **Important**
>
> Gli utenti IAM con chiavi di accesso rappresentano un rischio per la sicurezza dell'account. Gestisci le chiavi di accesso in modo sicuro. Non fornire le chiavi di accesso a parti non autorizzate, neppure per contribuire a trovare gli identificatori di account. In questo modo, potresti concedere a qualcuno l'accesso permanente al tuo account.
> Quando utilizzi le chiavi di accesso, tieni presente quanto segue:
>
> - NON utilizzare le credenziali root dell'account per creare chiavi di accesso.
> - NON inserire chiavi di accesso o informazioni sulle credenziali nei file dell'applicazione.
> - NON includere file che contengono chiavi di accesso o informazioni sulle credenziali nell'area del progetto.

- Le chiavi di accesso o le informazioni sulle credenziali archiviate nel file delle AWS credenziali condivise vengono archiviate in testo non crittografato.

Consigli sul monitoraggio

Dopo aver creato le chiavi di accesso:

- Utilizzatelo AWS CloudTrail per monitorare l'utilizzo delle chiavi di accesso e rilevare eventuali tentativi di accesso non autorizzati. Per ulteriori informazioni, consulta Registrazione di IAM e AWS STS Chiamate API con AWS CloudTrail.
- Imposta CloudWatch allarmi per notificare agli amministratori i tentativi di accesso negato per aiutare a rilevare attività dannose. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.
- Rivedi, aggiorna ed elimina regolarmente le chiavi di accesso secondo necessità.

I seguenti argomenti descrivono in dettaglio le attività di gestione associate alle chiavi di accesso.

Argomenti

- Controllo dell'uso delle chiavi di accesso collegando una policy inline a un utente IAM
- Autorizzazioni necessarie per gestire le chiavi di accesso
- Come gli utenti IAM possono gestire le proprie chiavi di accesso
- In che modo un amministratore IAM può gestire le chiavi di accesso dell'utente IAM
- Aggiornare le chiavi di accesso
- Proteggere le chiavi di accesso

Controllo dell'uso delle chiavi di accesso collegando una policy inline a un utente IAM

Come best practice, consigliamo di far utilizzare ai carichi di lavoro credenziali temporanee con ruoli IAM per l'accesso ad AWS. Agli utenti IAM con chiavi di accesso deve essere assegnato l'accesso con privilegio minimo e deve essere abilitata l'autenticazione a più fattori (MFA). Per ulteriori informazioni sull'assunzione dei ruoli IAM, consulta Metodi per assumere un ruolo.

Tuttavia, se stai creando un test di prova dell'automazione di un servizio o di un altro caso d'uso a breve termine e scegli di eseguire i carichi di lavoro utilizzando un utente IAM con chiavi di accesso,

ti consigliamo di utilizzare le condizioni delle policy per limitare ulteriormente l'accesso alle sue credenziali utente IAM.

In questa situazione puoi creare una policy a tempo limitato che faccia scadere le credenziali dopo il tempo specificato oppure, se stai eseguendo un carico di lavoro da una rete sicura, puoi utilizzare una policy di limitazione dell'IP.

Per entrambi questi casi d'uso, puoi utilizzare una policy inline collegata all'utente IAM che dispone delle chiavi di accesso.

Per configurare una policy a tempo limitato per un'utente IAM

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel pannello di navigazione, scegli Utenti, quindi seleziona l'utente per il caso d'uso a breve termine. Se non hai ancora creato l'utente, puoi crearlo ora.
3. Nella pagina Dettagli, scegli la scheda Autorizzazioni.
4. Scegli Aggiungi autorizzazioni, quindi scegli Crea policy inline.
5. Nella sezione Editor di policy, seleziona JSON per visualizzare l'editor JSON.
6. Nell'editor JSON, immetti la seguente policy, sostituendo il valore del timestamp `aws:CurrentTime` con la data e l'ora di scadenza desiderate:

   JSON

```
{
"Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
      "DateGreaterThan": {
      "aws:CurrentTime": "2025-03-01T00:12:00Z"
        }
      }
    }
  ]
}
```

Questa policy utilizza l'effetto `Deny` per limitare tutte le operazioni su tutte le risorse dopo la data specificata. La condizione `DateGreaterThan` confronta l'ora corrente con il timestamp impostato.

7. Seleziona Next (Successivo) per passare alla pagina Review and create (Rivedi e crea). Nei dettagli della Policy, in Nome della policy inserisci un nome per la policy, quindi scegli Crea policy.

Una volta creata, la policy viene visualizzata nella scheda Autorizzazioni per l'utente. Quando l'ora corrente è maggiore o uguale al tempo specificato nella policy, l'utente non avrà più accesso alle AWS risorse. Assicurati di informare gli sviluppatori dei carichi di lavoro della data di scadenza specificata per queste chiavi di accesso.

Per configurare una policy di limitazione dell'IP per un utente IAM

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Utenti, quindi seleziona l'utente che eseguirà il carico di lavoro dalla rete sicura. Se non hai ancora creato l'utente, puoi crearlo ora.
3. Nella pagina Dettagli, scegli la scheda Autorizzazioni.
4. Scegli Aggiungi autorizzazioni, quindi scegli Crea policy inline.
5. Nella sezione Editor di policy, seleziona JSON per visualizzare l'editor JSON.
6. Copia la seguente policy IAM nell'editor JSON e modifica gli intervalli o gli indirizzi pubblici IPv4 o IPv6 in base alle tue esigenze. Puoi utilizzarlo https://checkip.amazonaws.com/ per determinare il tuo attuale indirizzo IP pubblico. Puoi specificare singoli indirizzi IP o intervalli di indirizzi IP utilizzando la notazione slash. Per ulteriori informazioni, consulta era: SourceIp.

> **Note**
>
> Gli indirizzi IP non devono essere offuscati da una VPN o da un server proxy.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"IpRestrictionIAMPolicyForIAMUser",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/24",
            "2001:DB8:1234:5678::/64",
            "203.0.114.1"
          ]
        },
        "BoolIfExists": {
          "aws:ViaAWSService": "false"
        }
      }
    }
  ]
}
```

Questo esempio di policy nega l'uso delle chiavi di accesso di un utente IAM con questa policy applicata, a meno che la richiesta non provenga dalle reti (specificate nella notazione CIDR) «203.0.113». 0/24», «2001DB8:1234:5678: :/64» o l'indirizzo IP specifico «203.0.114.1»

7. Seleziona Next (Successivo) per passare alla pagina Review and create (Rivedi e crea). Nei dettagli della Policy, in Nome della policy inserisci un nome per la policy, quindi scegli Crea policy.

Una volta creata, la policy viene visualizzata nella scheda Autorizzazioni per l'utente.

Puoi anche applicare questa politica come policy di controllo del servizio (SCP) su più AWS account. Ti consigliamo di utilizzare una condizione aggiuntiva AWS Organizations, in modo che questa dichiarazione si applichi solo agli utenti IAM all'interno degli account soggetti a questo SCP. `aws:PrincipalArn` AWS La seguente politica include tale aggiornamento:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IpRestrictionServiceControlPolicyForIAMUsers",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/24",
            "2001:DB8:1234:5678::/64",
            "203.0.114.1"
          ]
        },
        "BoolIfExists": {
          "aws:ViaAWSService": "false"
        },
        "ArnLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:user/*"
        }
      }
    }
  ]
}
```

Autorizzazioni necessarie per gestire le chiavi di accesso

> **Note**
>
> `iam:TagUser` è un'autorizzazione facoltativa per l'aggiunta e la modifica di descrizioni della chiave di accesso. Per ulteriori informazioni, consulta Aggiungere tag agli utenti IAM

Per creare le chiavi di accesso per l'utente IAM, è necessario disporre delle autorizzazioni dalla policy seguente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CreateOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
                "iam:GetUser",
                "iam:ListAccessKeys",
                "iam:TagUser"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

Per aggiornare le chiavi di accesso per l'utente IAM, è necessario disporre delle autorizzazioni concesse dalla policy seguente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ManageOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
                "iam:DeleteAccessKey",
                "iam:GetAccessKeyLastUsed",
                "iam:GetUser",
                "iam:ListAccessKeys",
                "iam:UpdateAccessKey",
                "iam:TagUser"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
```

```
        }
    ]
}
```

Come gli utenti IAM possono gestire le proprie chiavi di accesso

Gli amministratori IAM possono concedere agli utenti IAM l'autorizzazione per gestire autonomamente le proprie chiavi di accesso collegando la policy descritta in Autorizzazioni necessarie per gestire le chiavi di accesso.

Con queste autorizzazioni, l'utente IAM può utilizzare le seguenti procedure per creare, attivare, disattivare ed eliminare le chiavi di accesso associate al proprio nome utente.

Argomenti

- Creare una chiave di accesso per se stessi (console)
- Disattivare la tua chiave di accesso (console)
- Attivare la tua chiave di accesso (console)
- Eliminare la tua chiave di accesso (console)

Creare una chiave di accesso per se stessi (console)

Se ti sono state concesse le autorizzazioni appropriate, puoi utilizzarle Console di gestione AWS per creare le tue chiavi di accesso.

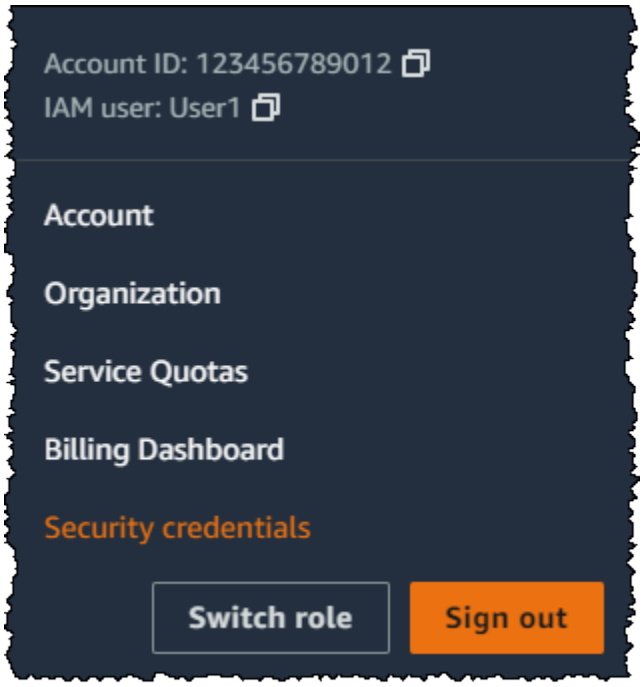
Per creare le proprie chiavi di accesso (console)

1. Utilizza l' AWS ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM.

> **Note**
>
> Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

Per ottenere il tuo Account AWS ID, contatta l'amministratore.

2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella sezione Chiavi di accesso, scegliere Crea chiave di accesso. Se dispone già di due chiavi di accesso, questo pulsante è disattivato e sarà necessario eliminare una chiave di accesso prima di crearne una nuova.
4. Sulla pagina Access key best practices & alternatives (Best practice e alternative per le chiavi di accesso), scegli il tuo caso d'uso per scoprire altre opzioni che possono aiutarti a evitare di creare una chiave di accesso a lungo termine. Se ritieni che il tuo caso d'uso richieda comunque una chiave di accesso, scegli Other (Altro) e poi Next (Successivo).
5. (Facoltativo) Imposta un valore del tag descrittivo per la chiave di accesso. Questo aggiunge una coppia chiave-valore di tag all'utente IAM. Ciò consente di identificare e aggiornare le chiavi di accesso in un secondo momento. La chiave di tag è impostata sull'ID della chiave di accesso. Il valore del tag è impostato sulla descrizione della chiave di accesso specificata. Al termine, scegli Create access key (Crea chiave di accesso).
6. Nella pagina Retrieve access keys (Recupera chiavi di accesso), scegli Show (Mostra) per rivelare il valore della chiave di accesso segreta dell'utente o Download .csv file (Scarica il file .csv). Questa è la tua unica opportunità di salvare la chiave di accesso segreta. Dopo aver salvato la chiave di accesso segreta in una posizione sicura, scegli Done (Fatto).

Disattivare la tua chiave di accesso (console)

Se ti sono state concesse le autorizzazioni appropriate, puoi utilizzarle Console di gestione AWS per disattivare la tua chiave di accesso.

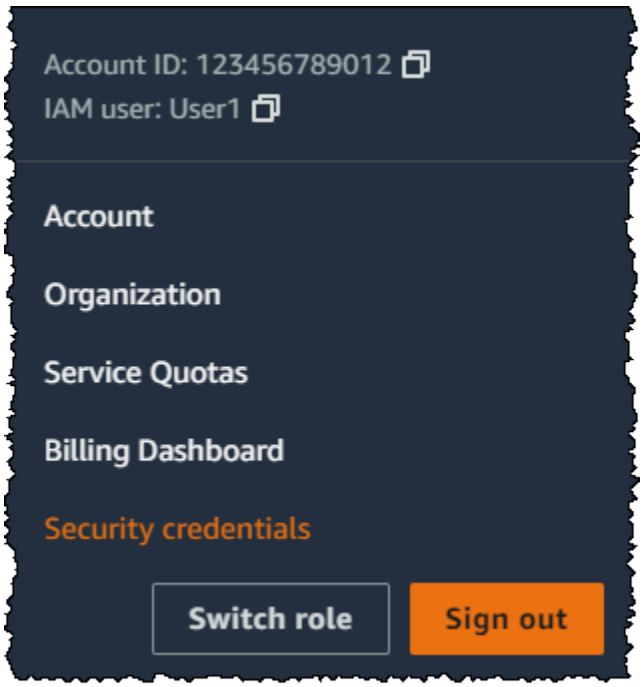
Disattivazione di una chiave di accesso

1. Utilizza l'ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM. AWS

   > **Note**
   >
   > Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

   Per ottenere il tuo Account AWS ID, contatta l'amministratore.

2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella sezione Access keys (Chiavi di accesso) individua la chiave che desideri disattivare, quindi scegli Actions (Operazioni) e poi Deactivate (Disattiva). Quando viene richiesta la conferma, scegliere Disattiva. Una chiave di accesso disattivata viene comunque conteggiata per il limite di due chiavi di accesso.

Attivare la tua chiave di accesso (console)

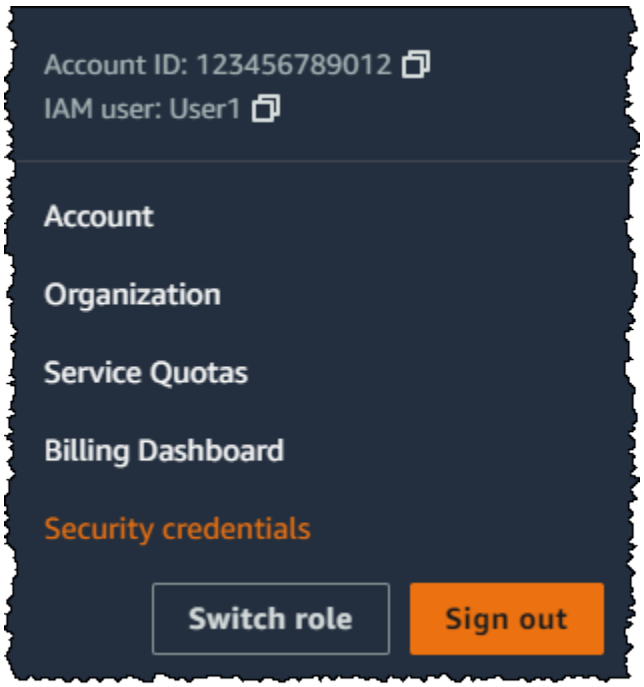
Se ti sono state concesse le autorizzazioni appropriate, puoi utilizzarle Console di gestione AWS per attivare la tua chiave di accesso.

Attivazione di una chiave di accesso

1. Utilizza l' AWS ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM.

   > **Note**
   >
   > Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

   Per ottenere il tuo Account AWS ID, contatta l'amministratore.
2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella sezione Access keys (Chiavi di accesso) individua la chiave che desideri attivare, quindi scegli Actions (Operazioni) e poi Activate (Attiva).

Eliminare la tua chiave di accesso (console)

Se ti sono state concesse le autorizzazioni appropriate, puoi utilizzarle Console di gestione AWS per eliminare la tua chiave di accesso.

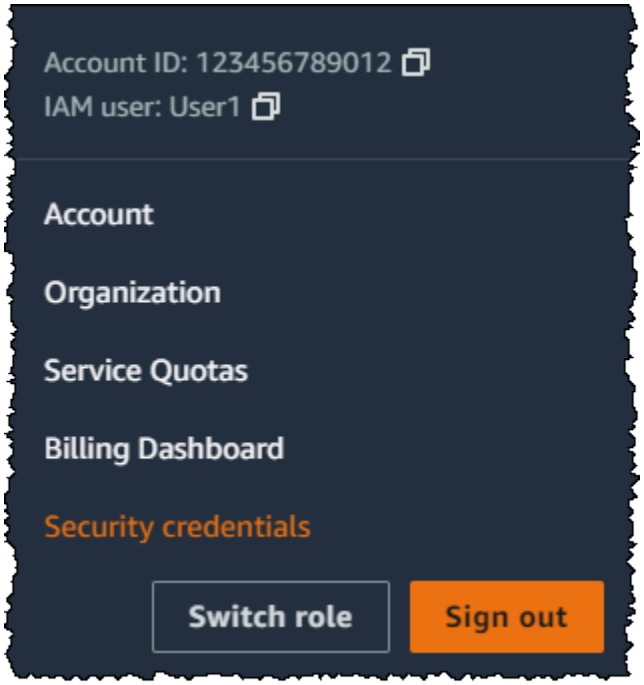
Eliminazione di una chiave di accesso quando non è più necessaria

1. Utilizza l' AWS ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM.

> **Note**
>
> Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

Per ottenere il tuo Account AWS ID, contatta l'amministratore.

2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella sezione Access keys (Chiavi di accesso) individua la chiave che desideri eliminare, quindi scegli Actions (Operazioni) e poi Delete (Elimina). Segui le istruzioni nella finestra di dialogo prima per disattivare la chiave e poi conferma l'eliminazione. Si consiglia di verificare che la chiave di accesso non sia più in uso prima di eliminarla definitivamente.

In che modo un amministratore IAM può gestire le chiavi di accesso dell'utente IAM

Gli amministratori IAM possono creare, attivare, disattivare ed eliminare le chiavi di accesso associate ai singoli utenti IAM. Possono anche elencare gli utenti IAM dell'account che dispongono delle chiavi di accesso e individuare quale utente IAM dispone di una chiave di accesso specifica.

Argomenti

- Come creare una chiave di accesso per un utente IAM
- Per disattivare una chiave di accesso per un utente IAM
- Per attivare una chiave di accesso per un utente IAM
- Come eliminare una chiave di accesso per un utente IAM

- Per elencare le chiavi di accesso per un utente IAM
- Per visualizzare tutti gli ID delle chiavi di accesso per gli utenti del tuo account
- Per utilizzare l'ID di una chiave di accesso per trovare un utente
- Per trovare l'uso più recente di un ID della chiave di accesso

Come creare una chiave di accesso per un utente IAM

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Scegli il nome utente per passare alla pagina dei dettagli dell'utente.
4. Nella scheda Credenziali di sicurezza, nella sezione Chiavi di accesso, scegli Crea chiave di accesso.

   Se il pulsante è disattivato, dovrai eliminare una delle chiavi esistenti prima di poterne creare una nuova.
5. Sulla pagina Access key best practices & alternatives (Best practice e alternative per le chiavi di accesso), esamina le best practice e le alternative. Scegli il tuo caso d'uso per scoprire altre opzioni che possono aiutarti a evitare di creare una chiave di accesso a lungo termine.
6. Se ritieni che il tuo caso d'uso richieda comunque una chiave di accesso, scegli Other (Altro) e poi Next (Successivo).
7. (Facoltativo) Nella pagina Imposta il tag di descrizione, puoi aggiungere un tag descrittivo alla chiave di accesso per monitorare la chiave di accesso. Seleziona Crea chiave di accesso.
8. Nella pagina Retrieve access key (Recupera chiave di accesso), scegli Show (Mostra) per rivelare il valore della chiave di accesso segreta dell'utente.
9. Per salvare l'ID della chiave di accesso e la chiave di accesso segreta in un file `.csv` in una posizione sicura sul computer, seleziona il pulsante Download .csv file (Scarica file .csv).

   > **Important**
   >
   > Questa è l'unica opportunità per visualizzare o scaricare la chiave di accesso appena creata e non sarà possibile recuperarla. Assicurati di conservare in modo sicuro la tua chiave di accesso.

Quando crei una chiave di accesso per il tuo utente, la coppia di chiavi è attiva di default e può essere utilizzata immediatamente.

AWS CLI

Esegui il comando seguente:

- `aws iam create-access-key`

API

Chiamare l'operazione seguente:

- `CreateAccessKey`

Per disattivare una chiave di accesso per un utente IAM

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Scegli il nome utente per passare alla pagina dei dettagli dell'utente.
4. Nella scheda Credenziali di sicurezza, nella sezione Chiavi di accesso, scegli il menu a discesa Azioni, quindi scegli Disattiva.
5. Nella finestra di dialogo Disattiva, conferma di voler disattivare la chiave di accesso selezionando Disattiva

Una volta disattivata, una chiave di accesso non può più essere utilizzata dalle chiamate API. Se necessario, puoi riattivarla.

AWS CLI

Esegui il comando seguente:

- `aws iam update-access-key`

API

Chiamare l'operazione seguente:

- `UpdateAccessKey`

Per attivare una chiave di accesso per un utente IAM

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Scegli il nome utente per passare alla pagina dei dettagli dell'utente.
4. Nella scheda Credenziali di sicurezza, nella sezione Chiavi di accesso, scegli il menu a discesa Azioni, quindi scegli Attiva.

Una volta attivata, una chiave di accesso può essere utilizzata dalle chiamate API. Se necessario, puoi disattivarla.

AWS CLI

Esegui il comando seguente:

- `aws iam update-access-key`

API

Chiamare l'operazione seguente:

- `UpdateAccessKey`

Come eliminare una chiave di accesso per un utente IAM

Dopo aver disattivato una chiave di accesso, se non è più necessaria, eliminala.

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione, seleziona Utenti.
3. Scegli il nome utente per passare alla pagina dei dettagli dell'utente.
4. Nella scheda Credenziali di sicurezza, nella sezione Chiavi di accesso, scegli il menu a discesa Azioni per la chiave di accesso inattiva, quindi scegli Elimina.
5. Nella finestra di dialogo Elimina, conferma di voler eliminare la chiave di accesso inserendo l'ID della chiave di accesso nel campo di immissione del testo e quindi selezionando Elimina.

Una volta eliminata, una chiave di accesso non può essere recuperata.

AWS CLI

Esegui il comando seguente:

- `aws iam delete-access-key`

API

Chiamare l'operazione seguente:

- `DeleteAccessKey`

Per elencare le chiavi di accesso per un utente IAM

Puoi visualizzare un elenco degli ID delle chiavi di accesso associate a un utente IAM.

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Scegli il nome utente per passare alla pagina dei dettagli dell'utente.
4. Nella scheda Credenziali di sicurezza, la sezione Chiavi di accesso elenca gli ID delle chiavi di accesso per l'utente, incluso lo stato di ciascuna chiave visualizzata.

> **Note**
>
> Solo l'ID chiave di accesso dell'utente è visibile. La chiave di accesso segreta può essere recuperata solo al momento della creazione.

Ogni utente IAM può avere due chiavi di accesso.

AWS CLI

Esegui il comando seguente:

- `aws iam list-access-keys`

API

Chiamare l'operazione seguente:

- `ListAccessKeys`

Per visualizzare tutti gli ID delle chiavi di accesso per gli utenti del tuo account

È possibile visualizzare un elenco degli ID delle chiavi di accesso per gli utenti nell' Account AWS.

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Scegli il nome utente per passare alla pagina dei dettagli dell'utente.
4. Se necessario, aggiungere la colonna Access key ID (ID chiave di accesso) alla tabella degli utenti mediante la procedura seguente:
   a. Sopra la tabella all'estrema destra, seleziona l'icona Preferenze ( ).
   b. Nella finestra di dialogo Preferenze, in Seleziona colonne visibili, attiva ID della chiave di accesso.

c. Seleziona Confirm (Conferma) per tornare all'elenco degli utenti. L'elenco viene aggiornato in modo da includere l'ID della chiave di accesso.

5. La colonna ID chiave di accesso mostra lo stato di ogni chiave di accesso seguito dal relativo ID, ad esempio **`Active - AKIAIOSFODNN7EXAMPLE`** o **`Inactive - AKIAI44QH8DHBEXAMPLE`**.

   È possibile utilizzare queste informazioni per visualizzare e copiare gli ID delle chiavi di accesso per gli utenti con una o due chiavi di accesso. Per gli utenti senza chiavi di accesso nella colonna è riportato **`-`**.

   > **Note**
   >
   > La chiave di accesso segreta può essere recuperata solo al momento della creazione.

Ogni utente IAM può avere due chiavi di accesso.

Per utilizzare l'ID di una chiave di accesso per trovare un utente

Puoi utilizzare un ID di chiave di accesso per trovare un utente nel tuo Account AWS.

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, nella casella di ricerca, inserisci il valore ID chiave di accesso, ad esempio AKIAI44QH8DHBEXAMPLE.
3. L'utente IAM a cui è associato l'ID della chiave di accesso viene visualizzato nel riquadro di navigazione. Scegli il nome utente per passare alla pagina dei dettagli dell'utente.

Per trovare l'uso più recente di un ID della chiave di accesso

L'uso più recente di una chiave di accesso viene visualizzato nell'elenco degli utenti nella pagina degli utenti IAM, nella pagina dei dettagli dell'utente e fa parte del report sulle credenziali.

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nell'elenco degli utenti, consulta la colonna Ultimo utilizzo chiave di accesso.

   Se la colonna non è visualizzata, scegli l'icona Preferenze
   ( )
   e in Seleziona colonne visibili attiva l'opzione Chiave di accesso utilizzata per l'ultima volta per visualizzare la colonna.
3. (Facoltativo) Nel riquadro di navigazione, in Report di accesso, seleziona Report delle credenziali per scaricare un report che include le informazioni sull'ultimo utilizzo della chiave di accesso per tutti gli utenti IAM del tuo account.
4. (Facoltativo) Seleziona l'utente IAM per visualizzare i dettagli dell'utente. La sezione Riepilogo include gli ID delle chiavi di accesso, il loro stato e l'ultima volta che sono state utilizzate.

AWS CLI

Esegui il comando seguente:

- `aws iam get-access-key-last-used`

API

Chiamare l'operazione seguente:

- `GetAccessKeyLastUsed`

Aggiornare le chiavi di accesso

Come best practice di sicurezza, è consigliabile aggiornare le chiavi di accesso degli utenti IAM all'occorrenza, ad esempio quando un dipendente lascia l'azienda. Gli utenti IAM possono aggiornare le proprie chiavi di accesso se dispongono delle autorizzazioni necessarie.

Per informazioni dettagliate su come concedere agli utenti IAM le autorizzazioni per aggiornare le proprie chiavi di accesso, consulta la pagina AWS: consente agli utenti IAM di gestire la propria

password, le chiavi di accesso e le chiavi pubbliche SSH nella pagina Credenziali di sicurezza. Inoltre, è possibile applicare all'account una policy delle password per richiedere che tutti gli utenti IAM aggiornino periodicamente le loro password e con quale frequenza. Per ulteriori informazioni, consulta Configurare una policy delle password di un account per gli utenti IAM.

> **Note**
>
> Se perdi la chiave di accesso segreta, è necessario eliminarla e crearne una nuova. La chiave di accesso segreta può essere recuperata solo al momento della creazione. Utilizza questa procedura per disattivare e quindi sostituire eventuali chiavi di accesso perse con nuove credenziali.

Argomenti

- Aggiornamento delle chiavi di accesso dell'utente IAM (console)
- Aggiornamento delle chiavi di accesso (AWS CLI)
- Aggiornamento delle chiavi di accesso (AWS API)

Aggiornamento delle chiavi di accesso dell'utente IAM (console)

È possibile aggiornare le chiavi di accesso dalla Console di gestione AWS.

Per aggiornare le chiavi di accesso per un utente IAM senza interrompere le applicazioni (console)

1. Mentre la prima chiave di accesso è ancora attiva, creare una seconda chiave di accesso.
    a. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
    b. Nel pannello di navigazione, seleziona Utenti.
    c. Selezionare il nome dell'utente, quindi selezionare la scheda Security credentials (Credenziali di sicurezza).
    d. Nella sezione Chiavi di accesso, scegliere Crea chiave di accesso. Sulla pagina Access key best practices & alternatives (Best practice e alternative per le chiavi di accesso), scegli Other (Altro), quindi scegli Next (Successivo).
    e. (Facoltativo) Imposta un valore del tag di descrizione per la chiave di accesso per aggiungere una coppia chiave-valore del tag a questo utente IAM. Ciò consente di identificare e aggiornare le chiavi di accesso in un secondo momento. La chiave di tag è

impostata sull'ID della chiave di accesso. Il valore del tag è impostato sulla descrizione della chiave di accesso specificata. Al termine, scegli Create access key (Crea chiave di accesso).

f. Nella pagina Retrieve access keys (Recupera chiavi di accesso), scegli Show (Mostra) per rivelare il valore della chiave di accesso segreta dell'utente o Download .csv file (Scarica il file .csv). Questa è la tua unica opportunità di salvare la chiave di accesso segreta. Dopo aver salvato la chiave di accesso segreta in una posizione sicura, scegli Done (Fatto).

   Quando crei una chiave di accesso per il tuo utente, la coppia di chiavi è attiva di default e può essere utilizzata immediatamente. A questo punto, l'utente dispone di due chiavi di accesso attive.

2. Aggiornare tutte le applicazioni e gli strumenti in modo che utilizzino la nuova chiave di accesso.
3. Determina se la prima chiave di accesso è ancora in uso consultando la colonna Last used (Ultimo utilizzo) della chiave di accesso più vecchia. Un approccio è aspettare diversi giorni e quindi verificare se la vecchia chiave di accesso sia stata utilizzata prima di procedere.
4. Anche se il valore della colonna Last used (Ultimo utilizzo) indica che la vecchia chiave non è mai stata utilizzata, è consigliabile non eliminare immediatamente la prima chiave di accesso. Al contrario, seleziona Actions (Azioni) e poi Deactivate (Disattiva) per disattivare la prima chiave di accesso.
5. Utilizzare solo la nuova chiave di accesso per verificare che le applicazioni funzionino. Tutte le applicazioni e gli strumenti che utilizzano ancora la chiave di accesso originale smetteranno di funzionare a questo punto perché non hanno più accesso alle AWS risorse. Se questo è il caso, puoi riattivare la prima chiave di accesso. Quindi, tornare a Step 3 e aggiornare l'applicazione in modo che utilizzi la nuova chiave.
6. Dopo aver atteso un certo periodo di tempo per verificare che tutte le applicazioni e gli strumenti siano stati aggiornati, puoi eliminare la prima chiave di accesso:

   a. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
   b. Nel pannello di navigazione, seleziona Utenti.
   c. Selezionare il nome dell'utente, quindi selezionare la scheda Security credentials (Credenziali di sicurezza).
   d. Nella sezione Access keys (Chiavi di accesso) individua la chiave di accesso che desideri eliminare, quindi scegli Actions (Operazioni) e poi Delete (Elimina). Segui le istruzioni nella finestra di dialogo prima per disattivare la chiave e poi per confermare l'eliminazione.

Per determinare quali chiavi di accesso devono essere aggiornate o eliminate (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Se necessario, aggiungere la colonna Access key age (Durata chiave di accesso) alla tabella degli utenti mediante la procedura seguente:

   a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ().

   b. In Manage columns (Gestisci colonne) selezionare Access key age (Durata chiave di accesso).

   c. Selezionare Close (Chiudi) per tornare all'elenco degli utenti.
4. La colonna Access key age (Durata chiave di accesso) mostra il numero di giorni trascorsi dalla creazione della più vecchia chiave di accesso attiva. È possibile utilizzare queste informazioni per trovare gli utenti per i quali potrebbe essere necessario aggiornare o eliminare le chiavi di accesso. La colonna visualizza None (Nessuna) per gli utenti senza chiavi di accesso.

Aggiornamento delle chiavi di accesso (AWS CLI)

È possibile aggiornare le chiavi di accesso dalla AWS Command Line Interface.

Per aggiornare le chiavi di accesso senza interrompere le applicazioni (AWS CLI)

1. Mentre la prima chiave di accesso è ancora attiva, creare una seconda chiave di accesso, che è attiva per default. Esegui il comando seguente:

   - `aws iam create-access-key`

   A questo punto, l'utente dispone di due chiavi di accesso attive.
2. Aggiornare tutte le applicazioni e gli strumenti in modo che utilizzino la nuova chiave di accesso.
3. Determinare se la prima chiave di accesso è ancora in uso utilizzando questo comando:

   - `aws iam get-access-key-last-used`

   Un approccio è aspettare diversi giorni e quindi verificare se la vecchia chiave di accesso sia stata utilizzata prima di procedere.

4. Anche se la fase Step 3 indica che la vecchia chiave non è stata utilizzata, è consigliabile non eliminare immediatamente la prima chiave di accesso. Al contrario, modificare lo stato della prima chiave di accesso in `Inactive` utilizzando questo comando:

   - `aws iam update-access-key`

5. Utilizzare solo la nuova chiave di accesso per verificare che le applicazioni funzionino. Tutte le applicazioni e gli strumenti che utilizzano ancora la chiave di accesso originale smetteranno di funzionare a questo punto perché non hanno più accesso alle AWS risorse. Se questo è il caso, è possibile ripristinare lo stato `Active` per riattivare la prima chiave di accesso. Quindi, tornare alla fase Step 2 e aggiornare l'applicazione in modo che utilizzi la nuova chiave.
6. Dopo aver atteso un certo periodo di tempo per verificare che tutte le applicazioni e gli strumenti siano stati aggiornati, puoi eliminare la prima chiave di accesso con questo comando:

   - `aws iam delete-access-key`

Aggiornamento delle chiavi di accesso (AWS API)

Puoi aggiornare le chiavi di accesso utilizzando l' AWS API.

Per aggiornare le chiavi di accesso senza interrompere le applicazioni (AWS API)

1. Mentre la prima chiave di accesso è ancora attiva, creare una seconda chiave di accesso, che è attiva per default. Chiamare l'operazione seguente:

   - `CreateAccessKey`

   A questo punto, l'utente dispone di due chiavi di accesso attive.
2. Aggiornare tutte le applicazioni e gli strumenti in modo che utilizzino la nuova chiave di accesso.
3. Determinare se la prima chiave di accesso è ancora in uso chiamando questa operazione:

   - `GetAccessKeyLastUsed`

   Un approccio è aspettare diversi giorni e quindi verificare se la vecchia chiave di accesso sia stata utilizzata prima di procedere.
4. Anche se la fase Step 3 indica che la vecchia chiave non è stata utilizzata, è consigliabile non eliminare immediatamente la prima chiave di accesso. Cambia invece lo stato della prima chiave di accesso `Inactive` chiamando questa operazione:

- `UpdateAccessKey`

5. Utilizzare solo la nuova chiave di accesso per verificare che le applicazioni funzionino. Tutte le applicazioni e gli strumenti che utilizzano ancora la chiave di accesso originale smetteranno di funzionare a questo punto perché non hanno più accesso alle AWS risorse. Se questo è il caso, è possibile ripristinare lo stato `Active` per riattivare la prima chiave di accesso. Quindi, tornare alla fase Step 2 e aggiornare l'applicazione in modo che utilizzi la nuova chiave.
6. Dopo aver atteso un certo periodo di tempo per verificare che tutte le applicazioni e gli strumenti siano stati aggiornati, puoi eliminare la prima chiave di accesso eseguendo questa operazione:

   - `DeleteAccessKey`

Proteggere le chiavi di accesso

Chiunque disponga delle tue chiavi di accesso ha lo stesso livello di accesso alle tue AWS risorse che hai tu. Di conseguenza, AWS fa di tutto per proteggere le tue chiavi di accesso e, in linea con il nostro modello di responsabilità condivisa, dovresti farlo anche tu.

Espandi le seguenti sezioni per ulteriori informazioni su come proteggere le chiavi di accesso.

> **Note**
>
> La tua organizzazione potrebbe avere policy e requisiti di sicurezza differenti rispetto a quelli descritti in questo argomento. I suggerimenti qui forniti sono destinati a essere linee guida generali.

Rimuovi (o non generare) Utente root dell'account AWS chiavi di accesso

Uno dei modi migliori per proteggere il tuo account è non disporre di chiavi di accesso dell' Utente root dell'account AWS. A meno che non necessiti di disporre delle chiavi di accesso dell'utente root (il che è raro), è consigliabile non generarle. Crea invece un utente amministrativo Centro identità AWS IAM per le attività amministrative quotidiane. Per informazioni su come creare un utente amministrativo in IAM Identity Center, consulta la Guida introduttiva nella Guida per l'utente di IAM Identity Center.

Se già disponi di chiavi di accesso dell'utente root per il tuo account, ti consigliamo di attenerti alle seguenti indicazioni: trova i punti nelle applicazioni in cui stai attualmente utilizzando le

chiavi di accesso (se presenti) e sostituisci le chiavi di accesso dell'utente root con le chiavi di accesso dell'utente IAM. Quindi disabilita e rimuovi le chiavi di accesso dell'utente root. Per ulteriori informazioni sull'aggiornamento delle chiavi di accesso, consulta la pagina Aggiornare le chiavi di accesso

Utilizzo di credenziali di sicurezza temporanee (ruoli IAM) al posto delle chiavi di accesso a lungo termine

In molti scenari, non è necessaria una chiave di accesso a lungo termine a validità illimitata (come accade invece per gli utenti IAM). Al contrario, è possibile creare ruoli IAM e generare credenziali di sicurezza temporanee. Tali credenziali sono composte dall'ID della chiave di accesso e dalla chiave di accesso segreta, ma includono anche un token di sicurezza che ne indica la scadenza.

Long-term le chiavi di accesso, come quelle associate agli utenti IAM e all'utente root, rimangono valide finché non le revochi manualmente. Tuttavia, le credenziali di sicurezza temporanee ottenute tramite i ruoli IAM e altre funzionalità di IAM Servizio di token di sicurezza AWS scadono dopo un breve periodo di tempo. Utilizza le credenziali di sicurezza temporanee per ridurre i rischi in caso di esposizione accidentale delle credenziali.

Utilizzare un ruolo IAM e le credenziali di sicurezza temporanee in questi scenari:

- Hai un'applicazione o AWS CLI degli script in esecuzione su un'istanza Amazon EC2. Non utilizzare le chiavi di accesso direttamente nell'applicazione. Non passare le chiavi di accesso all'applicazione, incorporarle nell'applicazione o lasciare che l'applicazione legga una chiave da qualsiasi origine. Al contrario, definisci un ruolo IAM con le autorizzazioni appropriate per l'applicazione e avvia l'istanza Amazon Elastic Compute Cloud (Amazon EC2) con i ruoli per EC2. In questo modo viene associato un ruolo IAM all'istanza Amazon EC2. Questa pratica, inoltre, consente all'applicazione di ottenere credenziali di sicurezza temporanee, che a sua volta può utilizzare per effettuare chiamate a livello di programmazione ad AWS. Gli AWS SDK e AWS Command Line Interface (AWS CLI) possono ottenere automaticamente credenziali temporanee dal ruolo.
- Devi concedere l'accesso tra account. Utilizzare un ruolo IAM per stabilire l'attendibilità tra gli account, quindi concedere agli utenti di un account autorizzazioni limitate per accedere all'account attendibile. Per ulteriori informazioni, consulta Tutorial IAM: Delega l'accesso attraverso AWS account che utilizzano ruoli IAM.
- Hai a disposizione un'app mobile. Non integrare le chiavi di accesso con l'app, anche nell'archiviazione crittografata. Al contrario, utilizzare Amazon Cognito per la gestione dell'identità degli utenti nell'applicazione. Questo servizio consente di autenticare gli utenti utilizzando Login

with Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC). È quindi possibile utilizzare il provider di credenziali Amazon Cognito per gestire le credenziali che l'app usa per le richieste ad AWS.

- Vuoi federarti AWS e la tua organizzazione supporta SAML 2.0. Se si lavora per un'organizzazione che dispone di un provider di identità che supporta SAML 2.0, configurare il provider per l'utilizzo di SAML. Puoi utilizzare SAML per scambiare informazioni di autenticazione AWS e recuperare un set di credenziali di sicurezza temporanee. Per ulteriori informazioni, consulta Federazione SAML 2.0.
- Se desideri effettuare la federazione, la tua organizzazione dispone di un archivio di identità locale. AWS Se gli utenti possono autenticarsi all'interno della tua organizzazione, puoi scrivere un'applicazione in grado di rilasciare loro credenziali di sicurezza temporanee per l'accesso alle AWS risorse. Per ulteriori informazioni, consulta Abilita l'accesso personalizzato del broker di identità al AWS console.
- Utilizza le condizioni nelle policy IAM per consentire l'accesso solo dalle reti previste. Puoi limitare dove e come vengono utilizzate le tue chiavi di accesso implementando policy IAM con condizioni che specificano e consentano solo le reti previste, come gli indirizzi IP pubblici o i cloud privati virtuali (VPC). In questo modo sai che le chiavi di accesso possono essere utilizzate solo da reti previste e accettabili.

> **Note**
>
> Stai utilizzando un'istanza Amazon EC2 con un'applicazione che richiede l'accesso programmatico alle risorse? AWS In tal caso, utilizza i ruoli IAM per EC2.

Gestione corretta delle chiavi di accesso dell'utente IAM

Se devi creare chiavi di accesso per l'accesso programmatico a AWS, creale per gli utenti IAM, concedendo agli utenti solo le autorizzazioni necessarie.

Osserva queste precauzioni per proteggere le chiavi di accesso degli utenti IAM:

- Non incorporare le chiavi di accesso direttamente nel codice. Gli SDK AWS e gli Strumenti da linea di comando AWS consentono di collocare le chiavi di accesso in posizioni note in modo da evitare di conservarle nel codice.

  Colloca le chiavi di accesso in una delle posizioni seguenti:

- Il AWS file delle credenziali. Gli AWS SDK utilizzano AWS CLI automaticamente le credenziali archiviate nel file delle AWS credenziali.

  Per informazioni sull'utilizzo del file delle AWS credenziali, consulta la documentazione del tuo SDK. Gli esempi includono Set AWS Credentials and Region nella Guida per gli AWS SDK per Java sviluppatori e i file di configurazione e credenziali nella Guida per l'utente. AWS Command Line Interface

  Per archiviare le credenziali per AWS SDK per .NET e per AWS Tools for Windows PowerShell, ti consigliamo di utilizzare l'SDK Store. Per ulteriori informazioni, consulta Utilizzo dell'SDK Store nella Guida per gli sviluppatori di AWS SDK per .NET .

- Variabili di ambiente. In un sistema multi-tenant, scegli le variabili di ambiente dell'utente e non le variabili di ambiente del sistema.

  Per ulteriori informazioni sull'utilizzo di variabili di ambiente per archiviare le credenziali, consultare la sezione Variabili di ambiente nella Guida per l'utente di AWS Command Line Interface .

- Utilizza chiavi di accesso diverse per applicazioni differenti. Esegui questa operazione in modo da isolare le autorizzazioni e revocare le chiavi di accesso per le singole applicazioni in caso una di esse venga esposta. Avere chiavi di accesso separate per applicazioni diverse genera anche voci distinte nei file di log AWS CloudTrail. Questa configurazione consente di determinare più facilmente quale applicazione ha eseguito azioni specifiche.
- Aggiorna le chiavi di accesso all'occorrenza. Se esiste il rischio che la chiave di accesso possa essere compromessa, aggiorna la chiave di accesso ed elimina quella precedente. Per maggiori dettagli, consultare Aggiornare le chiavi di accesso.
- Rimuovi le chiavi di accesso inutilizzate. Se un utente lascia l'organizzazione, rimuovere l'utente IAM corrispondente in modo che non possa più accedere alle risorse. Per scoprire quando è stata utilizzata l'ultima volta una chiave di accesso, usa l'`GetAccessKeyLastUsed`API (AWS CLI comando: `aws iam get-access-key-last-used`).
- Utilizza le credenziali temporanee e configura l'autenticazione a più fattori (MFA) per le operazioni API più sensibili. Con le policy IAM, è possibile specificare le operazioni API che un utente è autorizzato a chiamare. In alcuni casi, potresti desiderare la sicurezza aggiuntiva di richiedere agli utenti di autenticarsi con AWS MFA prima di consentire loro di eseguire azioni particolarmente sensibili. Potrebbe ad esempio esserci una policy che permette a un utente di eseguire le operazioni `RunInstances`, `DescribeInstances` e `StopInstances` di Amazon EC2. Tuttavia, potresti voler limitare un'azione distruttiva come `TerminateInstances` fare in modo che gli utenti

possano eseguirla solo se si autenticano con un dispositivo MFA. AWS Per ulteriori informazioni, consulta Accesso sicuro alle API con MFA.

Accedi all'app mobile utilizzando AWS chiavi di accesso

Puoi accedere a un set limitato di AWS servizi e funzionalità utilizzando l'app AWS mobile. L'app mobile ti aiuta a supportare la risposta agli incidenti mentre sei in viaggio. Per ulteriori informazioni e per scaricare l'app, consulta AWS Console Mobile Application.

È possibile accedere all'app per dispositivi mobili utilizzando la password della console o le chiavi di accesso. Come best practice, non utilizzare le chiavi di accesso dell'utente root. Consigliamo invece vivamente, oltre a utilizzare una password o un lucchetto biometrico sul dispositivo mobile, di creare un utente IAM specifico per la gestione AWS delle risorse tramite l'app mobile. Se si perde il dispositivo mobile, è possibile rimuovere l'accesso dell'utente IAM.

Accesso mediante le chiavi di accesso (app per dispositivi mobili)

1. Apri l'app sul tuo dispositivo mobile.
2. Se questa è la prima volta che aggiungi un'identità al dispositivo, scegli Add an identity (Aggiungi un'identità) e scegli Access keys (Chiavi di accesso).

   Se hai già effettuato l'accesso utilizzando un'altra identità, scegli l'icona del menu e scegli Switch identity (Cambia identità). Quindi scegli Sign in as a different identity (Accedi come identità diversa) e quindi Access keys (Chiavi di accesso).
3. Nella pagina Access keys (Chiavi di accesso) immetti le informazioni nei campi.

   - ID chiave di accesso: immettere l'ID chiave di accesso.
   - Chiave di accesso segreta: inserire la chiave di accesso segreta.
   - Nome dell'identità: immettere il nome dell'identità che verrà visualizzata nell'applicazione per dispositivi mobili. Non è necessario che corrisponda al nome utente IAM.
   - PIN identità: creare un PIN (Personal Identification Number) da utilizzare per gli accessi futuri.

   > **Note**
   >
   > Se abiliti la biometria per l'app AWS mobile, ti verrà richiesto di utilizzare l'impronta digitale o il riconoscimento facciale per la verifica anziché il PIN. Se la biometria restituisce un errore, potrebbe venire richiesto il PIN.

4. Scegliere Verify and add keys (Verifica e aggiungi chiavi).

   È ora possibile accedere a un set selezionato di risorse mediante l'app per dispositivi mobili.

Informazioni correlate

I seguenti argomenti forniscono indicazioni per la configurazione degli AWS SDK e l'uso delle chiavi di accesso: AWS CLI

- Imposta AWS le credenziali e la regione nella Guida per gli sviluppatori AWS SDK per Java
- Utilizzo dell'SDK Store nella Guida per gli sviluppatori di AWS SDK per .NET .
- Specifica delle credenziali all'SDK nella Guida per gli sviluppatori di AWS SDK per PHP .
- Configurazione nella documentazione di Boto 3 (AWS SDK per Python)
- Utilizzo delle credenziali AWS nella Guida per l'utente di AWS Tools for Windows PowerShell
- File di configurazione e delle credenziali nella Guida per l'utente di AWS Command Line Interface .
- Concessione dell'accesso utilizzando un ruolo IAM nella Guida per gli sviluppatori di AWS SDK per .NET
- Configurazione dei ruoli IAM per Amazon EC2 nell'AWS SDK for Java 2.x

Utilizzo delle chiavi di accesso e delle credenziali segrete per l'accesso alla console

È possibile utilizzare la chiave di accesso e le credenziali della chiave segreta per l'accesso diretto alla Console di gestione AWS , non solo la AWS CLI. Ciò può essere ottenuto utilizzando la chiamata API. AWS STS `GetFederationToken` Creando un URL della console utilizzando le credenziali e il token temporanei forniti da `GetFederationToken`, i principali IAM possono accedere alla console. Per ulteriori informazioni, consulta Abilita l'accesso personalizzato del broker di identità al AWS console.

È bene chiarire che quando si accede alla console direttamente utilizzando credenziali utente IAM o root con MFA abilitata, sarà richiesta l'MFA. Tuttavia, se si utilizza il metodo sopra descritto (utilizzando credenziali temporanee con `GetFederationToken`), l'MFA NON sarà richiesta.

Audit delle chiavi di accesso

Puoi esaminare le chiavi di AWS accesso nel tuo codice per determinare se provengono da un account di tua proprietà. Puoi passare l'ID di una chiave di accesso utilizzando il `aws sts get-access-key-info` AWS CLI comando o l'operazione `GetAccessKeyInfo` AWS API.

Le operazioni AWS API AWS CLI e restituiscono l'ID della chiave di accesso Account AWS a cui appartiene la chiave di accesso. Gli ID delle chiavi di accesso che iniziano con `AKIA` sono credenziali a lungo termine per un utente IAM o un Utente root dell'account AWS. Gli ID delle chiavi di accesso che iniziano con `ASIA` sono credenziali temporanee create utilizzando AWS STS le operazioni. Se l'account nella risposta appartiene a te, puoi effettuare l'accesso come utente root e rivedere le chiavi di accesso dell'utente root. Quindi, puoi estrarre un report delle credenziali per scoprire quale utente IAM possiede le chiavi. Per sapere chi ha richiesto le credenziali temporanee per una chiave di `ASIA` accesso, visualizza gli AWS STS eventi nei tuoi CloudTrail log.

Per motivi di sicurezza, puoi esaminare AWS CloudTrail i log per scoprire chi ha eseguito un'azione in AWS. È possibile utilizzare la chiave di condizione `sts:SourceIdentity` nella policy di attendibilità del ruolo per richiedere agli utenti di specificare un'identità quando assumono un ruolo. Ad esempio, è possibile richiedere che gli utenti IAM specifichino il proprio nome utente come identità di origine. In questo modo è possibile determinare quale utente ha eseguito un'operazione specifica in AWS. Per ulteriori informazioni, consulta `sts:SourceIdentity`.

Questa operazione non indica lo stato della chiave di accesso. La chiave potrebbe essere attiva, inattiva o eliminata. Le chiavi attive potrebbero non disporre delle autorizzazioni per eseguire un'operazione. Fornire una chiave di accesso eliminata potrebbe restituire un errore indicante che la chiave non esiste.

AWS Multi-factor autenticazione in IAM

Per una maggiore sicurezza, ti consigliamo di configurare l'autenticazione a più fattori (MFA) per proteggere le tue risorse. AWS Puoi abilitare l'MFA per tutti Account AWS, inclusi gli account autonomi, gli account Utente root dell'account AWS di gestione e gli account dei membri, nonché per i tuoi utenti IAM. Ti consigliamo di utilizzare un MFA resistente al phishing, ad esempio passkey e chiavi di sicurezza, laddove possibile. Questi FIDO-based autenticatori utilizzano la crittografia a chiave pubblica e sono resistenti agli attacchi di phishing, man-in-the-middle e replay, fornendo un livello di sicurezza più elevato rispetto alle opzioni. TOTP-based

L'MFA viene applicata a tutti i tipi di account per il relativo utente root. Per ulteriori informazioni, consulta Proteggi i tuoi AWS Organizations credenziali utente root dell'account.

Quando abiliti MFA per l'utente root, questa impostazione influisce solo sulle credenziali dell'utente root. Gli utenti IAM nell'account sono identità distinte con proprie credenziali e ogni identità ha la propria configurazione MFA. Per ulteriori informazioni sull'utilizzo dell'MFA per proteggere l'utente root, consulta Multi-factor autenticazione per Utente root dell'account AWS.

I tuoi utenti Utente root dell'account AWS e quelli di IAM possono registrare fino a otto dispositivi MFA di qualsiasi tipo. La registrazione di più dispositivi MFA può offrire flessibilità e contribuire a ridurre il rischio di interruzione dell'accesso in caso di smarrimento o guasto di un dispositivo. È necessario un solo dispositivo MFA per accedere alla Console di gestione AWS o creare una sessione tramite la AWS CLI.

> **Note**
>
> Ti consigliamo di richiedere agli utenti umani di utilizzare credenziali temporanee per l'accesso. AWS Hai preso in considerazione l'utilizzo Centro identità AWS IAM? Puoi utilizzare IAM Identity Center per gestire centralmente l'accesso a più account Account AWS e fornire agli MFA-protected utenti l'accesso Single Sign-on a tutti gli account assegnati da un'unica posizione. Con IAM Identity Center puoi creare e gestire le identità degli utenti in IAM Identity Center o connetterti facilmente al tuo attuale gestore dell'identità digitale (IdP) compatibile con SAML 2.0. Per ulteriori informazioni, consulta Cos'è IAM Identity Center? nella Guida per l'utente di Centro identità AWS IAM .

L'MFA aggiunge una sicurezza aggiuntiva che richiede agli utenti di fornire un'autenticazione univoca tramite un meccanismo MFA AWS supportato oltre alle credenziali di accesso quando accedono a siti Web o servizi. AWS

Tipi di MFA

AWS supporta i seguenti tipi di MFA:

Indice

- Passkey e chiavi di sicurezza
- Applicazioni di autenticazione virtuale
- Token TOTP hardware

Passkey e chiavi di sicurezza

AWS Identity and Access Management supporta le chiavi di accesso e le chiavi di sicurezza per l'MFA. In base agli standard FIDO, le password utilizzano la crittografia a chiave pubblica per fornire un'autenticazione forte e resistente al phishing e più sicura delle password. AWS supporta due tipi di password: password associate al dispositivo (chiavi di sicurezza) e chiavi di accesso sincronizzate.

- Chiavi di sicurezza: si tratta di dispositivi fisici, ad esempio YubiKey, utilizzati come secondo fattore di autenticazione. Una singola chiave di sicurezza può supportare più account utente root e utenti IAM.
- Passkey sincronizzate: come secondo fattore utilizzano gestori di credenziali di provider come Google, Apple, account Microsoft e servizi di terze parti come 1Password, Dashlane e Bitwarden come secondo fattore.

Puoi utilizzare gli autenticatori biometrici integrati, come Touch ID su Apple MacBooks, per sbloccare il gestore delle credenziali e accedere a. AWS Le passkey vengono create con il provider scelto utilizzando l'impronta digitale, il viso o il PIN del dispositivo. Puoi anche utilizzare una passkey di autenticazione tra dispositivi (CDA) da un dispositivo, ad esempio un dispositivo mobile o una chiave di sicurezza hardware, per accedere su un altro dispositivo, ad esempio un laptop. Per ulteriori informazioni, consulta Cross-device authentication (CDA).

Puoi sincronizzare le password tra i tuoi dispositivi per facilitare l'accesso, migliorando l'usabilità e la recuperabilità. AWS Per ulteriori informazioni sull'abilitazione delle passkey e delle chiavi di sicurezza, consulta Abilitare una passkey o una chiave di sicurezza per l'utente root (console).

FIDO Alliance mantiene un elenco di tutti i prodotti certificati FIDO compatibili con le specifiche FIDO.

Applicazioni di autenticazione virtuale

Un'applicazione di autenticazione virtuale che viene eseguita su un telefono o altro dispositivo e simula un dispositivo fisico. Le app di autenticazione virtuale implementano l'algoritmo TOTP (password monouso) e supportano più token su un singolo dispositivo. L'utente deve immettere un codice valido dal dispositivo quando richiesto durante la procedura di accesso. Ogni token assegnato a un utente deve essere univoco. Per autenticarsi, un utente non può digitare un codice dal token di un altro utente.

Ti consigliamo di utilizzare un MFA resistente al phishing, ad esempio passkey o chiavi di sicurezza, per una protezione più efficace. ??? Se non sei ancora in grado di utilizzare password o chiavi di sicurezza, ti consigliamo di utilizzare un dispositivo MFA virtuale come misura provvisoria in attesa dell'approvazione dell'acquisto dell'hardware o in attesa dell'arrivo dell'hardware. Per un elenco di alcune app supportate che puoi utilizzare come dispositivi MFA virtuali, vedi Multi-Factor Autenticazione (MFA).

Per istruzioni sulla configurazione di un dispositivo MFA virtuale per un utente IAM, consulta Assegna un dispositivo MFA virtuale nel Console di gestione AWS.

> **Note**
>
> I dispositivi MFA virtuali non assegnati presenti nel tuo Account AWS vengono eliminati quando aggiungi nuovi dispositivi MFA virtuali tramite Console di gestione AWS o durante il processo di accesso. I dispositivi MFA virtuali non assegnati sono dispositivi presenti nell'account ma non utilizzati dall'utente root dell'account o dagli utenti IAM per il processo di accesso. Vengono eliminati in modo da poter aggiungere nuovi dispositivi MFA virtuali al tuo account. Questo consente anche di riutilizzare i nomi dei dispositivi.
>
> - Per visualizzare i dispositivi MFA virtuali non assegnati nel tuo account, puoi utilizzare il comando list-virtual-mfa-devices o la chiamata API. AWS CLI https://docs.aws.amazon.com/IAM/latest/APIReference/API_ListVirtualMFADevices.html
> - Per disattivare un dispositivo MFA virtuale, puoi utilizzare il comando deactivate-mfa-device o la chiamata API. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/deactivate-mfa-device.html AWS CLI https://docs.aws.amazon.com/IAM/latest/APIReference/API_DeactivateMFADevice.html Verrà annullata l'assegnazione del dispositivo.
> - Per collegare un dispositivo MFA virtuale non assegnato all'utente Account AWS root o agli utenti IAM, avrai bisogno del codice di autenticazione generato dal dispositivo insieme al comando enable-mfa-device o alla chiamata API. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/enable-mfa-device.html AWS CLI https://docs.aws.amazon.com/IAM/latest/APIReference/API_EnableMFADevice.html

Token TOTP hardware

Un dispositivo hardware che genera un codice numerico a sei cifre basato sull'algoritmo con password monouso. L'utente deve immettere un codice valido dal dispositivo su una seconda pagina Web durante la procedura di accesso.

Questi token vengono utilizzati esclusivamente con. Account AWS Puoi utilizzare solo i token con cui i relativi token seed univoci sono condivisi in modo sicuro. AWS I token seed sono chiavi segrete generate al momento della produzione dei token. I token acquistati da altre origini non funzioneranno con IAM. Per garantire la compatibilità, è necessario acquistare il dispositivo hardware MFA da uno dei seguenti link: token OTP o scheda video OTP.

- Ogni dispositivo MFA assegnato a un utente deve essere univoco. Per essere autenticati, gli utenti non possono digitare un codice generato dal dispositivo di un altro utente. Per informazioni sui dispositivi hardware MFA supportati, consulta Multi-Factor Autenticazione (MFA).
- Se desideri utilizzare un dispositivo MFA fisico, ti consigliamo di utilizzare le chiavi di sicurezza come alternativa ai dispositivi TOTP hardware. Le chiavi di sicurezza non richiedono batterie, sono resistenti al phishing e supportano più utenti su un singolo dispositivo.

Puoi abilitare una passkey o una chiave di sicurezza dall' Console di gestione AWS unica API, non dall' AWS CLI API or. AWS Prima di abilitare una chiave di sicurezza, è necessario disporre di un accesso fisico al dispositivo.

Per le istruzioni di configurazione di un token TOTP hardware per un utente IAM, consulta Assegna un token TOTP hardware nel Console di gestione AWS.

> **Note**
>
> MFA basato su SMS: AWS è terminato il supporto per l'abilitazione dell'autenticazione a più fattori (MFA) SMS. Consigliamo ai clienti con utenti IAM che utilizzano la MFA basata su SMS di passare a uno dei seguenti metodi alternativi di autenticazione a più fattori: passkey o chiave di sicurezza, dispositivo MFA virtuale (basato su software) o dispositivo MFA basato su hardware. Puoi identificare gli utenti nel tuo account con un dispositivo MFA SMS assegnato. Nella console IAM, seleziona Utenti dal pannello di navigazione e cerca gli utenti con l'opzione SMS nella colonna MFA della tabella.

Suggerimenti per MFA

Per proteggere le tue AWS identità, segui questi consigli per l'autenticazione MFA.

- Ti consigliamo di utilizzare un MFA resistente al phishing, come passkey e chiavi di sicurezza, come dispositivo MFA. Questi FIDO-based autenticatori offrono la protezione più efficace contro attacchi come il phishing.
- Ti consigliamo di abilitare più dispositivi MFA agli utenti IAM presenti nel Utente root dell'account AWS tuo. Account AWS Ciò consente di alzare il livello di sicurezza Account AWS e semplificare la gestione dell'accesso agli utenti con privilegi Utente root dell'account AWS elevati, come.
- Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi MFA attualmente supportati con i tuoi utenti Utente root dell'account AWS e quelli IAM. Con più dispositivi MFA, è

necessario un solo dispositivo MFA per accedere Console di gestione AWS o creare una sessione tramite tale utente. AWS CLI Per abilitare o disabilitare un dispositivo MFA aggiuntivo, un utente IAM deve prima autenticarsi con un dispositivo MFA esistente.

- In caso di smarrimento, furto o inaccessibilità di un dispositivo MFA, è possibile utilizzare uno dei dispositivi MFA rimanenti per accedervi senza eseguire la Account AWS procedura di ripristino. Account AWS In caso di smarrimento o furto di un dispositivo MFA, consigliamo di dissociare il dispositivo dal principale IAM a cui era associato.
- L'uso di più MFA consente ai dipendenti dislocati geograficamente o che lavorano in remoto di utilizzare l'MFA basato su hardware per accedere AWS senza dover coordinare lo scambio fisico di un singolo dispositivo hardware tra i dipendenti.
- L'uso di dispositivi MFA aggiuntivi per i principali IAM consente di utilizzare uno o più MFA per l'uso quotidiano, mantenendo al contempo i dispositivi MFA fisici in un luogo fisico sicuro come un vault o una cassaforte per il backup e la ridondanza.

> **Note**
>
> - Non è possibile trasmettere le informazioni MFA relative a una chiave di sicurezza o passkey alle operazioni API per richiedere credenziali temporanee. AWS STS È possibile ottenere le credenziali da utilizzare con gli AWS SDK AWS CLI e quando si utilizza una chiave di sicurezza o una passkey eseguendo il comando. `aws login`
> - Non è possibile utilizzare AWS CLI comandi o operazioni AWS API per abilitare le chiavi di sicurezza FIDO.
> - Non puoi utilizzare lo stesso nome per più di un utente root o dispositivo MFA IAM.

Risorse aggiuntive

Le seguenti risorse possono rivelarsi utili per saperne di più sull'MFA.

- Per ulteriori informazioni sull'utilizzo dell'MFA per accedere AWS, vedere. Accesso abilitato con MFA
- Puoi utilizzare IAM Identity Center per abilitare l'accesso MFA sicuro al tuo portale di AWS accesso, alle app integrate con IAM Identity Center e al. AWS CLI Per ulteriori informazioni, consulta Abilitare l'MFA nel Centro identità IAM.
- Per verificare se l'MFA è già abilitato per i tuoi utenti, consulta. Verifica dello stato MFA

Assegna una passkey o una chiave di sicurezza nel Console di gestione AWS

Le password sono un tipo di dispositivo di autenticazione a più fattori (MFA) che puoi utilizzare per proteggere le tue risorse. AWS AWS supporta le chiavi di accesso sincronizzate e le chiavi di accesso associate al dispositivo, note anche come chiavi di sicurezza.

Le passkey sincronizzate consentono agli utenti IAM di accedere alle proprie credenziali di accesso FIDO su molti dei loro dispositivi, anche su quelli nuovi, senza dover registrare nuovamente tutti i dispositivi su ogni account. Le passkey sincronizzate includono gestori di credenziali proprietari come Google, Apple e Microsoft e gestori di credenziali di terze parti come 1Password, Dashlane e Bitwarden come secondo fattore. Puoi anche utilizzare la biometria sul dispositivo (ad esempio, TouchID, FaceID) per sbloccare il gestore di credenziali scelto per utilizzare le passkey.

In alternativa, le passkey collegate al dispositivo sono associate a una chiave di sicurezza FIDO da collegare a una porta USB del computer e quindi toccare quando richiesto per completare in modo sicuro la procedura di accesso. Se utilizzi già una chiave di sicurezza FIDO con altri servizi e ha una configurazione AWS supportata (ad esempio, la serie YubiKey 5 di Yubico), puoi utilizzarla anche con. AWS Altrimenti, è necessario acquistare una chiave di sicurezza FIDO se si desidera utilizzarla per l' WebAuthn MFA in. AWS Inoltre, le chiavi di sicurezza FIDO possono supportare più utenti IAM o root sullo stesso dispositivo, migliorandone l'utilità per la sicurezza degli account. Per le specifiche e le informazioni di acquisto per entrambi i tipi di dispositivo, vedi Multi-Factor Autenticazione.

Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi MFA attualmente supportati con i tuoi utenti Utente root dell'account AWS e quelli IAM. Con più dispositivi MFA, è necessario un solo dispositivo MFA per accedere Console di gestione AWS o creare una sessione tramite tale utente. AWS CLI Consigliamo di registrare più dispositivi MFA. Ad esempio, è possibile registrare un autenticatore integrato e anche una chiave di sicurezza da conservare in un luogo fisicamente sicuro. Se è impossibile utilizzare l'autenticatore integrato, si può utilizzare la chiave di sicurezza registrata. Per le applicazioni di autenticazione, consigliamo inoltre di abilitare le funzionalità di backup o sincronizzazione su cloud in tali app per evitare di perdere l'accesso all'account in caso di smarrimento o guasto del dispositivo che dispone delle app di autenticazione.

> **Note**
>
> Consigliamo di richiedere agli utenti di utilizzare credenziali temporanee per l'accesso a AWS. Gli utenti possono connettersi con un provider di identità dove si autenticano AWS con le proprie credenziali aziendali e le configurazioni MFA. Per gestire l'accesso AWS

e le applicazioni aziendali, ti consigliamo di utilizzare IAM Identity Center. Per ulteriori informazioni, consulta la Guida per l'utente del Centro identità IAM.

Argomenti

- Autorizzazioni richieste
- Abilitare una passkey o una chiave di sicurezza per il proprio utente IAM (console)
- Abilitare una passkey o una chiave di sicurezza per un altro utente IAM (console)
- Sostituire una passkey o una chiave di sicurezza
- Configurazioni supportate per l'uso delle passkey e delle chiavi di sicurezza

Autorizzazioni richieste

Per gestire una passkey FIDO per il tuo utente IAM proteggendo al contempo MFA-related le azioni sensibili, devi disporre delle autorizzazioni previste dalla seguente politica:

> **Note**
>
> I valori dell'ARN sono valori statici e non sono un indicatore del protocollo utilizzato per registrare l'autenticatore. Abbiamo reso obsoleto l'U2F, quindi tutte le nuove implementazioni lo utilizzano. WebAuthn

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowManageOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
```

```
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "DenyAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
    ]
}
```

Abilitare una passkey o una chiave di sicurezza per il proprio utente IAM (console)

Puoi abilitare una passkey o una chiave di sicurezza per il tuo utente IAM dall' Console di gestione AWS unica API, non dall'API or. AWS CLI AWS Prima di abilitare una chiave di sicurezza, è necessario disporre di un accesso fisico al dispositivo.

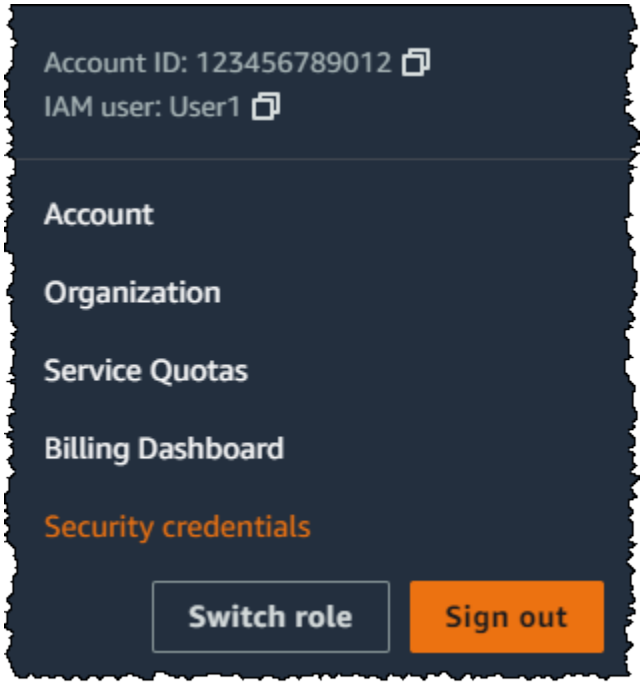
Per abilitare una passkey o una chiave di sicurezza per il proprio utente IAM (console)

1. Usa l'ID o l'alias dell'account, il tuo nome utente IAM e la tua password per accedere alla console IAM. AWS

> **Note**
>
> Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias del tuo AWS account per essere reindirizzato alla pagina di accesso utente IAM del tuo account.

Per ottenere il tuo Account AWS ID, contatta il tuo amministratore.

2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella pagina dell'utente IAM selezionato, scegli la scheda Credenziali di sicurezza.
4. In Multi-factor autenticazione (MFA), scegli Assegna dispositivo MFA.
5. Nella pagina Nome del dispositivo MFA, inserisci un nome per il dispositivo, scegli Passkey o chiave di sicurezza, quindi scegli Avanti.
6. In Configura dispositivo, configura la tua passkey. Crea una passkey con dati biometrici come il viso o l'impronta digitale, con un pin del dispositivo oppure inserendo la chiave di sicurezza FIDO nella porta USB del computer e toccandola.
7. Segui le istruzioni sul tuo browser e poi scegli Continua.

Ora hai registrato la tua passkey o chiave di sicurezza da utilizzare con. AWS Per informazioni sull'utilizzo dell'MFA con il Console di gestione AWS, vedere. Accesso abilitato con MFA

Abilitare una passkey o una chiave di sicurezza per un altro utente IAM (console)

Puoi abilitare una passkey o una chiave di sicurezza per un altro utente IAM Console di gestione AWS solo dall'API AWS CLI or AWS , non da.

Per abilitare una passkey o una chiave di sicurezza per un altro utente IAM (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. In Utenti, scegli il nome dell'utente per il quale desideri abilitare l'MFA.
4. Nella pagina dell'utente IAM selezionato, scegli la scheda Credenziali di sicurezza.
5. In Multi-factor autenticazione (MFA), scegli Assegna dispositivo MFA.
6. Nella pagina Nome del dispositivo MFA, inserisci un nome per il dispositivo, scegli Passkey o chiave di sicurezza, quindi scegli Avanti.
7. In Configura dispositivo, configura la tua passkey. Crea una passkey con dati biometrici come il viso o l'impronta digitale, con un pin del dispositivo oppure inserendo la chiave di sicurezza FIDO nella porta USB del computer e toccandola.
8. Segui le istruzioni sul tuo browser e poi scegli Continua.

Ora hai registrato una passkey o una chiave di sicurezza per un altro utente IAM da utilizzare con AWS. Per informazioni sull'utilizzo dell'MFA con, vedere. Console di gestione AWSAccesso abilitato con MFA

Sostituire una passkey o una chiave di sicurezza

Puoi avere fino a otto dispositivi MFA di qualsiasi combinazione dei tipi MFA attualmente supportati assegnati a un utente alla volta con i tuoi utenti Utente root dell'account AWS e quelli IAM. Se l'utente dovesse perdere l'autenticatore FIDO o in caso di sostituzione, dovrai prima disattivare il vecchio autenticatore FIDO. e quindi aggiungere un nuovo dispositivo MFA.

- Per disattivare il dispositivo correntemente associato a un utente IAM, consulta Disattivare un dispositivo MFA.
- Per aggiungere una nuova chiave di sicurezza FIDO per un utente IAM, consulta la sezione Abilitare una passkey o una chiave di sicurezza per il proprio utente IAM (console).

Se non hai accesso a una nuova passkey o a una chiave di sicurezza, puoi abilitare un nuovo dispositivo MFA virtuale o un token TOTP hardware. Per istruzioni, consulta uno dei seguenti articoli:

- Assegna un dispositivo MFA virtuale nel Console di gestione AWS
- Assegna un token TOTP hardware nel Console di gestione AWS

Configurazioni supportate per l'uso delle passkey e delle chiavi di sicurezza

Puoi utilizzare le passkey legate al dispositivo FIDO2, note anche come chiavi di sicurezza, come metodo di autenticazione a più fattori (MFA) con IAM tramite le configurazioni attualmente supportate. Sono inclusi i dispositivi FIDO2 supportati da IAM e i browser che supportano FIDO2.

Prima di registrare il dispositivo FIDO2, verifica di utilizzare la versione più recente del browser e del sistema operativo (OS). Le funzionalità potrebbero comportarsi diversamente nei diversi browser, autenticatori e client del sistema operativo. Se la registrazione del dispositivo non riesce su un browser, puoi provare a registrarti con un altro browser.

FIDO2 è uno standard di autenticazione aperto e un'estensione di FIDO U2F che offre lo stesso livello elevato di sicurezza basato sulla crittografia a chiave pubblica. FIDO2 è costituito dalla specifica W3C Web Authentication (WebAuthn API) e dal FIDO Alliance Client-to-Authenticator Protocol (CTAP), un protocollo a livello di applicazione. CTAP consente la comunicazione tra client o piattaforma, come un browser o un sistema operativo, con un autenticatore esterno.

Quando si abilita un autenticatore certificato FIDO in AWS, la chiave di sicurezza crea una nuova coppia di chiavi utilizzabile solo con. AWS In primo luogo, è necessario immettere le credenziali. Quando richiesto, tocca la chiave di sicurezza, che risponde alla richiesta di autenticazione emessa da AWS. Per ulteriori informazioni sullo standard FIDO2, consulta la pagina Progetto FIDO2.

Dispositivi FIDO2 supportati da AWS

IAM supporta dispositivi di sicurezza FIDO2 che si connettono ai dispositivi tramite USB, Bluetooth o NFC. IAM supporta anche gli autenticatori di piattaforma come TouchID o FaceID. IAM non supporta la registrazione locale delle passkey per Windows Hello. Per creare e utilizzare le passkey, gli utenti Windows devono utilizzare l'autenticazione tra dispositivi, che prevede l'utilizzo di una passkey di un dispositivo, ad esempio un dispositivo mobile, o di una chiave di sicurezza hardware per accedere su un altro dispositivo, ad esempio un laptop.

> **Note**
>
> AWS richiede l'accesso alla porta USB fisica del computer per verificare il dispositivo FIDO2. Le chiavi di sicurezza non funzioneranno con una macchina virtuale, una connessione remota o la modalità in incognito di un browser.

FIDO Alliance mantiene un elenco di tutti i prodotti FIDO2 compatibili con le specifiche FIDO.

Browser che supportano FIDO2

La disponibilità dei dispositivi di sicurezza FIDO2 che funzionano in un browser Web dipende dalla combinazione di browser e sistema operativo. Al momento i seguenti browser supportano l'uso delle chiavi di sicurezza:

Browser	macOS 10.15+	Windows 10	Linux	iOS 14.5+	Android 7+
Chrome	Sì	Sì	Sì	Sì	No
Safari	Sì	No	No	Sì	No
Edge	Sì	Sì	No	Sì	No
Firefox	Sì	Sì	No	Sì	No

> **Note**
>
> Per la maggior parte delle versioni di Firefox che attualmente supportano FIDO2, il supporto non è abilitato per impostazione predefinita. Per le istruzioni per abilitare il supporto FIDO2 in Firefox, consulta Risoluzione dei problemi relativi alle passkey e alle chiavi di sicurezza FIDO. Firefox su macOS potrebbe non supportare completamente i flussi di lavoro di autenticazione tra dispositivi per le password. È possibile che ti venga richiesto di toccare una chiave di sicurezza invece di procedere con l'autenticazione su più dispositivi. Per accedere con passkey su macOS, consigliamo di utilizzare un browser diverso, come Chrome o Safari.

Per ulteriori informazioni sul supporto del browser per FIDO2-Certified dispositivi come YubiKey, ad esempio, vedi Supporto del sistema operativo e del browser web per FIDO2 e U2F.

Plug-in del browser

AWS supporta solo i browser che supportano nativamente FIDO2. AWS non supporta l'utilizzo di plugin per aggiungere il supporto per il browser FIDO2. Alcuni plug-in del browser non sono compatibili con lo standard FIDO e possono causare risultati imprevisti se utilizzati con le chiavi di sicurezza FIDO2.

Per informazioni su come disabilitare i plug-in del browser e altri suggerimenti per la risoluzione dei problemi, consulta Non riesco ad abilitare la chiave di sicurezza FIDO.

Certificazioni dei dispositivi

Acquisiamo e assegniamo le certificazioni relative ai dispositivi, come la convalida FIPS e il livello di certificazione FIDO, solo durante la registrazione di una chiave di sicurezza. La certificazione del dispositivo viene recuperata dal FIDO Alliance Metadata Service (MDS).

Se lo stato o il livello di certificazione della chiave di sicurezza cambia, ciò non si rifletterà automaticamente nei tag del dispositivo. Per aggiornare le informazioni di certificazione di un dispositivo, è necessario registrarlo nuovamente per recuperare le informazioni di certificazione aggiornate.

AWS fornisce i seguenti tipi di certificazione come chiavi di condizione durante la registrazione del dispositivo, ottenute dai livelli di certificazione FIDO MDS: FIPS-140-2 FIPS-140-3, e FIDO. È possibile limitare la registrazione dell'autenticatore nelle politiche IAM in base al tipo e al livello di certificazione. Per ulteriori informazioni, consulta le policy seguenti.

Policy di esempio per le certificazioni dei dispositivi

I seguenti casi d'uso mostrano policy di esempio che consentono di registrare i dispositivi MFA con certificazioni FIPS.

Argomenti

- Caso d'uso 1: consenti la registrazione solo dei dispositivi con certificazioni L2 FIPS-140-2
- Caso d'uso 2: consentire la registrazione di dispositivi con certificazioni L2 e FIDO FIPS-140-2 L1
- Caso d'uso 3: consentire la registrazione di dispositivi con certificazioni L2 o L2 FIPS-140-2 FIPS-140-3
- Caso d'uso 4: consenti la registrazione di dispositivi con certificazione FIPS-140-2 L2 e supportano altri tipi di MFA come autenticatori virtuali e TOTP hardware

Caso d'uso 1: consenti la registrazione solo dei dispositivi con certificazioni L2 FIPS-140-2

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-2-certification": "L2"
                }
            }
        }
    ]
}
```

Caso d'uso 2: consentire la registrazione di dispositivi con certificazioni L2 e FIDO FIPS-140-2 L1

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-2-certification": "L2",
                    "iam:FIDO-certification": "L1"
                }
            }
        }
    ]
}
```

Caso d'uso 3: consentire la registrazione di dispositivi con certificazioni L2 o L2 FIPS-140-2 FIPS-140-3

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-2-certification": "L2"
                }
            }
        },
```

```
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-3-certification": "L2"
                }
            }
        }
    ]
}
```

Caso d'uso 4: consenti la registrazione di dispositivi con certificazione FIPS-140-2 L2 e supportano altri tipi di MFA come autenticatori virtuali e TOTP hardware

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:EnableMFADevice",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:RegisterSecurityKey": "Create"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:EnableMFADevice",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:RegisterSecurityKey": "Activate",
          "iam:FIDO-FIPS-140-2-certification": "L2"
        }
```

```
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:EnableMFADevice",
      "Resource": "*",
      "Condition": {
        "Null": {
          "iam:RegisterSecurityKey": "true"
        }
      }
    }
  ]
}
```

AWS CLI and AWS "Hello, World!"

AWS supporta l'utilizzo di password e chiavi di sicurezza solo in. Console di gestione AWS L'utilizzo di password e chiavi di sicurezza per l'MFA non è supportato nell'AWS API AWS CLI and o per l'accesso alle MFA-protected operazioni API.

Risorse aggiuntive

- Per ulteriori informazioni sull'utilizzo di password e chiavi di sicurezza in AWS, consulta. Assegna una passkey o una chiave di sicurezza nel Console di gestione AWS
- Per assistenza nella risoluzione dei problemi relativi alle chiavi di accesso e alle chiavi di sicurezza in AWS, vedere. Risoluzione dei problemi relativi alle passkey e alle chiavi di sicurezza FIDO
- Per informazioni generiche sul supporto FIDO2, consulta la pagina del Progetto FIDO2.

Assegna un dispositivo MFA virtuale nel Console di gestione AWS

> **Important**
>
> AWS consiglia di utilizzare una passkey o una chiave di sicurezza per l'MFA AWS, laddove possibile. Per ulteriori informazioni, consulta Assegna una passkey o una chiave di sicurezza nel Console di gestione AWS.

Puoi utilizzare un telefono o un altro dispositivo come dispositivo di autenticazione a più fattori (MFA) virtuale. A tale scopo, installa un'app mobile conforme a RFC 6238, un algoritmo TOTP (password monouso temporanea) basato su standard. Queste app generano un codice di autenticazione a sei cifre. Poiché gli autenticatori possono essere eseguiti su dispositivi mobili non protetti e i codici potrebbero essere condivisi con parti non autorizzate, l' TOTP-based MFA non offre lo stesso livello di sicurezza delle opzioni resistenti al phishing come le chiavi di sicurezza e le passkey FIDO2. https://en.wikipedia.org/wiki/FIDO_Alliance#FIDO2 Ti consigliamo di utilizzare passkey o chiavi di sicurezza per l'MFA per una protezione più efficace contro attacchi come il phishing.

Se non sei ancora in grado di utilizzare password o chiavi di sicurezza, ti consigliamo di utilizzare un dispositivo MFA virtuale come misura provvisoria in attesa dell'approvazione dell'acquisto dell'hardware o dell'arrivo dell'hardware.

La maggior parte delle app MFA virtuali supporta la creazione di più dispositivi virtuali, consentendo di utilizzare la stessa app per più utenti. Account AWS Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione di tipi MFA con i tuoi utenti Utente root dell'account AWS e quelli IAM. È necessario un solo dispositivo MFA per accedere Console di gestione AWS o creare una sessione tramite. AWS CLI Consigliamo di registrare più dispositivi MFA. Per le applicazioni di autenticazione, consigliamo inoltre di abilitare le funzionalità di backup o sincronizzazione su cloud per evitare di perdere l'accesso all'account in caso di smarrimento o guasto del dispositivo che dispone delle app di autenticazione.

AWS richiede un'app MFA virtuale che produca un OTP a sei cifre. Per un elenco delle app MFA virtuali che puoi usare, vedi Autenticazione. Multi-Factor

Argomenti

- Autorizzazioni richieste
- Abilitazione di un dispositivo MFA virtuale per un utente IAM (console)
- Sostituzione di un dispositivo MFA virtuale

Autorizzazioni richieste

Per gestire i dispositivi MFA virtuali per l'utente IAM, è necessario disporre delle autorizzazioni dalla policy seguente: AWS: consente agli utenti IAM autenticati con MFA di gestire il proprio dispositivo MFA nella pagina Credenziali di sicurezza.

Abilitazione di un dispositivo MFA virtuale per un utente IAM (console)

Puoi utilizzare IAM Console di gestione AWS per abilitare e gestire un dispositivo MFA virtuale per un utente IAM nel tuo account. È possibile collegare tag alle risorse IAM, inclusi i dispositivi MFA virtuali, per identificare, organizzare e controllare l'accesso a tali risorse. Puoi etichettare i dispositivi MFA virtuali solo quando usi l'API AWS CLI or AWS . Per abilitare e gestire un dispositivo MFA utilizzando l' AWS API AWS CLI or, consulta. Assegna dispositivi MFA nell' AWS CLI API o AWS Per ulteriori informazioni sul tagging delle risorse IAM, consulta Tag per AWS Identity and Access Management risorse.

> **Note**
>
> È necessario avere l'accesso fisico all'hardware che ospiterà il dispositivo MFA virtuale dell'utente per configurare MFA. Ad esempio, è possibile configurare MFA per un utente che utilizzerà un dispositivo MFA virtuale in esecuzione su uno smartphone. In questo caso, è necessario disporre di uno smartphone per completare la procedura guidata. Per questo motivo, è possibile consentire agli utenti di configurare e gestire i propri dispositivi MFA virtuali. In questo caso è necessario concedere agli utenti le autorizzazioni per eseguire le necessarie operazioni IAM. Per ulteriori informazioni e per un esempio di una policy IAM che concede queste autorizzazioni, consulta Tutorial IAM: consentire agli utenti di gestire le proprie credenziali e impostazioni MFA e la policy di esempio AWS: consente agli utenti IAM autenticati con MFA di gestire il proprio dispositivo MFA nella pagina Credenziali di sicurezza.

Come abilitare un dispositivo MFA virtuale per un utente IAM (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Nell'elenco degli Utenti, seleziona il nome dell'utente IAM.
4. Seleziona la scheda Credenziali di sicurezza. In Multi-factor autenticazione (MFA), scegli Assegna dispositivo MFA.
5. Nella procedura guidata, digita un nome per il dispositivo, scegli l'app Authenticator e quindi scegli Next (Avanti).

IAM genera e visualizza le informazioni di configurazione per il dispositivo MFA virtuale, tra cui il codice grafico QR. Il grafico è una rappresentazione della "chiave di configurazione segreta" disponibile per l'inserimento manuale sui dispositivi che non supportano i codici QR.

6. Aprire l'app MFA virtuale. Per un elenco di app che puoi utilizzare per ospitare dispositivi MFA virtuali, vedi Autenticazione. Multi-Factor

   Se l'app MFA virtuale supporta più account o dispositivi MFA virtuali, selezionare l'opzione che consente di creare un nuovo account o dispositivo MFA virtuale.

7. Determinare se l'app MFA supporta i codici QR e procedere in uno dei seguenti modi:

   - Nella procedura guidata, scegliere Mostra codice QR ed eseguire la scansione del codice QR tramite l'app. Potrebbe trattarsi dell'icona della fotocamera o dell'opzione Scannerizza codice che utilizza la fotocamera del dispositivo per eseguire la scansione del codice.
   - Nella procedura guidata, scegli Show secret key (Mostra chiave segreta) e digitare la chiave segreta nell'app MFA.

   Al termine, il dispositivo MFA virtuale avvia la generazione di password una tantum.

8. Nella pagina Set up device (Configura il dispositivo), nella casella MFA code 1 (Codice MFA 1), digita la password monouso correntemente visualizzata nel dispositivo MFA virtuale. Attendere fino a un massimo di 30 secondi prima che il dispositivo generi una nuova password una tantum. Quindi, digitare la seconda password monouso nella casella Codice MFA 2. Scegli Aggiungi MFA.

   > **Important**
   >
   > Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, il dispositivo MFA si associa correttamente con l'utente ma il dispositivo MFA non viene sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo MFA virtuale è ora pronto per l'uso con. AWS Per informazioni sull'utilizzo dell'MFA con Console di gestione AWS, vedere. Accesso abilitato con MFA

> **Note**
>
> I dispositivi MFA virtuali non assegnati all'interno dell'utente Account AWS vengono eliminati quando si aggiungono nuovi dispositivi MFA virtuali tramite Console di gestione AWS o durante il processo di accesso. I dispositivi MFA virtuali non assegnati sono dispositivi presenti nell'account ma non utilizzati dall'utente root dell'account o dagli utenti IAM per il processo di accesso. Vengono eliminati in modo da poter aggiungere nuovi dispositivi MFA virtuali al tuo account. Questo consente anche di riutilizzare i nomi dei dispositivi.
>
> - Per visualizzare i dispositivi MFA virtuali non assegnati nel tuo account, puoi utilizzare il comando list-virtual-mfa-devices o la chiamata API. AWS CLI https://docs.aws.amazon.com/IAM/latest/APIReference/API_ListVirtualMFADevices.html
> - Per disattivare un dispositivo MFA virtuale, puoi utilizzare il comando deactivate-mfa-device o la chiamata API. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/deactivate-mfa-device.html AWS CLI https://docs.aws.amazon.com/IAM/latest/APIReference/API_DeactivateMFADevice.html Verrà annullata l'assegnazione del dispositivo.
> - Per collegare un dispositivo MFA virtuale non assegnato all'utente Account AWS root o agli utenti IAM, avrai bisogno del codice di autenticazione generato dal dispositivo insieme al comando enable-mfa-device o alla chiamata API. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/enable-mfa-device.html AWS CLI https://docs.aws.amazon.com/IAM/latest/APIReference/API_EnableMFADevice.html

Sostituzione di un dispositivo MFA virtuale

Il tuo utente Utente root dell'account AWS e quello di IAM possono registrare fino a otto dispositivi MFA di qualsiasi combinazione di tipi MFA. Se l'utente dovesse perdere il proprio dispositivo o in caso di sostituzione, dovrai disattivare il vecchio dispositivo. e quindi aggiungere quello nuovo.

- Per disattivare il dispositivo correntemente associato a un altro utente IAM, consulta Disattivare un dispositivo MFA.
- Per aggiungere un dispositivo MFA virtuale sostitutivo per un altro utente IAM, segui la procedura Abilitazione di un dispositivo MFA virtuale per un utente IAM (console) descritta sopra.
- Per aggiungere un dispositivo MFA virtuale sostitutivo per Utente root dell'account AWS, segui i passaggi della procedura. Abilitare un dispositivo MFA virtuale per l'utente root (console)

Assegna un token TOTP hardware nel Console di gestione AWS

> **Important**
>
> AWS consiglia di utilizzare una passkey o una chiave di sicurezza per l'MFA, ove possibile. AWS Per ulteriori informazioni, consulta Assegna una passkey o una chiave di sicurezza nel Console di gestione AWS.

Un dispositivo MFA hardware genera un codice numerico di sei cifre basato su un algoritmo di password monouso sincronizzato nel tempo. L'utente deve immettere un codice valido dal dispositivo quando richiesto durante la procedura di accesso. Ogni dispositivo MFA assegnato a un utente deve essere univoco; un utente non può immettere un codice dal dispositivo di un altro utente per effettuare l'autenticazione. I dispositivi MFA non possono essere condivisi tra account o utenti.

I dispositivi MFA hardware e le chiavi di sicurezza FIDO sono entrambi dispositivi fisici che si acquistano. I dispositivi hardware MFA generano codici TOTP per l'autenticazione al momento dell'accesso. AWS Si basano sulle batterie, che potrebbero dover essere sostituite e risincronizzate nel tempo. AWS Le chiavi di sicurezza FIDO, che utilizzano la crittografia a chiave pubblica, non richiedono batterie e offrono un processo di autenticazione senza interruzioni. Consigliamo di utilizzare le chiavi di sicurezza FIDO per la loro resistenza al phishing e perché forniscono un'alternativa più sicura ai dispositivi TOTP. Inoltre, le chiavi di sicurezza FIDO possono supportare più utenti IAM o root sullo stesso dispositivo, migliorandone l'utilità per la sicurezza degli account. Per le specifiche e le informazioni di acquisto per entrambi i tipi di dispositivo, vedi Autenticazione. Multi-Factor

Puoi abilitare un token TOTP hardware per un utente IAM dalla Console di gestione AWS riga di comando o dall'API IAM. Per abilitare un dispositivo MFA per te Utente root dell'account AWS, consulta. Abilitazione di un token TOTP hardware per l'utente root dell' (console)

Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi MFA attualmente supportati con i tuoi utenti Utente root dell'account AWS e quelli IAM. Con più dispositivi MFA, è necessario un solo dispositivo MFA per accedere Console di gestione AWS o creare una sessione tramite tale utente. AWS CLI

> **Important**
>
> Ti consigliamo di abilitare più dispositivi MFA per gli utenti in modo da garantire l'accesso continuo al tuo account in caso di smarrimento del dispositivo MFA o se diventa inaccessibile.

> **Note**
>
> Per abilitare il dispositivo MFA dalla riga di comando, utilizzare `aws iam enable-mfa-device`. Per abilitare il dispositivo MFA con l'API IAM, utilizza l'operazione `EnableMFADevice`.

Argomenti

- Autorizzazioni richieste
- Abilitazione di un dispositivo MFA hardware per un utente IAM (console)
- Abilitazione di un dispositivo MFA hardware per un altro utente IAM (console)
- Sostituzione di un dispositivo MFA fisico

Autorizzazioni richieste

Per gestire un token TOTP hardware per il tuo utente IAM proteggendo al contempo MFA-related le azioni sensibili, devi disporre delle autorizzazioni previste dalla seguente politica:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowManageOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
```

```
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "DenyAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
    ]
}
```

Abilitazione di un dispositivo MFA hardware per un utente IAM (console)

È possibile abilitare il proprio dispositivo MFA hardware tramite Console di gestione AWS.

> **Note**
>
> Prima di abilitare un dispositivo MFA hardware è necessario disporre di accesso fisico al dispositivo.

Come abilitare un dispositivo MFA hardware per un utente IAM (console)

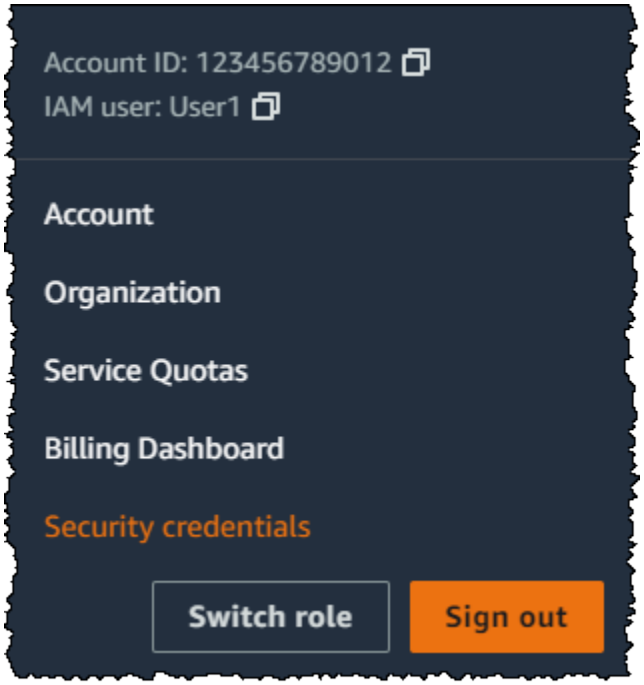
1. Utilizza l' AWS ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM. https://console.aws.amazon.com/iam

   > **Note**
   >
   > Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito

> l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias del tuo AWS account per essere reindirizzato alla pagina di accesso utente IAM del tuo account.

Per ottenere il tuo Account AWS ID, contatta il tuo amministratore.

2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella scheda Credenziali AWS IAM, nella sezione Multi-factor autenticazione (MFA), scegli Assegna dispositivo MFA.
4. Nella procedura guidata, digitate il nome del dispositivo, scegliete il token Hardware TOTP e quindi scegliete Avanti.
5. Digitare il numero di serie del dispositivo. In genere, il numero di serie è indicato sulla parte posteriore del dispositivo.
6. Nella casella MFA code 1 (Codice MFA 1) digitare il numero di sei cifre visualizzato nel dispositivo MFA. Per visualizzare il numero, potrebbe essere necessario premere il pulsante sul lato anteriore del dispositivo.



7. Attendere 30 secondi per consentire al dispositivo di aggiornare il codice, quindi digitare il nuovo numero a sei cifre nella casella MFA code 2 (Codice MFA 2). Per visualizzare il secondo numero, potrebbe essere necessario premere nuovamente il pulsante sul lato anteriore del dispositivo.
8. Scegli Aggiungi MFA.

> **Important**
>
> La richiesta deve essere inviata immediatamente dopo la generazione dei codici di autenticazione. Se dopo avere generato i codici attendi troppo a lungo prima di inviare la richiesta, il dispositivo MFA si associerà correttamente con l'utente, ma perderà la sincronizzazione. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo è pronto per l'uso con AWS. Per ulteriori informazioni sull'utilizzo di MFA con la Console di gestione AWS, consulta Accesso abilitato con MFA.

Abilitazione di un dispositivo MFA hardware per un altro utente IAM (console)

Puoi abilitare un dispositivo MFA hardware per un altro utente IAM dalla Console di gestione AWS.

Come abilitare un dispositivo MFA hardware per un altro utente IAM (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Scegli il nome del segreto per il quale desideri abilitare la rotazione.
4. Seleziona la scheda Credenziali di sicurezza. In Multi-factor autenticazione (MFA), scegli Assegna dispositivo MFA.
5. Nella procedura guidata, digitate il nome del dispositivo, scegliete il token Hardware TOTP e quindi scegliete Avanti.
6. Digitare il numero di serie del dispositivo. In genere, il numero di serie è indicato sulla parte posteriore del dispositivo.

7. Nella casella MFA code 1 (Codice MFA 1) digitare il numero di sei cifre visualizzato nel dispositivo MFA. Per visualizzare il numero, potrebbe essere necessario premere il pulsante sul lato anteriore del dispositivo.



8. Attendere 30 secondi per consentire al dispositivo di aggiornare il codice, quindi digitare il nuovo numero a sei cifre nella casella MFA code 2 (Codice MFA 2). Per visualizzare il secondo numero, potrebbe essere necessario premere nuovamente il pulsante sul lato anteriore del dispositivo.
9. Scegli Aggiungi MFA.

> **Important**
>
> La richiesta deve essere inviata immediatamente dopo la generazione dei codici di autenticazione. Se dopo avere generato i codici attendi troppo a lungo prima di inviare la richiesta, il dispositivo MFA si associerà correttamente con l'utente, ma perderà la sincronizzazione. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo è pronto per l'uso con AWS. Per ulteriori informazioni sull'utilizzo di MFA con la Console di gestione AWS, consulta Accesso abilitato con MFA.

Sostituzione di un dispositivo MFA fisico

Puoi avere fino a otto dispositivi MFA di qualsiasi combinazione dei tipi MFA attualmente supportati assegnati a un utente alla volta con i tuoi utenti Utente root dell'account AWS e quelli IAM. Se l'utente dovesse perdere il proprio dispositivo o in caso di sostituzione, dovrai disattivare il vecchio dispositivo e quindi aggiungere quello nuovo.

- Per disattivare il dispositivo associato al momento con un utente, consultare Disattivare un dispositivo MFA.
- Per aggiungere un dispositivo MFA hardware sostitutivo per un utente IAM, segui la procedura Abilitazione di un dispositivo MFA hardware per un altro utente IAM (console)descritta prima in questo argomento.

- Per aggiungere un token TOTP hardware sostitutivo per Utente root dell'account AWS, segui i passaggi della procedura Abilitazione di un token TOTP hardware per l'utente root dell' (console) precedente in questo argomento.

Assegna dispositivi MFA nell' AWS CLI API o AWS

Puoi utilizzare AWS CLI comandi o operazioni AWS API per abilitare un dispositivo MFA virtuale per un utente IAM. Non è possibile abilitare un dispositivo MFA per il Utente root dell'account AWS AWS CLI, AWS API, Tools for Windows PowerShell o qualsiasi altro strumento da riga di comando. Tuttavia, è possibile utilizzare il Console di gestione AWS per abilitare un dispositivo MFA per l'utente root.

Quando abiliti un dispositivo MFA da Console di gestione AWS, la console esegue più passaggi per te. Se invece crei AWS CLI un dispositivo virtuale utilizzando Tools for Windows PowerShell o AWS API, devi eseguire i passaggi manualmente e nell'ordine corretto. Ad esempio, per creare un dispositivo MFA virtuale, è necessario creare l'oggetto IAM ed estrarre il codice sotto forma di stringa o di codice grafico QR. Quindi è necessario sincronizzare il dispositivo e associarlo a un utente IAM. Vedi la sezione Esempi di New- IAMVirtual MFADevice per maggiori dettagli. Per un dispositivo fisico, si salta la fase di creazione per andare direttamente a sincronizzare il dispositivo e associarlo con l'utente.

È possibile collegare tag alle risorse IAM, inclusi i dispositivi MFA virtuali, per identificare, organizzare e controllare l'accesso a tali risorse. È possibile etichettare i dispositivi MFA virtuali solo quando si utilizza l'API AWS CLI o AWS .

Un utente IAM che utilizza l'SDK o la CLI può abilitare un dispositivo MFA aggiuntivo chiamando `EnableMFADevice` o disattivarlo chiamando `DeactivateMFADevice`. Per eseguire correttamente questa operazione, devono prima chiamare `GetSessionToken` e inviare i codici MFA con un dispositivo MFA esistente. Questa chiamata restituisce credenziali di sicurezza temporanee che possono quindi essere utilizzate per firmare operazioni API che richiedono l'autenticazione MFA. Per un esempio di richiesta e risposta, consulta `GetSessionToken`: credenziali temporanee per gli utenti in ambienti non attendibili.

Creazione dell'entità del dispositivo virtuale in IAM in modo che rappresenti un dispositivo MFA virtuale

Questi comandi forniscono un ARN per il dispositivo che viene usato al posto di un numero di serie in molti dei seguenti comandi.

- AWS CLI: `aws iam create-virtual-mfa-device`
- AWS API: `CreateVirtualMFADevice`

Per abilitare un dispositivo MFA da utilizzare con AWS

Questi comandi sincronizzano il dispositivo AWS e lo associano a un utente. Se il dispositivo è virtuale, utilizzare l'ARN di un dispositivo virtuale come numero di serie.

> **Important**
>
> La richiesta deve essere inviata immediatamente dopo la generazione dei codici di autenticazione. Se dopo avere generato i codici attendi troppo a lungo prima di inviare la richiesta, il dispositivo MFA si associerà correttamente con l'utente, ma perderà la sincronizzazione. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. In questo caso, è possibile risincronizzare il dispositivo utilizzando i comandi descritti di seguito.

- AWS CLI: `aws iam enable-mfa-device`
- AWS API: `EnableMFADevice`

Per disattivare un dispositivo

Utilizzare questi comandi per dissociare il dispositivo dall'utente e disattivarlo. Se il dispositivo è virtuale, utilizzare l'ARN di un dispositivo virtuale come numero di serie. È inoltre necessario eliminare separatamente l'entità del dispositivo virtuale.

- AWS CLI: `aws iam deactivate-mfa-device`
- AWS API: `DeactivateMFADevice`

Elenco delle entità del dispositivo MFA virtuale

Utilizzare questi comandi per elencare le entità di un dispositivo MFA virtuale.

- AWS CLI: `aws iam list-virtual-mfa-devices`
- AWS API: `ListVirtualMFADevices`

Per applicare tag a un dispositivo MFA virtuale

Utilizzare questi comandi per applicare tag a un dispositivo MFA virtuale.

- AWS CLI: `aws iam tag-mfa-device`
- AWS API: `TagMFADevice`

Per elencare i tag per un dispositivo MFA virtuale

Utilizzare questi comandi per elencare i tag collegati a un dispositivo MFA virtuale.

- AWS CLI: `aws iam list-mfa-device-tags`
- AWS API: `ListMFADeviceTags`

Per rimuovere i tag da un dispositivo MFA virtuale

Utilizzare questi comandi per rimuovere i tag collegati a un dispositivo MFA virtuale.

- AWS CLI: `aws iam untag-mfa-device`
- AWS API: `UntagMFADevice`

Risincronizzare un dispositivo MFA

Usa questi comandi se il dispositivo genera codici che non sono accettati da AWS. Se il dispositivo è virtuale, utilizzare l'ARN di un dispositivo virtuale come numero di serie.

- AWS CLI: `aws iam resync-mfa-device`
- AWS API: `ResyncMFADevice`

Come eliminare un'entità del dispositivo MFA virtuale in IAM

Dopo che il dispositivo è stato dissociato da parte dell'utente, è possibile eliminare l'entità del dispositivo.

- AWS CLI: `aws iam delete-virtual-mfa-device`
- AWS API: `DeleteVirtualMFADevice`

Per recuperare un dispositivo MFA virtuale che è stato smarrito o non funziona

Potrebbe accadere che il dispositivo di un utente che ospita l'app MFA virtuale venga smarrito o sostituito o non funzioni. Quando ciò si verifica, l'utente non può recuperarlo autonomamente. L'utente deve contattare un amministratore per disattivare il dispositivo. Per ulteriori informazioni, consulta Recuperare un'identità protetta da MFA in IAM.

Verifica dello stato MFA

Utilizza la console IAM per verificare se un utente Utente root dell'account AWS o un utente IAM ha abilitato un dispositivo MFA valido.

Come verificare lo stato MFA di un utente root

1. Accedi a Console di gestione AWS con le tue credenziali utente root, quindi apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).
3. Controlla in Multi-factor Autenticazione (MFA) per vedere se l'MFA è abilitata o disabilitata. Se l'autenticazione MFA non è abilitata, viene visualizzato un simbolo di avviso ( ).

Per abilitare l'autenticazione MFA per l'account, consultare uno dei seguenti articoli:

- Abilitare un dispositivo MFA virtuale per l'utente root (console)
- Abilitare una passkey o una chiave di sicurezza per l'utente root (console)
- Abilitazione di un token TOTP hardware per l'utente root dell' (console)

Come verificare lo stato MFA degli utenti IAM

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Se necessario, aggiungere la colonna MFA alla tabella degli utenti mediante la procedura seguente:
   a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ( ).
   b. In Manage Columns (Gestisci colonne) selezionare MFA.

c. (Facoltativo) Deselezionare la casella di controllo per le intestazioni di colonna che non si desidera visualizzare nella tabella utenti.

d. Selezionare Close (Chiudi) per tornare all'elenco degli utenti.

4. La colonna MFA fornisce informazioni sul dispositivo MFA abilitato. Se per l'utente non è attivo alcun dispositivo MFA, la console visualizza None (Nessuno). Se l'utente dispone di un dispositivo MFA abilitato, la colonna MFA mostra il tipo di dispositivo abilitato con il valore Virtual (Virtuale), Security Key (Chiave di sicurezza), Hardware o SMS.

> **Note**
>
> AWS è terminato il supporto per l'abilitazione dell'autenticazione a più fattori (MFA) tramite SMS. Consigliamo ai clienti con utenti IAM che utilizzano la MFA basata su messaggi di testo SMS di passare a uno dei seguenti metodi alternativi di autenticazione a più fattori: dispositivo MFA virtuale (basato su software), chiave di sicurezza FIDO o dispositivo MFA basato su hardware. Puoi identificare gli utenti nel tuo account con un dispositivo MFA SMS assegnato. Per farlo, vai alla console IAM, scegli Users (Utenti) dal riquadro di navigazione e individua gli utenti con SMS nella colonna della tabella MFA.

5. Per visualizzare ulteriori informazioni sul dispositivo MFA per un utente, selezionare il nome dell'utente di cui si desidera verificare lo stato MFA. Quindi, selezionare la scheda Security credentials (Credenziali di sicurezza).

6. Se per l'utente non è attivo alcun dispositivo MFA, la console visualizza No MFA devices (Nessun dispositivo MFA). Assegna un dispositivo MFA per migliorare la sicurezza del AWS tuo ambiente nella Multi-factor sezione di autenticazione (MFA). Se l'utente ha i dispositivi MFA abilitati, la sezione di Multi-factor autenticazione (MFA) mostra i dettagli sui dispositivi:

   - Il nome del dispositivo
   - Il tipo di dispositivo
   - L'identificatore del dispositivo, ad esempio un numero di serie per un dispositivo fisico o l'ingresso ARN AWS per un dispositivo virtuale
   - Quando il dispositivo è stato creato

Per rimuovere o risincronizzare un dispositivo, scegli il pulsante d'opzione accanto al dispositivo e scegli Remove (Rimuovi) o Resync (Risincronizza).

Per ulteriori informazioni sull'abilitazione di MFA, consultare quanto segue:

- Assegna un dispositivo MFA virtuale nel Console di gestione AWS
- Assegna una passkey o una chiave di sicurezza nel Console di gestione AWS
- Assegna un token TOTP hardware nel Console di gestione AWS

Risincronizzare i dispositivi MFA virtuali e hardware

Puoi utilizzarlo AWS per risincronizzare i tuoi dispositivi di autenticazione a più fattori (MFA) virtuali e hardware. Se il dispositivo dell'utente non è sincronizzato quando si tenta di utilizzarlo, il tentativo di accesso dell'utente non riesce e IAM richiede all'utente di risincronizzare il dispositivo.

> **Note**
>
> Le chiavi di sicurezza FIDO sono sempre sincronizzate. Se una chiave di sicurezza FIDO viene smarrita o danneggiata, puoi disattivarla. Per istruzioni sulla disattivazione dei dispositivi MFA, consulta Come disattivare un dispositivo MFA per un altro utente IAM (console).

In qualità di AWS amministratore, puoi risincronizzare i dispositivi MFA virtuali e hardware dei tuoi utenti IAM se la sincronizzazione non è più valida.

Se il dispositivo Utente root dell'account AWS MFA non funziona, puoi risincronizzarlo utilizzando la console IAM con o senza completare la procedura di accesso. Se non riesci a risincronizzare correttamente il dispositivo, potrebbe essere necessario dissociarlo e associarlo nuovamente. Per ulteriori informazioni su come effettuare tale operazione, consulta Disattivare un dispositivo MFA e AWS Multi-factor autenticazione in IAM.

Argomenti

- Autorizzazioni richieste
- Risincronizzazione dei dispositivi MFA virtuali e hardware (console IAM)
- Risincronizzazione dei dispositivi MFA virtuali e hardware (AWS CLI)
- Risincronizzazione dei dispositivi MFA virtuali e hardware (AWS API)

Autorizzazioni richieste

Per sincronizzare nuovamente i dispositivi MFA virtuali o hardware per l'utente IAM, è necessario disporre delle autorizzazioni dalla politica seguente. Questa politica non consente di creare o disattivare un dispositivo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListActions",
            "Effect": "Allow",
            "Action": [
                "iam:ListVirtualMFADevices"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUserToViewAndManageTheirOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "BlockAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:ListMFADevices",
                "iam:ListVirtualMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
```

```
    ]
}
```

Risincronizzazione dei dispositivi MFA virtuali e hardware (console IAM)

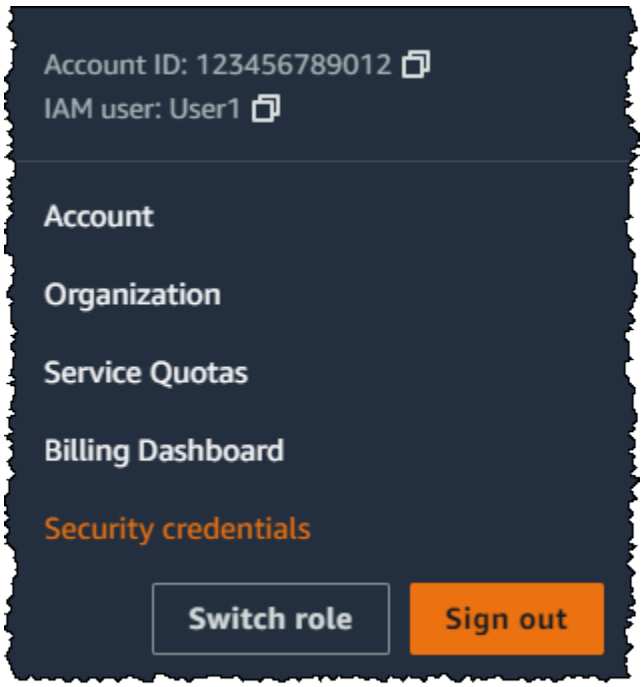
Puoi utilizzare la console IAM per risincronizzare i dispositivi MFA virtuali e hardware.

Come risincronizzare un dispositivo MFA virtuale o hardware per un utente IAM (console)

1. Usa l'ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM. AWS https://console.aws.amazon.com/iam

   > **Note**
   >
   > Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias del tuo AWS account per essere reindirizzato alla pagina di accesso utente IAM del tuo account.

   Per ottenere il tuo Account AWS ID, contatta il tuo amministratore.
2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella scheda Credenziali AWS IAM, nella sezione Multi-factor autenticazione (MFA), scegli il pulsante di opzione accanto al dispositivo MFA e scegli Resync.
4. Digitare i successivi due codici generati in sequenza dal dispositivo in MFA code 1 (Codice MFA 1) e MFA code 2 (Codice MFA 2). Quindi scegli Resync (Risincronizza).

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, la richiesta sembra riuscita ma il dispositivo non è comunque sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo.

Come risincronizzare un dispositivo MFA virtuale o hardware per un altro utente IAM (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione, selezionare Users (Utenti)e il nome dell'utente con dispositivo MFA da risincronizzare.
3. Selezionare la scheda Security Credentials (Credenziali di sicurezza). Nella sezione di Multi-factor autenticazione (MFA), scegli il pulsante di opzione accanto al dispositivo MFA e scegli Resync.

4. Digitare i successivi due codici generati in sequenza dal dispositivo in MFA code 1 (Codice MFA 1) e MFA code 2 (Codice MFA 2). Quindi scegli Resync (Risincronizza).

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, la richiesta sembra riuscita ma il dispositivo non è comunque sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo.

Come risincronizzare l'autenticazione MFA dell'utente root prima dell'accesso (console)

1. Nella pagina Accesso ad Amazon Web Services con dispositivo di autenticazione, seleziona Problemi con il tuo dispositivo di autenticazione? Fai clic qui.

> **Note**
>
> È possibile che il testo visualizzato sia differente, ad esempio Sign in using MFA (Accesso con un dispositivo MFA) e Troubleshoot your authentication device (Risoluzione dei problemi del dispositivo di autenticazione). Tuttavia, le funzionalità sono identiche.

2. Nella sezione Re-Sync With Our Servers, digita i due codici successivi generati in sequenza dal dispositivo nel codice MFA 1 e nel codice MFA 2. Quindi scegli Re-sync il dispositivo di autenticazione.
3. Se necessario, digitare di nuovo la password e selezionare Accedi. Quindi completare l'accesso utilizzando il dispositivo MFA.

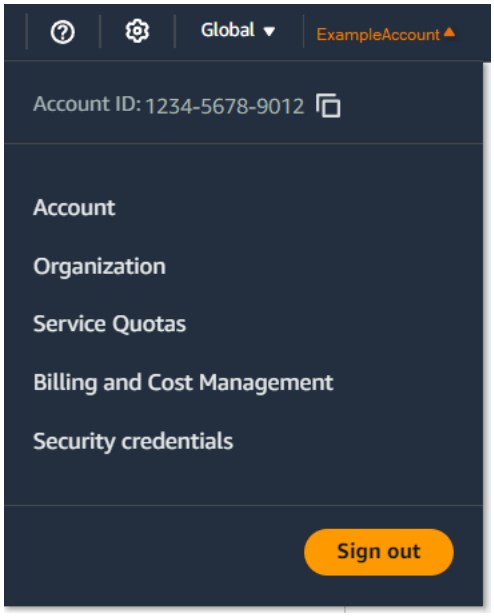
Come risincronizzare l'autenticazione MFA dell'utente root dopo l'accesso (console)

1. Accedi alla console IAM come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

> **Note**
>
> Come utente root, non puoi accedere alla pagina Sign in as IAM user (Accedi come utente IAM). Se viene visualizzata la pagina Sign in as IAM user (Accedi come utente

IAM), scegli Sign in using root user email (Accedi con l'indirizzo e-mail dell'utente root) nella parte inferiore della pagina. Per assistenza nell'accesso come utente root, consulta Accesso Console di gestione AWS come utente root nella Guida per l'Accedi ad AWS utente.

2. Sul lato destro della barra di navigazione, seleziona il nome dell'account, quindi Security Credentials (Credenziali di sicurezza). Se necessario, seleziona Continue to Security Credentials (Continua con le credenziali di sicurezza).



3. Espandi la sezione di Multi-factor autenticazione (MFA) nella pagina.
4. Seleziona il pulsante d'opzione accanto al dispositivo e scegli Resync (Risincronizza).
5. Nella finestra di dialogo Resync MFA device (Risincronizza dispositivo MFA), digita i successivi due codici generati in sequenza dal dispositivo in MFA code 1 (Codice MFA 1) e MFA code 2 (Codice MFA 2). Quindi scegli Resync (Risincronizza).

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si attende troppo a lungo per inviare la richiesta dopo la generazione dei codici, il dispositivo MFA verrà associato all'utente, ma non verrà sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo.

Risincronizzazione dei dispositivi MFA virtuali e hardware (AWS CLI)

Puoi risincronizzare i dispositivi MFA virtuali e hardware da AWS CLI.

Per risincronizzare un dispositivo MFA virtuale o hardware per un utente IAM (AWS CLI)

Al prompt dei comandi, emettere il comando aws iam resync-mfa-device:

- Dispositivo MFA virtuale: specificare l'Amazon Resource Name (ARN) del dispositivo come numero di serie.

```
aws iam resync-mfa-device --user-name Richard --serial-number
 arn:aws:iam::123456789012:mfa/RichardsMFA --authentication-code1 123456 --
authentication-code2 987654
```

- Dispositivo MFA hardware: specificare il numero di serie del dispositivo hardware come numero di serie. Il formato è specifico del fornitore. Ad esempio, puoi acquistare un token gemalto da Amazon. Il suo numero di serie è in genere quattro lettere seguite da quattro numeri.

```
aws iam resync-mfa-device --user-name Richard --serial-number ABCD12345678 --
authentication-code1 123456 --authentication-code2 987654
```

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo lungo per inviare la richiesta, la richiesta ha esito negativo perché i codici scadono dopo un breve periodo di tempo.

Risincronizzazione dei dispositivi MFA virtuali e hardware (AWS API)

In IAM è disponibile una chiamata API che esegue la sincronizzazione. In questo caso, ti consigliamo di fornire agli utenti del dispositivo MFA virtuale e hardware l'autorizzazione per accedere a questa chiamata API. Quindi, crea uno strumento basato su tale chiamata API per consentire agli utenti di risincronizzare i propri dispositivi ogni volta che è necessario.

Per risincronizzare un dispositivo MFA virtuale o hardware per un utente IAM (AWS API)

- Invia la richiesta ResyncMFADevice.

Disattivare un dispositivo MFA

In caso di problemi di accesso con un dispositivo con autenticazione a più fattori (MFA) come utente IAM, contatta il tuo amministratore per assistenza.

In qualità di amministratore, puoi disattivare il dispositivo per un altro utente IAM. In questo modo, l''utente potrà effettuare l'accesso senza utilizzare MFA. Questa soluzione può essere temporanea, se il dispositivo MFA è momentaneamente non disponibile o in attesa che venga sostituito. Tuttavia, ti consigliamo di abilitare un nuovo dispositivo per l'utente quanto prima possibile. Per informazioni su come abilitare un nuovo dispositivo MFA, consultare AWS Multi-factor autenticazione in IAM.

> **Note**
>
> Se utilizzi l'API o desideri AWS CLI eliminare un utente dal tuo Account AWS, devi disattivare o eliminare il dispositivo MFA dell'utente. Tale modifica fa paarte del processo di rimozione dell'utente. Per ulteriori informazioni sulla rimozione di utenti, consulta Rimuovere o disattivare un utente IAM.

Argomenti

- Disattivazione dei dispositivi MFA (console)
- Disattivazione dei dispositivi MFA (AWS CLI)
- Disattivazione dei dispositivi MFA (AWS API)

Disattivazione dei dispositivi MFA (console)

Come disattivare un dispositivo MFA per un altro utente IAM (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel pannello di navigazione, seleziona Utenti.
3. Per disattivare un dispositivo MFA, scegli il nome dell'utente a cui appartiene l'MFA da rimuovere.
4. Selezionare la scheda Security Credentials (Credenziali di sicurezza).
5. In Multi-factor Autenticazione (MFA), scegli il pulsante di opzione accanto al dispositivo MFA, scegli Rimuovi, quindi scegli Rimuovi.

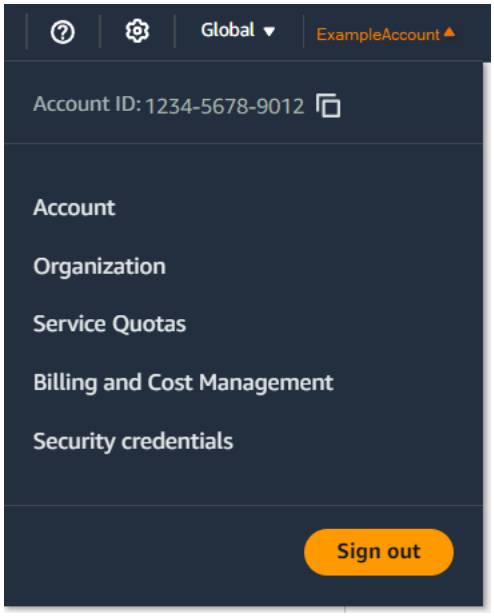
Il dispositivo viene rimosso da. AWS Non può essere utilizzato per accedere o autenticare le richieste finché non viene riattivato e associato a un AWS utente o. Utente root dell'account AWS

Per disattivare il dispositivo MFA per Utente root dell'account AWS (console)

1. Accedi alla console IAM come proprietario dell'account selezionando Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   > **Note**
   >
   > Come utente root, non puoi accedere alla pagina Sign in as IAM user (Accedi come utente IAM). Se viene visualizzata la pagina Sign in as IAM user (Accedi come utente IAM), scegli Sign in using root user email (Accedi con l'indirizzo e-mail dell'utente root) nella parte inferiore della pagina. Per informazioni sull'accesso come utente root, consulta Accedere Console di gestione AWS come utente root nella Guida per l'Accedi ad AWS utente.

2. Sul lato destro della barra di navigazione, seleziona il nome dell'account, quindi Security Credentials (Credenziali di sicurezza). Se necessario, seleziona Continue to Security Credentials (Continua con le credenziali di sicurezza).



3. Nella sezione Multi-factor Autenticazione (MFA), scegli il pulsante di opzione accanto al dispositivo MFA che desideri disattivare e scegli Rimuovi.
4. Scegli Rimuovi.

Il dispositivo MFA è disattivato per l' Account AWS. Controlla l'e-mail associata al tuo indirizzo Account AWS per ricevere un messaggio di conferma da Amazon Web Services. L'e-mail ti informa che la tua autenticazione multifattore (MFA) di Amazon Web Services è stata disattivata. Il messaggio verrà inviato da `@amazon.com` o `@aws.amazon.com`.

> **Note**
>
> I dispositivi MFA virtuali non assegnati nel Account AWS tuo vengono eliminati quando aggiungi nuovi dispositivi MFA virtuali tramite o durante Console di gestione AWS la procedura di accesso. I dispositivi MFA virtuali non assegnati sono dispositivi presenti nell'account ma non utilizzati dall'utente root dell'account o dagli utenti IAM per il processo di accesso. Vengono eliminati in modo da poter aggiungere nuovi dispositivi MFA virtuali al tuo account. Questo consente anche di riutilizzare i nomi dei dispositivi.

Disattivazione dei dispositivi MFA (AWS CLI)

Per disattivare un dispositivo MFA per un utente IAM (AWS CLI)

- Eseguire il comando: `aws iam deactivate-mfa-device`

Disattivazione dei dispositivi MFA (AWS API)

Per disattivare un dispositivo MFA per un utente IAM (AWS API)

- Richiamare l'operazione: `DeactivateMFADevice`

Recuperare un'identità protetta da MFA in IAM

Se il tuo dispositivo MFA virtuale o il token TOTP hardware sembra funzionare correttamente, ma non puoi utilizzarlo per accedere alle tue AWS risorse, potrebbe non essere sincronizzato con. AWS Per informazioni sulla sincronizzazione di un dispositivo MFA virtuale o hardware, consulta Risincronizzare i dispositivi MFA virtuali e hardware. Le chiavi di sicurezza FIDO sono sempre sincronizzate.

Se il dispositivo MFA per a Utente root dell'account AWS viene perso, danneggiato o non funziona, puoi recuperare l'accesso al tuo account. Gli utenti IAM devono contattare un amministratore per disattivare il dispositivo.

> **Important**
>
> Consigliamo di attivare più dispositivi MFA. La registrazione di più dispositivi MFA aiuta a garantire l'accesso continuo in caso di smarrimento o danneggiamento di un dispositivo. I tuoi utenti Utente root dell'account AWS e quelli di IAM possono registrare fino a otto dispositivi MFA di qualsiasi tipo.

Prerequisito: utilizzare un altro dispositivo MFA

Se il tuo dispositivo di autenticazione a più fattori (MFA) viene perso, danneggiato o non funziona, potrai accedere utilizzando un altro dispositivo MFA registrato dello stesso utente root o utente IAM.

Per accedere utilizzando un altro dispositivo MFA

1. Accedi alla Console di gestione AWS con il tuo ID Account AWS o l'alias e la password dell'account.
2. Nella pagina Verifica aggiuntiva richiesta o nella pagina di Multi-factor autenticazione, scegli Prova un altro metodo MFA.
3. Effettua l'autenticazione con il tipo di dispositivo MFA selezionato.
4. Il passaggio successivo varia a seconda che l'accesso sia stato eseguito correttamente con un dispositivo MFA alternativo.
   - Se hai effettuato correttamente l'accesso, puoi Risincronizzare i dispositivi MFA virtuali e hardware, il che potrebbe risolvere il problema. In caso di smarrimento o di malfunzionamento del dispositivo MFA, puoi disattivarlo. Per istruzioni sulla disattivazione dei dispositivi MFA, consulta Disattivare un dispositivo MFA.
   - Se non riesci ad accedere con MFA, segui i passaggi indicati in Ripristino di un dispositivo MFA per l'utente root o Ripristino di un dispositivo MFA dell'utente IAM per recuperare la tua identità protetta da MFA.

Ripristino di un dispositivo MFA per l'utente root

Se non è possibile effettuare con MFA, puoi utilizzare metodi di autenticazione alternativi per accedere tramite la verifica dell'identità, utilizzando l'e-mail e il numero di telefono di contatto principale registrati nell'account.

Prima di utilizzare fattori di autenticazione alternativi per accedere come utente root, assicurati di avere accesso all'e-mail e al numero di telefono di contatto principale associati all'account. Se devi aggiornare il numero di telefono di contatto principale, puoi accedere come utente IAM con accesso da Amministratore anziché come utente root. Per ulteriori istruzioni sull'aggiornamento delle informazioni di contatto dell'account, consulta Modifica le informazioni di contatto nella Guida per l'utente AWS Billing . Se non hai accesso a un'e-mail e al numero di telefono di contatto principale, contatta Supporto AWS.

Important

Ti consigliamo di mantenere aggiornati l'indirizzo email e il numero di telefono di contatto collegati all'utente root per ripristinare correttamente l'account. Per ulteriori informazioni, consulta Aggiornare il contatto principale per Account AWS nella Guida di riferimento Gestione dell'account AWS .

Per accedere utilizzando fattori di autenticazione alternativi come Utente root dell'account AWS

1. Accedi Console di gestione AWS come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.
2. Nella pagina Verifica aggiuntiva richiesta, seleziona un metodo MFA con cui eseguire l'autenticazione e scegli Avanti.

Note

È possibile che venga visualizzato un testo alternativo, ad esempio Sign in using MFA (Accedi utilizzando la MFA), Troubleshoot your authentication device (Risolvi i problemi del tuo dispositivo di autenticazione) oppure Troubleshoot MFA (Risolvi i problemi di MFA), ma la funzionalità è la stessa. Se non riesci a utilizzare i fattori alternativi di autenticazione per verificare l'indirizzo e-mail e il numero di telefono di contatto principale, contatta Supporto AWS per disattivare il dispositivo MFA.

3. A seconda del tipo di MFA in uso, verrà visualizzata una pagina diversa, ma l'opzione Risoluzione dei problemi relativi all'autenticazione MFA funziona comunque. Nella pagina Verifica aggiuntiva richiesta o nella pagina di Multi-factor autenticazione, scegli Risoluzione dei problemi MFA.
4. Se necessario, digitare di nuovo la password e selezionare Accedi.
5. Nella pagina Risoluzione dei problemi del dispositivo di autenticazione, in Accedi utilizzando fattori alternativi di autenticazione, scegli Accedi utilizzando fattori alternativi.
6. Nella pagina Accedi utilizzando fattori di autenticazione alternativi, autentica il tuo account verificando l'indirizzo e-mail e scegli Invia email di verifica.
7. Controlla l'email associata alla tua Account AWS per ricevere un messaggio da Amazon Web Services (recover-mfa-no-reply@verify.signin.aws). Seguire le istruzioni nel messaggio.

   Se il messaggio non fosse presente nell'account e-mail, controllare la cartella spam o tornare al browser e selezionare Resend the email (Rinvia l'e-mail).
8. Una volta verificato l'indirizzo e-mail, puoi continuare con l'autenticazione dell'account. Per verificare il numero di telefono di contatto principale, scegli Call me now (Chiamami ora).
9. Rispondi alla chiamata AWS e, quando richiesto, inserisci il numero a 6 cifre indicato sul AWS sito Web sulla tastiera del telefono.

   Se non ricevi una chiamata da AWS, scegli Accedi per accedere nuovamente alla console e ricominciare da capo. Oppure consulta Dispositivo di Multi-Factor autenticazione smarrito o inutilizzabile (MFA) per contattare l'assistenza e ricevere assistenza.
10. Una volta effettuata la verifica del numero di telefono, è possibile effettuare l'accesso all'account selezionando Sign in to the console (Accedi alla console).
11. La prossima fase varia a seconda del tipo di MFA in uso:

    - Se si utilizza un dispositivo MFA virtuale, rimuovere l'account dal dispositivo. Quindi passa alla pagina Credenziali di sicurezza AWS ed elimina l'entità del vecchio dispositivo MFA virtuale prima di crearne una nuova.
    - Per una chiave di sicurezza FIDO, consulta la pagina Credenziali AWS di sicurezza e disattiva la vecchia chiave di sicurezza FIDO prima di abilitarne una nuova.
    - Per un token TOTP hardware, contatta il provider di terze parti per assistenza con la riparazione o la sostituzione del dispositivo. L'accesso tramite fattori alternativi di autenticazione può essere utilizzato fino alla ricezione di un nuovo dispositivo. Dopo aver

installato il nuovo dispositivo MFA hardware, consulta la pagina Credenziali AWS di sicurezza ed elimina il vecchio dispositivo MFA.

> **Note**
>
> Non è necessario sostituire il dispositivo MFA smarrito o rubato con lo stesso tipo di dispositivo. Ad esempio, se la chiave di sicurezza FIDO viene danneggiata e ne ordini una nuova, potrai utilizzare il dispositivo MFA virtuale o il token TOTP hardware finché non ricevi la nuova chiave FIDO.

> **Important**
>
> In caso di smarrimento o furto del dispositivo MFA, modifica la password dell'utente root dopo aver effettuato l'accesso e aver installato il dispositivo MFA sostitutivo. Un malintenzionato potrebbe aver rubato il dispositivo di autenticazione e potrebbe anche avere la tua password attuale. Per ulteriori informazioni, consulta Cambia la password per Utente root dell'account AWS.

Ripristino di un dispositivo MFA dell'utente IAM

Se sei un utente IAM che non può accedere con MFA, non puoi recuperare un dispositivo MFA da solo, ma devi contattare un amministratore per disattivare il dispositivo. Quindi puoi abilitare un nuovo dispositivo.

Come ottenere assistenza per un dispositivo MFA in qualità di utente IAM

1. Contatta l' AWS amministratore o l'altra persona che ti ha fornito il nome utente e la password per l'utente IAM. L'amministratore deve disattivare il dispositivo MFA, come descritto in Disattivare un dispositivo MFA, in modo da consentire l'accesso.
2. La prossima fase varia a seconda del tipo di MFA in uso:
   - Se si utilizza un dispositivo MFA virtuale, rimuovere l'account dal dispositivo. Attivare il dispositivo virtuale come descritto in Assegna un dispositivo MFA virtuale nel Console di gestione AWS.

- Per una chiave di sicurezza FIDO, contatta il provider di terze parti per ricevere assistenza con la sostituzione del dispositivo. Quando ricevi la nuova chiave di sicurezza FIDO, devi abilitarla come descritto nella sezione Assegna una passkey o una chiave di sicurezza nel Console di gestione AWS.
- Per un token TOTP hardware, contatta il provider di terze parti per assistenza con la riparazione o la sostituzione del dispositivo. Una volta ottenuto il nuovo dispositivo MFA fisico, abilitarlo nel modo descritto in Assegna un token TOTP hardware nel Console di gestione AWS.

> **Note**
>
> Non è necessario sostituire il dispositivo MFA smarrito o rubato con lo stesso tipo di dispositivo. Puoi avere fino a otto dispositivi MFA in una qualsiasi combinazione. Ad esempio, se la chiave di sicurezza FIDO viene danneggiata e ne ordini una nuova, potrai utilizzare il dispositivo MFA virtuale o il token TOTP hardware finché non ricevi la nuova chiave FIDO.

3. Se il dispositivo MFA è stato smarrito o rubato, modificare anche la password nel caso in cui chi si è impossessato del dispositivo di autenticazione possieda anche la password attualmente in uso. Per ulteriori informazioni, consulta Gestione delle password per gli utenti IAM

Accesso sicuro alle API con MFA

Con le policy IAM, è possibile specificare le operazioni API che un utente è autorizzato a chiamare. È possibile applicare una sicurezza aggiuntiva richiedendo agli utenti di eseguire l'autenticazione a più fattori (MFA) prima di consentire l'esecuzione di operazioni particolarmente sensibili.

Potrebbe ad esempio esserci una policy che permette a un utente di eseguire le operazioni `RunInstances`, `DescribeInstances` e `StopInstances` di Amazon EC2. Ma potresti voler limitare un'azione distruttiva come `TerminateInstances` quella di fare in modo che gli utenti possano eseguirla solo se si autenticano con un dispositivo MFA. AWS

Argomenti

- Panoramica di
- Scenario: Protezione MFA per la delega tra account
- Scenario: Protezione MFA per l'accesso alle operazioni API nell'account corrente

- Scenario: Protezione MFA per le risorse che hanno policy basate su risorse

Panoramica di

L'aggiunta della protezione MFA alle operazioni API prevede le operazioni seguenti:

1. L'amministratore configura un dispositivo AWS MFA per ogni utente che deve effettuare richieste API che richiedono l'autenticazione MFA. Per ulteriori informazioni, consulta AWS Multi-factor autenticazione in IAM.
2. L'amministratore crea politiche per gli utenti che includono un `Condition` elemento che verifica se l'utente si è autenticato con un dispositivo MFA. AWS
3. L'utente chiama una delle operazioni AWS STS API che supportano i parametri MFA: o. AssumeRole GetSessionToken Durante la chiamata, l'utente include l'ID dispositivo per il dispositivo associato all'utente. L'utente include anche la password una tantum a tempo (TOTP) generata dal dispositivo. In entrambi i casi, l'utente ottiene le credenziali di sicurezza temporanee che può quindi usare per effettuare richieste aggiuntive in AWS.

> **Note**
>
> La protezione MFA per le operazioni API di un servizio è disponibile solo se il servizio supporta le credenziali di sicurezza temporanee. Per un elenco di questi servizi, consulta Utilizzo di credenziali di sicurezza temporanee per accedere ad AWS.

Se l'autorizzazione fallisce, AWS restituisce un messaggio di errore di accesso negato (come avviene per qualsiasi accesso non autorizzato). Con le politiche MFA-protected API in atto, AWS nega l'accesso alle operazioni API specificate nelle politiche se l'utente tenta di chiamare un'operazione API senza un'autenticazione MFA valida. L'operazione viene rifiutata anche se il timestamp della richiesta di operazione API è al di fuori dell'intervallo consentito specificato nella policy. L'utente deve essere autenticato di nuovo con MFA richiedendo nuove credenziali di sicurezza temporanee con un codice MFA e il numero di serie del dispositivo.

Policy IAM con condizioni MFA

Le policy con condizioni MFA possono essere collegate a:

- Un utente o un gruppo IAM

- Una risorsa, ad esempio un bucket Amazon S3, una coda Amazon SQS o un argomento Amazon SNS
- La policy di attendibilità di un ruolo IAM che può essere assunto da un utente

Puoi usare una condizione MFA in una policy per controllare le proprietà seguenti:

- Esistenza: per verificare semplicemente che l'utente abbia eseguito l'autenticazione con MFA, controlla che la chiave `aws:MultiFactorAuthPresent` sia `True` in una condizione `Bool`. La chiave è presente solo quando l'utente si autentica con credenziali a breve termine. Long-term le credenziali, come le chiavi di accesso, non includono questa chiave.
- Durata: se desideri concedere l'accesso solo per un periodo di tempo specificato dopo l'autenticazione MFA, usa una condizione di tipo numerico per confrontare la validità della chiave `aws:MultiFactorAuthAge` con un valore (ad esempio 3.600 secondi). Ricordati che la chiave `aws:MultiFactorAuthAge` non è presente se non è stata usata l'autenticazione MFA.

L'esempio seguente mostra la policy di attendibilità di un ruolo IAM che include una condizione MFA da testare per verificare l'esistenza dell'autenticazione MFA. Con questo criterio, gli utenti Account AWS specificati nell'`Principal`elemento (sostituiti `ACCOUNT-B-ID` con un Account AWS ID valido) possono assumere il ruolo a cui è associato questo criterio. ma solo se si sono autenticati tramite MFA.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"AWS": "ACCOUNT-B-ID"},
    "Action": "sts:AssumeRole",
    "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
  }
}
```

Per ulteriori informazioni sui tipi di condizioni per MFA, consulta AWS chiavi di contesto delle condizioni globali, Operatori di condizione numerici e Operatore di condizione per verificare la presenza di chiavi di condizione.

Scelta tra GetSessionToken e AssumeRole

AWS STS fornisce due operazioni API che consentono agli utenti di trasmettere informazioni MFA: `GetSessionToken` e`AssumeRole`. L'operazione API che l'utente chiama per ottenere le credenziali di sicurezza temporanee dipende dallo scenario applicabile tra quelli descritti di seguito.

Usa **`GetSessionToken`** per gli scenari seguenti:

- Chiama le operazioni API che accedono alle risorse nello Account AWS stesso modo in cui l'utente IAM effettua la richiesta. Tieni presente che le credenziali temporanee di una `GetSessionToken` richiesta possono accedere alle operazioni IAM e AWS STS API solo se includi informazioni MFA nella richiesta di credenziali. Poiché le credenziali temporanee restituite da `GetSessionToken` includono le informazioni MFA, puoi verificare l'MFA nelle singole operazioni API effettuate tramite le credenziali.
- Accesso alle risorse protette con policy basate su risorse che includono una condizione MFA.

Lo scopo dell'operazione `GetSessionToken` è autenticare l'utente tramite MFA. Non è possibile utilizzare le policy per controllare le operazioni di autenticazione.

Usa **`AssumeRole`** per gli scenari seguenti:

- Chiamata alle operazioni API che accedono alle risorse nello stesso Account AWS o in un account diverso. Le chiamate API possono includere qualsiasi IAM o API. AWS STS Per proteggere l'accesso, l'autenticazione MFA viene applicata quando l'utente assume il ruolo. Le credenziali temporanee restituite da `AssumeRole` non includono le informazioni MFA nel contesto, quindi non puoi verificare l'MFA nelle singole operazioni API. Per questo motivo, è necessario usare `GetSessionToken` per limitare l'accesso alle risorse protette da policy basate su risorse.

> **Note**
>
> AWS CloudTrail i log conterranno informazioni MFA quando l'utente IAM accede con MFA. Se l'utente IAM assume un ruolo IAM, CloudTrail `mfaAuthenticated: true` accederà anche `sessionContext` agli attributi per le azioni eseguite utilizzando il ruolo assunto. Tuttavia, la CloudTrail registrazione è separata da ciò che richiede IAM quando le chiamate API vengono effettuate con le credenziali del ruolo assunto. Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Le informazioni su come implementare questi scenari vengono fornite più avanti in questo documento.

Punti importanti sull'accesso alle API MFA-protected

È importante comprendere i seguenti aspetti della protezione MFA per le operazioni API:

- La protezione MFA è disponibile solo con le credenziali di sicurezza temporanee, che è possibile ottenere con `AssumeRole` o `GetSessionToken`.
- Non è possibile utilizzare l'accesso all' MFA-protected API con Utente root dell'account AWS credenziali.
- Non è possibile utilizzare l'accesso alle MFA-protected API con le chiavi di sicurezza U2F.
- Agli utenti federati non può essere assegnato un dispositivo MFA da utilizzare con i AWS servizi, quindi non possono accedere alle AWS risorse controllate dall'MFA. Vedi il punto successivo.
- Altre operazioni AWS STS API che restituiscono credenziali temporanee non supportano l'MFA. Per `AssumeRoleWithWebIdentity` e`AssumeRoleWithSAML`, l'utente è autenticato da un provider esterno e AWS non può determinare se quel provider richieda l'MFA. Per `GetFederationToken`, l'autenticazione MFA non è necessariamente associata a un utente specifico.
- Allo stesso modo, le credenziali a lungo termine (chiavi di accesso utente IAM e chiavi di accesso utente root) non possono essere utilizzate con l'accesso alle MFA-protected API perché non scadono.
- È possibile chiamare `AssumeRole` e `GetSessionToken` anche senza informazioni MFA. In tal caso, il chiamante riceve le credenziali di sicurezza temporanee, ma le informazioni di sessione per tali credenziali temporanee non indicano che l'utente ha eseguito l'autenticazione con MFA.
- Per stabilire la protezione MFA per le operazioni API, aggiungi condizioni MFA alle policy. Una policy deve includere la chiave di condizione `aws:MultiFactorAuthPresent` per implementare l'uso dell'MFA. Per la delega tra più account, la policy di attendibilità del ruolo deve includere la chiave di condizione.
- Quando consenti Account AWS a un altro di accedere alle risorse del tuo account, la sicurezza delle tue risorse dipende dalla configurazione dell'account attendibile (l'altro account, non il tuo). Questo vale anche quando è richiesta la multi-factor authentication. Qualsiasi identità nell'account attendibile che dispone dell'autorizzazione per creare dispositivi MFA virtuali può creare un'attestazione MFA per soddisfare tale parte della policy di affidabilità del ruolo. Prima di consentire ai membri di un altro account di accedere alle tue AWS risorse che richiedono l'autenticazione a più fattori, devi assicurarti che il proprietario dell'account attendibile segua

le migliori pratiche di sicurezza. Ad esempio, l'account attendibile deve limitare l'accesso alle operazioni API sensibili, ad esempio le operazioni API di gestione dei dispositivi MFA, a identità attendibili specifiche.

- Se una policy include una condizione MFA, una richiesta viene negata se gli utenti non sono stati autenticati tramite MFA oppure se forniscono una password TOTP o un identificatore di dispositivo MFA non valido.

Scenario: Protezione MFA per la delega tra account

In questo scenario, vuoi delegare l'accesso agli utenti IAM in un altro account, ma solo se gli utenti sono autenticati con un AWS dispositivo MFA. Per ulteriori informazioni sulla delega tra più account, consulta Termini e concetti dei ruoli.

Immagina di avere un account A (l'account che determina l'attendibilità, proprietario della risorsa a cui è necessario accedere), con l'utente IAM Anaya, che ha l'autorizzazione di amministratore. Anaya desidera concedere l'accesso all'utente Richard nell'account B (l'account attendibile), ma vuole assicurarsi che Richard sia autenticato con MFA prima di poter assumere il ruolo.

1. Nell'account di fiducia A, Anaya crea un ruolo IAM denominato `CrossAccountRole` e imposta il principale nella politica di fiducia del ruolo sull'ID dell'account B. La politica di fiducia concede l'autorizzazione all'azione. AWS STS `AssumeRole` Anaya aggiunge inoltre una condizione MFA alla policy di trust, come nell'esempio seguente.

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": {
       "Effect": "Allow",
       "Principal": {"AWS": "ACCOUNT-B-ID"},
       "Action": "sts:AssumeRole",
       "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
     }
   }
   ```

2. Anaya aggiunge una policy di autorizzazione al ruolo che specifica le attività consentite per il ruolo. La policy di autorizzazione per un ruolo con protezione MFA è uguale a qualsiasi altra policy di autorizzazione di un ruolo. L'esempio seguente mostra la policy aggiunta al ruolo da Anaya, che consente a un utente ipotetico di eseguire qualsiasi operazione Amazon DynamoDB sulla tabella

`Books` nell'account A. Questa policy consente anche l'operazione `dynamodb:ListTables`, necessaria per eseguire operazioni nella console.

> **Note**
>
> La policy di autorizzazione non include una condizione MFA. È importante comprendere che l'autenticazione MFA viene usata solo per determinare se un utente può assumere tale ruolo. Una volta che l'utente ha assunto il ruolo, non vengono svolti ulteriori controlli MFA.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "TableActions",
            "Effect": "Allow",
            "Action": "dynamodb:*",
            "Resource": "arn:aws:dynamodb:*:111122223333:table/Books"
        },
        {
            "Sid": "ListTables",
            "Effect": "Allow",
            "Action": "dynamodb:ListTables",
            "Resource": "*"
        }
    ]
}
```

3. Nell'account attendibile B, l'amministratore si assicura che l'utente IAM Richard sia configurato con un dispositivo AWS MFA e che conosca l'ID del dispositivo. ovvero il numero di serie se si tratta di un dispositivo MFA hardware o l'ARN del dispositivo se si tratta di un dispositivo MFA virtuale.
4. Nell'account B, l'amministratore collega all'utente Richard (o un gruppo di cui è membro) la policy seguente, che gli permette di chiamare l'operazione `AssumeRole`. La risorsa è impostata sull'ARN del ruolo creato da Anaya nella fase 1. Osserva che questa policy non contiene una condizione MFA.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/CrossAccountRole"
            ]
        }
    ]
}
```

5. Nell'account B, Richard (o un'applicazione che Richard sta eseguendo) chiama `AssumeRole`. La chiamata dell'API include l'ARN del ruolo da assumere (`arn:aws:iam::ACCOUNT-A-ID:role/CrossAccountRole`), l'ID del dispositivo MFA e la password TOTP corrente che Richard ottiene dal suo dispositivo.

   Quando Richard chiama`AssumeRole`, AWS determina se dispone di credenziali valide, incluso il requisito per l'MFA. In caso affermativo, Richard assume correttamente il ruolo e può eseguire qualsiasi operazione DynamoDB sulla tabella denominata `Books` nell'account A usando le credenziali temporanee del ruolo.

   Per un esempio di programma che chiama `AssumeRole`, consulta AssumeRole Chiamate con autenticazione MFA.

Scenario: Protezione MFA per l'accesso alle operazioni API nell'account corrente

In questo scenario, è necessario assicurarsi che un utente del proprio account Account AWS possa accedere alle operazioni API sensibili solo quando l'utente è autenticato tramite un AWS dispositivo MFA.

Immagina di avere un account A che contiene un gruppo di sviluppatori che devono usare le istanze EC2. In genere, gli sviluppatori possono usare le istanze, ma non hanno le autorizzazioni per le operazioni `ec2:StopInstances` e `ec2:TerminateInstances`. È opportuno limitare queste

operazioni privilegiate "distruttive" a pochi utenti attendibili, quindi aggiungi la protezione MFA alla policy che permette queste operazioni Amazon EC2 sensibili.

In questo scenario, uno degli utenti attendibili è Sofía. L'utente Anaya è un amministratore nell'account A.

1. Anaya si assicura che Sofía sia configurata con un dispositivo AWS MFA e che Sofía conosca l'ID del dispositivo. ovvero il numero di serie se si tratta di un dispositivo MFA hardware o l'ARN del dispositivo se si tratta di un dispositivo MFA virtuale.
2. Anaya crea un gruppo denominato `EC2-Admins` e aggiunge l'utente Sofía al gruppo.
3. Anaya collega la policy seguente al gruppo `EC2-Admins`. Questa policy concede agli utenti l'autorizzazione per chiamare le operazioni `StopInstances` e `TerminateInstances` di Amazon EC2 solo se l'utente ha eseguito l'autenticazione tramite MFA.

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [{
       "Effect": "Allow",
       "Action": [
         "ec2:StopInstances",
         "ec2:TerminateInstances"
       ],
       "Resource": ["*"],
       "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
     }]
   }
   ```

4.
   > **Note**
   >
   > Per rendere effettiva questa policy, gli utenti devono prima disconnettersi e quindi accedere nuovamente.

   Se l'utente Sofía deve arrestare o terminare un'istanza Amazon EC2, l'utente o un'applicazione da lei eseguita, chiama `GetSessionToken`. Questa operazione API passa l'ID o del dispositivo MFA e la password TOTP corrente che Sofía ottiene dal suo dispositivo.

5. L'utente Sofía (o un'applicazione che Sofía sta usando) usa le credenziali temporanee fornite da `GetSessionToken` per chiamare l'operazione `StopInstances` o `TerminateInstances` di Amazon EC2.

   Per un esempio di programma che chiama `GetSessionToken`, consulta GetSessionToken Chiamate con autenticazione MFA più avanti in questo documento.

Scenario: Protezione MFA per le risorse che hanno policy basate su risorse

In questo scenario, sei il proprietario di un bucket S3, una coda SQS o un argomento SNS. Vuoi assicurarti che qualsiasi utente di Account AWS chiunque acceda alla risorsa sia autenticato da un dispositivo MFA. AWS

Questo scenario illustra un modo per fornire la protezione MFA per più account senza richiedere agli utenti di assumere prima un ruolo. In tal caso, l'utente può accedere alla risorsa se vengono soddisfatte tre condizioni: l'utente deve essere autenticato mediante MFA, essere in grado di ottenere credenziali di sicurezza temporanee da `GetSessionToken` ed essere in un account ritenuto attendibile dalla policy della risorsa.

Immagina di avere l'account A e di creare un bucket S3. Vuoi concedere l'accesso a questo bucket a utenti che si trovano in diversi ambienti Account AWS, ma solo se tali utenti sono autenticati con MFA.

In questo scenario, l'utente Anaya è un amministratore nell'account A. L'utente Nikhil è un utente IAM nell'account C.

1. Nell'account A, Anaya crea un bucket denominato `Account-A-bucket`.
2. Anaya aggiunge la policy del bucket al bucket. La policy permette a qualsiasi utente in un account A, un account B o un account C di eseguire le operazioni `PutObject` e `DeleteObject` di Amazon S3 nel bucket. La policy include una condizione MFA.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"AWS": [
      "ACCOUNT-A-ID",
      "ACCOUNT-B-ID",
```

```
      "ACCOUNT-C-ID"
    ]},
    "Action": [
      "s3:PutObject",
      "s3:DeleteObject"
    ],
    "Resource": ["arn:aws:s3:::ACCOUNT-A-BUCKET-NAME/*"],
    "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
  }]
}
```

> **Note**
>
> Amazon S3 offre la funzionalità Cancellazione MFA solo per l'accesso all'account root. Puoi abilitare la funzionalità Cancellazione MFA di Amazon S3 quando imposti lo stato del controllo delle versioni del bucket. Amazon S3 MFA Delete non può essere applicato a un utente IAM ed è gestito indipendentemente dall'accesso alle API. MFA-protected Un utente IAM con l'autorizzazione per eliminare un bucket non può eliminare un bucket quando la funzionalità Cancellazione MFA di Amazon S3 è abilitata. Per ulteriori informazioni sulla funzionalità Cancellazione MFA di Amazon S3, consulta Cancellazione MFA.

3. Nell'account C, un amministratore verifica che l'utente Nikhil sia configurato con un dispositivo MFA AWS e che conosca l'ID del dispositivo. ovvero il numero di serie se si tratta di un dispositivo MFA hardware o l'ARN del dispositivo se si tratta di un dispositivo MFA virtuale.
4. Nell'account C, Nikhil (o un'applicazione che lui sta eseguendo) chiama `GetSessionToken`. La chiamata include l'ID o l'ARN del dispositivo MFA e la password TOTP corrente che Nikhil ottiene dal suo dispositivo.
5. Nikhil (o un'applicazione che lui sta usando) usa le credenziali temporanee restituite da `GetSessionToken` per chiamare l'operazione `PutObject` di Amazon S3 per caricare un file in `Account-A-bucket`.

   Per un esempio di programma che chiama `GetSessionToken`, consulta GetSessionToken Chiamate con autenticazione MFA più avanti in questo documento.

> **Note**
>
> Le credenziali temporanee che `AssumeRole` restituisce non funzionano in questo caso. Anche se l'utente è in grado di fornire informazioni MFA per assumere un ruolo, le

credenziali temporanee restituite da `AssumeRole` non includono le informazioni MFA. Queste informazione sono necessarie per soddisfare la condizione MFA nella policy.

Codice di esempio: richiesta di credenziali con l'autenticazione a più fattori (MFA)

I seguenti esempi mostrano come chiamare le operazioni `GetSessionToken` e `AssumeRole` e passare i parametri di autenticazione MFA. Non è richiesta alcuna autorizzazione per chiamare `GetSessionToken`, ma è necessario disporre di una policy che permetta di chiamare `AssumeRole`. Le credenziali restituite vengono quindi utilizzate per elencare tutti i bucket S3 nell'account.

GetSessionToken Chiamate con autenticazione MFA

I seguenti esempi mostrano come chiamare `GetSessionToken` e passare le informazioni sull'autenticazione MFA. Le credenziali di sicurezza temporanee restituite dall'operazione `GetSessionToken` vengono quindi utilizzate per elencare tutti i bucket S3 nell'account.

La policy collegata all'utente che esegue questo codice (o a un gruppo in cui si trova utente) fornisce le autorizzazioni per le credenziali temporanee restituite. Per questo codice di esempio, la policy deve concedere all'utente l'autorizzazione per richiedere l'operazione `ListBuckets` di Amazon S3.

Gli esempi di codice seguenti mostrano come utilizzare `GetSessionToken`.

CLI

AWS CLI

Come ottenere un set di credenziali a breve termine per un'identità IAM

il comando `get-session-token` seguente recupera un set di credenziali a breve termine per l'identità IAM che esegue la chiamata. Le credenziali risultanti possono essere utilizzate per richieste in cui l'autenticazione a più fattori (MFA) è richiesta dalla policy. Le credenziali scadono 15 minuti dopo la loro generazione.

```
aws sts get-session-token \
    --duration-seconds 900 \
    --serial-number "YourMFADeviceSerialNumber" \
    --token-code 123456
```

Output:

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    }
}
```

Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza temporanee nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta GetSessionToken AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: restituisce un'istanza **Amazon.RuntimeAWSCredentials** contenente credenziali temporanee valide per un determinato periodo di tempo. Le credenziali utilizzate per richiedere credenziali temporanee vengono dedotte dalle impostazioni predefinite correnti della shell. Per specificare altre credenziali, utilizzare i parametri - ProfileName o - AccessKey SecretKey /.

```
Get-STSSessionToken
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 2: restituisce un'istanza **Amazon.RuntimeAWSCredentials** contenente credenziali temporanee valide per un'ora. Le credenziali utilizzate per effettuare la richiesta vengono ottenute dal profilo specificato.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 3: restituisce un'istanza **`Amazon.RuntimeAWSCredentials`** contenente credenziali temporanee valide per un'ora utilizzando il numero di identificazione del dispositivo MFA associato all'account le cui credenziali sono specificate nel profilo 'myprofilename' e il valore fornito dal dispositivo.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile -SerialNumber
 YourMFADeviceSerialNumber -TokenCode 123456
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

- Per i dettagli sull'API, vedere GetSessionTokenin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: restituisce un'istanza **`Amazon.RuntimeAWSCredentials`** contenente credenziali temporanee valide per un determinato periodo di tempo. Le credenziali utilizzate per richiedere credenziali temporanee vengono dedotte dalle impostazioni predefinite correnti della shell. Per specificare altre credenziali, utilizzare i parametri - ProfileName o - AccessKey SecretKey /.

```
Get-STSSessionToken
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 2: restituisce un'istanza **`Amazon.RuntimeAWSCredentials`** contenente credenziali temporanee valide per un'ora. Le credenziali utilizzate per effettuare la richiesta vengono ottenute dal profilo specificato.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 3: restituisce un'istanza **`Amazon.RuntimeAWSCredentials`** contenente credenziali temporanee valide per un'ora utilizzando il numero di identificazione del dispositivo MFA associato all'account le cui credenziali sono specificate nel profilo 'myprofilename' e il valore fornito dal dispositivo.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile -SerialNumber
 YourMFADeviceSerialNumber -TokenCode 123456
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

- Per i dettagli sull'API, vedere GetSessionTokenin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera un token di sessione passando un token MFA e utilizzalo per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_with_session_token_with_mfa(mfa_serial_number, mfa_totp,
 sts_client):
    """
    Gets a session token with MFA credentials and uses the temporary session
    credentials to list Amazon S3 buckets.

    Requires an MFA device serial number and token.

    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an Amazon Resource Name (ARN).
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
 role.
    """
    if mfa_serial_number is not None:
        response = sts_client.get_session_token(
            SerialNumber=mfa_serial_number, TokenCode=mfa_totp
        )
    else:
        response = sts_client.get_session_token()
    temp_credentials = response["Credentials"]

    s3_resource = boto3.resource(
        "s3",
```

```
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )

    print(f"Buckets for the account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

- Per i dettagli sull'API, consulta GetSessionToken AWSSDK for Python (Boto3) API Reference.

AssumeRole Chiamate con autenticazione MFA

I seguenti esempi mostrano come chiamare `AssumeRole` e passare le informazioni sull'autenticazione MFA. Le credenziali di sicurezza temporanee restituite da `AssumeRole` vengono quindi utilizzate per elencare tutti i bucket Amazon S3 nell'account.

Per ulteriori informazioni su questo scenario, consulta Scenario: Protezione MFA per la delega tra account.

Gli esempi di codice seguenti mostrano come utilizzare `AssumeRole`.

.NET

SDK per .NET

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon;
using Amazon.SecurityToken;
using Amazon.SecurityToken.Model;
```

```
namespace AssumeRoleExample
{
    class AssumeRole
    {
        /// <summary>
        /// This example shows how to use the AWS Security Token
        /// Service (AWS STS) to assume an IAM role.
        ///
        /// NOTE: It is important that the role that will be assumed has a
        /// trust relationship with the account that will assume the role.
        ///
        /// Before you run the example, you need to create the role you want to
        /// assume and have it trust the IAM account that will assume that role.
        ///
        /// See https://docs.aws.amazon.com/IAM/latest/UserGuide/
id_roles_create.html
        /// for help in working with roles.
        /// </summary>

        // A region property may be used if the profile or credentials loaded do
 not specify a region,
        // or to use a specific region.
        private static readonly RegionEndpoint REGION = RegionEndpoint.USWest2;

        static async Task Main()
        {
            // Create the SecurityToken client and then display the identity of
 the
            // default user.
            var roleArnToAssume = "arn:aws:iam::123456789012:role/
testAssumeRole";

            var client = new
 Amazon.SecurityToken.AmazonSecurityTokenServiceClient(REGION);

            // Get and display the information about the identity of the default
 user.
            var callerIdRequest = new GetCallerIdentityRequest();
            var caller = await client.GetCallerIdentityAsync(callerIdRequest);
            Console.WriteLine($"Original Caller: {caller.Arn}");

            // Create the request to use with the AssumeRoleAsync call.
            var assumeRoleReq = new AssumeRoleRequest()
```

```
            {
                DurationSeconds = 1600,
                RoleSessionName = "Session1",
                RoleArn = roleArnToAssume
            };

            var assumeRoleRes = await client.AssumeRoleAsync(assumeRoleReq);

            // Now create a new client based on the credentials of the caller
 assuming the role.
            var client2 = new AmazonSecurityTokenServiceClient(credentials:
 assumeRoleRes.Credentials, REGION);

            // Get and display information about the caller that has assumed the
 defined role.
            var caller2 = await client2.GetCallerIdentityAsync(callerIdRequest);
            Console.WriteLine($"AssumedRole Caller: {caller2.Arn}");
        }
    }
}
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
#############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
```

```
    echo "$@"
  fi
}

############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

############################################################################
# function sts_assume_role
#
# This function assumes a role in the AWS account and returns the temporary
#  credentials.
#
# Parameters:
#       -n role_session_name -- The name of the session.
#       -r role_arn -- The ARN of the role to assume.
#
# Returns:
#       [access_key_id, secret_access_key, session_token]
#     And:
#       0 - If successful.
#       1 - If an error occurred.
############################################################################
function sts_assume_role() {
  local role_session_name role_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function sts_assume_role"
    echo "Assumes a role in the AWS account and returns the temporary
 credentials:"
    echo "  -n role_session_name -- The name of the session."
    echo "  -r role_arn -- The ARN of the role to assume."
    echo ""
  }

  while getopts n:r:h option; do
```

```
    case "${option}" in
      n) role_session_name=${OPTARG} ;;
      r) role_arn=${OPTARG} ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  response=$(aws sts assume-role \
    --role-session-name "$role_session_name" \
    --role-arn "$role_arn" \
    --output text \
    --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta AssumeRole AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::STS::assumeRole(const Aws::String &roleArn,
                             const Aws::String &roleSessionName,
                             const Aws::String &externalId,
                             Aws::Auth::AWSCredentials &credentials,
                             const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::STS::STSClient sts(clientConfig);
    Aws::STS::Model::AssumeRoleRequest sts_req;

    sts_req.SetRoleArn(roleArn);
    sts_req.SetRoleSessionName(roleSessionName);
    sts_req.SetExternalId(externalId);

    const Aws::STS::Model::AssumeRoleOutcome outcome = sts.AssumeRole(sts_req);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error assuming IAM role. " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Credentials successfully retrieved." << std::endl;
        const Aws::STS::Model::AssumeRoleResult result = outcome.GetResult();
        const Aws::STS::Model::Credentials &temp_credentials =
 result.GetCredentials();

        // Store temporary credentials in return argument.
        // Note: The credentials object returned by assumeRole differs
        // from the AWSCredentials object used in most situations.
        credentials.SetAWSAccessKeyId(temp_credentials.GetAccessKeyId());
        credentials.SetAWSSecretKey(temp_credentials.GetSecretAccessKey());
        credentials.SetSessionToken(temp_credentials.GetSessionToken());
    }
```

```
    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come assumere un ruolo

Il comando `assume-role` seguente recupera un set di credenziali a breve termine per il ruolo IAM `s3-access-example`.

```
aws sts assume-role \
    --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
    --role-session-name s3-access-example
```

Output:

```
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
        "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-
access-example"
    },
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/
qwjzP2iEXAMPLEbw/m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8B
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2016-03-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    }
}
```

L'output del comando contiene una chiave di accesso, una chiave segreta e un token di sessione che puoi utilizzare per l'autenticazione in AWS.

Per l'utilizzo della AWS CLI, è possibile impostare un profilo denominato associato a un ruolo. Quando utilizzi il profilo, la AWS CLI chiamerà assume-role e gestirà le credenziali per te. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM nella CLI nella AWS CLI User Guide AWS .

- Per i dettagli sull'API, consulta AWS CLI Command AssumeRoleReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.sts.StsClient;
import software.amazon.awssdk.services.sts.model.AssumeRoleRequest;
import software.amazon.awssdk.services.sts.model.StsException;
import software.amazon.awssdk.services.sts.model.AssumeRoleResponse;
import software.amazon.awssdk.services.sts.model.Credentials;
import java.time.Instant;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;
import java.util.Locale;

/**
 * To make this code example work, create a Role that you want to assume.
 * Then define a Trust Relationship in the AWS Console. You can use this as an
 * example:
 *
 * {
 * "Version":"2012-10-17",
 * "Statement": [
 * {
 * "Effect": "Allow",
```

```
 * "Principal": {
 * "AWS": "<Specify the ARN of your IAM user you are using in this code example>"
 * },
 * "Action": "sts:AssumeRole"
 * }
 * ]
 * }
 *
 * For more information, see "Editing the Trust Relationship for an Existing
 * Role" in the AWS Directory Service guide.
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class AssumeRole {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <roleArn> <roleSessionName>\s

                Where:
                    roleArn - The Amazon Resource Name (ARN) of the role to
 assume (for example, arn:aws:iam::000008047983:role/s3role).\s
                    roleSessionName - An identifier for the assumed role session
 (for example, mysession).\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String roleArn = args[0];
        String roleSessionName = args[1];
        Region region = Region.US_EAST_1;
        StsClient stsClient = StsClient.builder()
                .region(region)
                .build();
```

```
        assumeGivenRole(stsClient, roleArn, roleSessionName);
        stsClient.close();
    }

    public static void assumeGivenRole(StsClient stsClient, String roleArn,
 String roleSessionName) {
        try {
            AssumeRoleRequest roleRequest = AssumeRoleRequest.builder()
                    .roleArn(roleArn)
                    .roleSessionName(roleSessionName)
                    .build();

            AssumeRoleResponse roleResponse = stsClient.assumeRole(roleRequest);
            Credentials myCreds = roleResponse.credentials();

            // Display the time when the temp creds expire.
            Instant exTime = myCreds.expiration();
            String tokenInfo = myCreds.sessionToken();

            // Convert the Instant to readable date.
            DateTimeFormatter formatter =
 DateTimeFormatter.ofLocalizedDateTime(FormatStyle.SHORT)
                    .withLocale(Locale.US)
                    .withZone(ZoneId.systemDefault());

            formatter.format(exTime);
            System.out.println("The token " + tokenInfo + "  expires on " +
 exTime);

        } catch (StsException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea il client.

```
import { STSClient } from "@aws-sdk/client-sts";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an AWS STS service client object.
export const client = new STSClient({ region: REGION });
```

Assumi il ruolo IAM.

```
import { AssumeRoleCommand } from "@aws-sdk/client-sts";

import { client } from "../libs/client.js";

export const main = async () => {
  try {
    // Returns a set of temporary security credentials that you can use to
    // access Amazon Web Services resources that you might not normally
    // have access to.
    const command = new AssumeRoleCommand({
      // The Amazon Resource Name (ARN) of the role to assume.
      RoleArn: "ROLE_ARN",
      // An identifier for the assumed role session.
      RoleSessionName: "session1",
      // The duration, in seconds, of the role session. The value specified
      // can range from 900 seconds (15 minutes) up to the maximum session
      // duration set for the role.
      DurationSeconds: 900,
    });
    const response = await client.send(command);
    console.log(response);
```

```
  } catch (err) {
    console.error(err);
  }
};
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
const AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

var roleToAssume = {
  RoleArn: "arn:aws:iam::123456789012:role/RoleName",
  RoleSessionName: "session1",
  DurationSeconds: 900,
};
var roleCreds;

// Create the STS service object
var sts = new AWS.STS({ apiVersion: "2011-06-15" });

//Assume Role
sts.assumeRole(roleToAssume, function (err, data) {
  if (err) console.log(err, err.stack);
  else {
    roleCreds = {
      accessKeyId: data.Credentials.AccessKeyId,
      secretAccessKey: data.Credentials.SecretAccessKey,
      sessionToken: data.Credentials.SessionToken,
    };
    stsGetCallerIdentity(roleCreds);
  }
```

```
});

//Get Arn of current identity
function stsGetCallerIdentity(creds) {
  var stsParams = { credentials: creds };
  // Create STS service object
  var sts = new AWS.STS(stsParams);

  sts.getCallerIdentity({}, function (err, data) {
    if (err) {
      console.log(err, err.stack);
    } else {
      console.log(data.Arn);
    }
  });
}
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: restituisce un set di credenziali temporanee (chiave di accesso, chiave segreta e token di sessione) che possono essere utilizzate per un'ora per accedere a AWS risorse a cui l'utente richiedente potrebbe normalmente non avere accesso. Le credenziali restituite hanno le autorizzazioni consentite dalla policy di accesso del ruolo assunto e dalla policy fornita (non è possibile utilizzare la policy fornita per concedere autorizzazioni superiori a quelle definite dalla policy di accesso del ruolo assunto).

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -Policy "...JSON policy..." -DurationInSeconds 3600
```

Esempio 2: restituisce un set di credenziali temporanee, valide per un'ora, con le stesse autorizzazioni definite nella policy di accesso del ruolo assunto.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600
```

Esempio 3: restituisce un set di credenziali temporanee che forniscono il numero di serie e il token generato da un MFA associato alle credenziali utente utilizzate per eseguire il cmdlet.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -SerialNumber "GAHT12345678" -TokenCode "123456"
```

Esempio 4: restituisce un set di credenziali temporanee che hanno assunto un ruolo definito in un account cliente. Per ogni ruolo che la terza parte può assumere, l'account cliente deve creare un ruolo utilizzando un identificatore che deve essere passato nel ExternalId parametro - ogni volta che viene assunto il ruolo.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -ExternalId "ABC123"
```

- Per i dettagli sull'API, vedere AssumeRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: restituisce un set di credenziali temporanee (chiave di accesso, chiave segreta e token di sessione) che possono essere utilizzate per un'ora per accedere a AWS risorse a cui l'utente richiedente potrebbe normalmente non avere accesso. Le credenziali restituite hanno le autorizzazioni consentite dalla policy di accesso del ruolo assunto e dalla policy fornita (non è possibile utilizzare la policy fornita per concedere autorizzazioni superiori a quelle definite dalla policy di accesso del ruolo assunto).

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -Policy "...JSON policy..." -DurationInSeconds 3600
```

Esempio 2: restituisce un set di credenziali temporanee, valide per un'ora, con le stesse autorizzazioni definite nella policy di accesso del ruolo assunto.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600
```

Esempio 3: restituisce un set di credenziali temporanee che forniscono il numero di serie e il token generato da un MFA associato alle credenziali utente utilizzate per eseguire il cmdlet.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -SerialNumber "GAHT12345678" -TokenCode "123456"
```

Esempio 4: restituisce un set di credenziali temporanee che hanno assunto un ruolo definito in un account cliente. Per ogni ruolo che la terza parte può assumere, l'account cliente deve creare un ruolo utilizzando un identificatore che deve essere passato nel ExternalId parametro - ogni volta che viene assunto il ruolo.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -ExternalId "ABC123"
```

- Per i dettagli sull'API, vedere AssumeRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Assumi un ruolo IAM che richiede un token MFA e utilizza le credenziali temporanee per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_from_assumed_role_with_mfa(
    assume_role_arn, session_name, mfa_serial_number, mfa_totp, sts_client
):
    """
    Assumes a role from another account and uses the temporary credentials from
    that role to list the Amazon S3 buckets that are owned by the other account.
    Requires an MFA device serial number and token.

    The assumed role must grant permission to list the buckets in the other
 account.

    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an ARN.
```

```
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
role.
    """
    response = sts_client.assume_role(
        RoleArn=assume_role_arn,
        RoleSessionName=session_name,
        SerialNumber=mfa_serial_number,
        TokenCode=mfa_totp,
    )
    temp_credentials = response["Credentials"]
    print(f"Assumed role {assume_role_arn} and got temporary credentials.")

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )

    print(f"Listing buckets for the assumed role's account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

- Per i dettagli sull'API, consulta AssumeRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
 # Creates an AWS Security Token Service (AWS STS) client with specified
credentials.
```

```
 # This is separated into a factory function so that it can be mocked for unit
testing.
 #
 # @param key_id [String] The ID of the access key used by the STS client.
 # @param key_secret [String] The secret part of the access key used by the STS
client.
 def create_sts_client(key_id, key_secret)
   Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
 end

 # Gets temporary credentials that can be used to assume a role.
 #
 # @param role_arn [String] The ARN of the role that is assumed when these
credentials
 #                          are used.
 # @param sts_client [AWS::STS::Client] An AWS STS client.
 # @return [Aws::AssumeRoleCredentials] The credentials that can be used to
assume the role.
 def assume_role(role_arn, sts_client)
   credentials = Aws::AssumeRoleCredentials.new(
     client: sts_client,
     role_arn: role_arn,
     role_session_name: 'create-use-assume-role-scenario'
   )
   @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
   credentials
 end
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
async fn assume_role(config: &SdkConfig, role_name: String, session_name:
 Option<String>) {
    let provider = aws_config::sts::AssumeRoleProvider::builder(role_name)
        .session_name(session_name.unwrap_or("rust_sdk_example_session".into()))
        .configure(config)
        .build()
        .await;

    let local_config = aws_config::from_env()
        .credentials_provider(provider)
        .load()
        .await;
    let client = Client::new(&local_config);
    let req = client.get_caller_identity();
    let resp = req.send().await;
    match resp {
        Ok(e) => {
            println!("UserID :               {}",
 e.user_id().unwrap_or_default());
            println!("Account:               {}",
 e.account().unwrap_or_default());
            println!("Arn    :               {}", e.arn().unwrap_or_default());
        }
        Err(e) => println!("{:?}", e),
    }
}
```

- Per i dettagli sulle API, consulta la AssumeRoleguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSSTS

    public func assumeRole(role: IAMClientTypes.Role, sessionName: String)
        async throws -> STSClientTypes.Credentials
    {
        let input = AssumeRoleInput(
            roleArn: role.arn,
            roleSessionName: sessionName
        )
        do {
            let output = try await stsClient.assumeRole(input: input)

            guard let credentials = output.credentials else {
                throw ServiceHandlerError.authError
            }

            return credentials
        } catch {
            print("Error assuming role: ", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la AssumeRoleguida di riferimento all'API AWS SDK for Swift.

Credenziali specifiche del servizio per gli utenti IAM

Le credenziali specifiche del servizio sono meccanismi di autenticazione specializzati progettati per servizi specifici. AWS Queste credenziali forniscono un'autenticazione semplificata rispetto alle AWS credenziali standard e sono personalizzate in base ai requisiti di autenticazione dei singoli servizi. AWS A differenza delle chiavi di accesso, che possono essere utilizzate su più AWS servizi, le credenziali specifiche del servizio sono progettate per essere utilizzate solo con il servizio per il quale sono state create. Questo approccio mirato migliora la sicurezza limitando l'ambito delle credenziali.

Generalmente, le credenziali specifiche del servizio sono costituite da una coppia di nome utente e password o da chiavi API specializzate formattate in base ai requisiti del servizio specifico. Quando si creano credenziali specifiche per un servizio, queste sono attive per impostazione predefinita e possono essere utilizzate immediatamente. Puoi avere un massimo di due set di credenziali

specifiche del servizio per ogni servizio supportato per utente IAM. Questo limite consente di mantenere un set attivo mentre si passa a un nuovo set quando necessario. AWS attualmente supporta credenziali specifiche del servizio per i seguenti servizi:

Quando utilizzare le credenziali specifiche del servizio

Le credenziali specifiche del servizio sono destinate alla compatibilità con librerie SDKs, strumenti o applicazioni di terze parti che non sono nativamente compatibili con le credenziali, oppure. AWS AWS SDKs AWS APIs Tali casi d'uso includono la migrazione a AWS servizi da infrastrutture ospitate autonomamente o da servizi ospitati da altri provider.

Quando si parte da zero, e laddove possibile, si consiglia di utilizzare credenziali AWS temporanee, come quelle fornite da un ruolo IAM, per l'autenticazione con un AWS servizio utilizzando un AWS SDK o una libreria che supporta credenziali temporanee. AWS

Rotazione delle credenziali specifiche del servizio

Come best practice di sicurezza, si consiglia di ruotare regolarmente le credenziali specifiche del servizio. Per ruotare le credenziali senza interrompere le applicazioni:

1. Creare un secondo set di credenziali specifiche del servizio per lo stesso servizio e lo stesso utente IAM
2. Aggiornare tutte le applicazioni per fare in modo che utilizzino le nuove credenziali e verificare che funzionino correttamente
3. Cambiare lo stato delle credenziali originali in “Inattiva”
4. Verificare che tutte le applicazioni funzionino ancora correttamente
5. Eliminare le credenziali specifiche del servizio inattive quando si ha la certezza che non siano più necessarie

Monitoraggio delle credenziali specifiche del servizio

Puoi utilizzarlo AWS CloudTrail per monitorare l'uso di credenziali specifiche del servizio nel tuo account. AWS Per visualizzare CloudTrail gli eventi relativi all'utilizzo delle credenziali specifiche del servizio, esamina CloudTrail i registri per verificare la presenza di eventi del servizio in cui vengono utilizzate le credenziali. Per ulteriori informazioni, consulta Registrazione di IAM e AWS STS Chiamate API con AWS CloudTrail.

Per una maggiore sicurezza, prendi in considerazione la possibilità di impostare CloudWatch allarmi per informarti di specifici modelli di utilizzo delle credenziali che potrebbero indicare accessi non autorizzati o altri problemi di sicurezza. Per ulteriori informazioni, consulta Monitoring CloudTrail Log Files with Amazon CloudWatch Logs nella Guida per l'AWS CloudTrail utente.

I seguenti argomenti forniscono informazioni sulle credenziali specifiche del servizio.

Argomenti

- Chiavi API per AWS services
- Utilizzare IAM con Amazon Keyspaces (per Apache Cassandra)

Chiavi API per AWS services

Alcuni AWS servizi supportano le chiavi API per l'autenticazione delle richieste programmatiche oltre alle credenziali IAM standard, come le credenziali di sicurezza temporanee e le chiavi di accesso a lungo termine. AWS offre due tipi di chiavi API:

- Long-term Chiavi API: le chiavi Long-term API sono associate a un utente IAM e generate utilizzando credenziali specifiche del servizio IAM. Queste credenziali sono progettate per essere utilizzate con un solo AWS servizio, migliorando la sicurezza limitando l'ambito delle credenziali. Per le chiavi API a lungo termine puoi impostare una scadenza. Per generare chiavi API a lungo termine, puoi utilizzare IAM o la console specifica del servizio, la, o l'API. AWS CLI AWS
- Short-term Chiavi API: una chiave API a breve termine è un URL prefirmato che utilizza la AWS Signature Version 4. Short-term Le chiavi API hanno le stesse autorizzazioni e la stessa scadenza delle credenziali dell'identità che genera la chiave API e sono valide fino a 12 ore o per il tempo rimanente della sessione della console, a seconda di quale sia il più breve. Puoi utilizzare la Claude Platform su AWS console Amazon Bedrock/, Python e i pacchetti per altri linguaggi di programmazione per generare chiavi API a breve termine. Per ulteriori informazioni, consulta Generazione di chiavi API Amazon Bedrock per un facile accesso all'API Amazon Bedrock nella Guida per l'utente di Amazon Bedrock e Autenticazione nella Guida per l'utente. Claude Platform su AWS

> **Note**
>
> Long-term Le chiavi API presentano un rischio di sicurezza più elevato rispetto alle chiavi API a breve termine. Quando possibile, consigliamo di utilizzare chiavi API a breve termine o

credenziali di sicurezza temporanee. Se utilizzi chiavi API a lungo termine, ti consigliamo di implementare pratiche di rotazione delle chiavi a intervalli regolari.

Servizi che supportano le chiavi API

La tabella seguente elenca i AWS servizi che supportano le chiavi API e il tipo di chiave API supportato da ciascun servizio.

#	Servizio	Long-term Chiavi API	Short-term chiavi API	Policy gestita allegata automatic amente	Service-specific documenta zione
1	Amazon Bedrock	Sì	Sì	AmazonBed rockLimit edAccess	Usa una chiave API Amazon Bedrock
2	Claude Platform su AWS	Sì	Sì	Anthropic Inference Access	Autentica zione
3	Amazon CloudWatch	Sì	N/A	CloudWatc hAPIKeyAc cess	Configura zione dell'aute nticazione con token al portatore per Metrics
4	Amazon Logs CloudWatch	Sì	N/A	CloudWatc hLogsAPIK eyAccess	Configura zione dell'aute nticazione con token al portatore

Quando si genera una chiave API a lungo termine per un servizio, la policy AWS gestita corrispondente viene automaticamente allegata all'utente IAM, garantendo l'accesso alle operazioni principali di quel servizio. Se hai bisogno di un accesso aggiuntivo, puoi modificare le autorizzazioni per l'utente IAM. Per informazioni sulla modifica delle autorizzazioni, consulta the section called "Aggiunta o rimozione di autorizzazioni per le identità".

Per saperne di più sulle chiavi API per servizi specifici, consulta i link alla Service-specific documentazione nella tabella precedente.

Prerequisiti per le chiavi API a lungo termine

Prima di poter generare una chiave API a lungo termine nella console IAM, devi soddisfare questi prerequisiti:

- Un utente IAM da associare alla chiave API a lungo termine. Per istruzioni su come creare un utente IAM, consulta the section called "Creazione di un utente".
- È necessario disporre delle seguenti autorizzazioni della policy IAM per gestire le credenziali specifiche del servizio per un utente IAM. La policy di esempio concede l'autorizzazione a creare, elencare, aggiornare, eliminare e reimpostare le credenziali specifiche del servizio. Sostituisci il *username* valore nell'elemento Resource con il nome dell'utente IAM per cui genererai chiavi API a lungo termine per:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ManageBedrockServiceSpecificCredentials",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceSpecificCredential",
                "iam:ListServiceSpecificCredentials",
                "iam:UpdateServiceSpecificCredential",
                "iam:DeleteServiceSpecificCredential",
                "iam:ResetServiceSpecificCredential"
            ],
            "Resource": "arn:aws:iam::*:user/username"
        }
    ]
}
```

Generazione di una chiave API a lungo termine (console)

Generare una chiave API a lungo termine per un servizio specifico nella console IAM

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegli Utenti.
3. Scegli l'utente IAM per cui desideri generare una chiave API a lungo termine.
4. Selezionare la scheda Security Credentials (Credenziali di sicurezza).
5. Nella sezione Chiavi API, scegli Genera chiave API.
6. Dall'elenco a discesa del AWS servizio, scegli il servizio su cui desideri autenticare la chiave API.
7. Per Scadenza della chiave API, esegui una delle seguenti operazioni:
   - Scegli una durata di scadenza della chiave API di 1, 5, 30 , 90 o 365 giorni.
   - Scegli Durata personalizzata per specificare una data di scadenza della chiave API personalizzata.
   - Scegli Non scade mai (scelta non consigliata).
8. Scegli Genera chiave API.
9. Copia o scarica la tua chiave API. Questo è l'unico momento in cui è possibile visualizzare il valore della chiave API.

> **Important**
>
> Archivia la chiave API in modo sicuro. Dopo aver chiuso la finestra di dialogo, non sarà possibile recuperare di nuovo la chiave API. Se perdi o dimentichi la tua chiave API, non puoi recuperarla. Invece, genera una nuova chiave API e rendi inattiva la vecchia chiave.

Generazione di una chiave API a lungo termine (AWS CLI)

Per generare una chiave API a lungo termine utilizzando il AWS CLI, utilizza i seguenti passaggi:

1. Crea un utente IAM che verrà utilizzato con il servizio utilizzando il comando create-user:

```
aws iam create-user \
```

```
    --user-name APIKeyUser_1
```

2. Allega la policy AWS gestita all'utente IAM utilizzando il comando https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/attach-user-policy.html attach-user-policy.

   Per Amazon Bedrock:

   ```
   aws iam attach-user-policy --user-name APIKeyUser_1 \
       --policy-arn arn:aws:iam::aws:policy/AmazonBedrockLimitedAccess
   ```

   Per Claude Platform su AWS:

   ```
   aws iam attach-user-policy --user-name APIKeyUser_1 \
       --policy-arn arn:aws:iam::aws:policy/AnthropicInferenceAccess
   ```

   Per Amazon CloudWatch:

   ```
   aws iam attach-user-policy --user-name APIKeyUser_1 \
       --policy-arn arn:aws:iam::aws:policy/CloudWatchAPIKeyAccess
   ```

   Per Amazon CloudWatch Logs:

   ```
   aws iam attach-user-policy --user-name APIKeyUser_1 \
       --policy-arn arn:aws:iam::aws:policy/CloudWatchLogsAPIKeyAccess
   ```

3. Genera la chiave API a lungo termine utilizzando il comando create-service-specific-credential.

   Per Amazon Bedrock:

   ```
   aws iam create-service-specific-credential \
       --user-name APIKeyUser_1 \
       --service-name bedrock.amazonaws.com \
       --credential-age-days 30
   ```

   Per Claude Platform su AWS:

   ```
   aws iam create-service-specific-credential \
       --user-name APIKeyUser_1 \
       --service-name aws-external-anthropic.amazonaws.com \
       --credential-age-days 30
   ```

Per Amazon CloudWatch:

```
aws iam create-service-specific-credential \
    --user-name APIKeyUser_1 \
    --service-name cloudwatch.amazonaws.com \
    --credential-age-days 30
```

Per Amazon CloudWatch Logs:

```
aws iam create-service-specific-credential \
    --user-name APIKeyUser_1 \
    --service-name logs.amazonaws.com \
    --credential-age-days 30
```

> **Note**
>
> Il parametro `--credential-age-days` è facoltativo. Puoi specificare un valore compreso tra 1 e 36600 giorni. Se ometti questo parametro, la chiave API non scade.

La risposta `ServiceApiKeyValue` restituita è la chiave API a lungo termine per il rispettivo servizio. Archivia il valore `ServiceApiKeyValue` in modo sicuro, poiché non potrai recuperarlo in un secondo momento.

Elenca le chiavi API a lungo termine (AWS CLI)

Per elencare i metadati delle chiavi API a lungo termine per un utente specifico, usa il comando list-service-specific-credentials con il parametro: `--user-name`

```
aws iam list-service-specific-credentials \
    --service-name bedrock.amazonaws.com \
    --user-name APIKeyUser_1
```

> **Note**
>
> Sostituisci `bedrock.amazonaws.com` con il nome del servizio appropriato (ad esempio, per Amazon Logs o per). `logs.amazonaws.com` CloudWatch `aws-external-anthropic.amazonaws.com` Claude Platform su AWS

Per elencare tutti i metadati delle chiavi API a lungo termine presenti nell'account, utilizza il comando list-service-specific-credentials con il parametro: `--all-users`

```
aws iam list-service-specific-credentials \
    --service-name bedrock.amazonaws.com \
    --all-users
```

Aggiorna lo stato della chiave API a lungo termine (AWS CLI)

Per aggiornare lo stato di una chiave API a lungo termine, usa il comando https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/update-service-specific-credential.html update-service-specific-credential:

```
aws iam update-service-specific-credential \
    --user-name "APIKeyUser_1" \
    --service-specific-credential-id "ACCA1234EXAMPLE1234" \
    --status Inactive|Active
```

Generazione di una chiave API a lungo termine (AWS API)

Puoi utilizzare le seguenti operazioni API IAM per gestire le chiavi API a lungo termine per qualsiasi servizio supportato:

- CreateServiceSpecificCredential
- ListServiceSpecificCredentials
- UpdateServiceSpecificCredential
- DeleteServiceSpecificCredential
- ResetServiceSpecificCredential

Short-term Chiavi API (servizi selezionati)

Short-term Le chiavi API sono attualmente supportate da servizi selezionati.

Per informazioni sulla generazione e l'utilizzo di chiavi API a breve termine con Amazon Bedrock, consulta Generare una chiave API nella Amazon Bedrock User Guide.

Per informazioni sulla generazione e l'utilizzo di chiavi API a breve termine per Claude Platform su AWS, consulta l'Autenticazione nella Guida per l'Claude Platform su AWS utente.

Service-specific informazioni

- Per ulteriori informazioni sull'utilizzo delle chiavi API con Amazon Bedrock, consulta Use an Amazon Bedrock API key nella Amazon Bedrock User Guide.
- Per ulteriori informazioni sull'utilizzo delle chiavi API con Claude Platform su AWS, consulta Autenticazione nella Guida per l'Claude Platform su AWS utente.
- Per ulteriori informazioni sull'utilizzo delle chiavi API con Amazon CloudWatch, consulta Configurare l'autenticazione con token al portatore per le metriche nella Amazon CloudWatch User Guide.
- Per ulteriori informazioni sull'utilizzo delle chiavi API con Amazon CloudWatch Logs, consulta Configurazione dell'autenticazione con token al portatore nella Amazon CloudWatch Logs User Guide.

Utilizzare IAM con Amazon Keyspaces (per Apache Cassandra)

Amazon Keyspaces (per Apache Cassandra) è un servizio di database gestito, scalabile, ad alta disponibilità e compatibile con Apache Cassandra. Puoi accedere ad Amazon Keyspaces tramite o Console di gestione AWS programmaticamente. Per accedere ad Amazon Keyspaces a livello di programmazione con credenziali specifiche del servizio, puoi utilizzare `cqlsh` o i driver Cassandra open source. Le credenziali specifiche del servizio includono un nome utente e una password come quelli che Cassandra utilizza per l'autenticazione e la gestione degli accessi. Puoi avere un massimo di due set di credenziali specifiche del servizio per ogni servizio supportato per utente.

Per accedere ad Amazon Keyspaces in modo programmatico con le chiavi di AWS accesso, puoi utilizzare l' AWS SDK, il AWS Command Line Interface (AWS CLI) o i driver Cassandra open source con il plug-in SigV4. Per ulteriori informazioni, consulta Creare e configurare AWS credenziali per Amazon Keyspaces nella Amazon Keyspaces (for Apache Cassandra) Developer Guide.

> **Note**
>
> Se prevedi di interagire con Amazon Keyspaces solo tramite la console, non è necessario generare credenziali specifiche del servizio. Per ulteriori informazioni, consulta la sezione Accesso ad Amazon Keyspaces (per Apache Cassandra) nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).

Per ulteriori informazioni sulle autorizzazioni richieste per accedere ad Amazon Keyspaces, consulta Esempi di policy basate su identità per Amazon Keyspaces (per Apache Cassandra) nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).

Generazione delle credenziali Amazon Keyspaces (console)

Puoi utilizzare Console di gestione AWS per generare credenziali Amazon Keyspaces (per Apache Cassandra) per i tuoi utenti IAM.

Come generare credenziali specifiche del servizio Amazon Keyspaces (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione, scegliere Users (Utenti) quindi selezionare il nome dell'utente che richiede le credenziali.
3. Nella scheda Credenziali di sicurezza in Credenziali per Amazon Keyspaces (per Apache Cassandra), scegli Genera credenziali.
4. Le credenziali specifiche del servizio sono ora disponibili. Questa è l'unica volta in cui è possibile visualizzare o scaricare la password. Non puoi recuperarla successivamente. Tuttavia, è possibile reimpostare la password in qualsiasi momento. Salva l'utente e la password in una posizione sicura, perché ne avrai bisogno in un secondo momento.

Generazione delle credenziali di Amazon Keyspaces (AWS CLI)

Puoi utilizzare AWS CLI per generare credenziali Amazon Keyspaces (per Apache Cassandra) per i tuoi utenti IAM.

Come generare credenziali specifiche del servizio Amazon Keyspaces (AWS CLI)

- Utilizza il seguente comando:
  - era iam create-service-specific-credential

Generazione di credenziali Amazon Keyspaces (API)AWS

Puoi utilizzare l' AWS API per generare credenziali Amazon Keyspaces (per Apache Cassandra) per i tuoi utenti IAM.

Per generare credenziali specifiche del servizio Amazon Keyspaces (API)AWS

- Completare la seguente operazione:
  - CreateServiceSpecificCredential

Trova oggetti inutilizzati AWS credenziali

Per aumentare la sicurezza del tuo account Account AWS, rimuovi le credenziali utente IAM (ovvero password e chiavi di accesso) che non sono necessarie. Ad esempio, quando gli utenti lasciano l'organizzazione o non hanno più bisogno di AWS accedervi, trovate le credenziali che stavano utilizzando e assicuratevi che non siano più operative. La soluzione ideale consiste nell'eliminare tutte le credenziali inutilizzate. Se l'utente dovesse averne bisogno in un secondo momento, potrai sempre ricrearle. Come minimo, dovresti modificare la password o disattivare le chiavi di accesso, per impedire l'accesso agli ex utenti.

Ovviamente, la definizione di inutilizzata è abbastanza ambigua, ma in genere si intende che una credenziale non è stata utilizzata per un determinato periodo di tempo.

Ricerca di password inutilizzate

Puoi utilizzarli Console di gestione AWS per visualizzare le informazioni sull'utilizzo delle password per i tuoi utenti. Se il numero di utenti è elevato, puoi utilizzare la console per scaricare un report delle credenziali, con informazioni sui tempi di utilizzo delle password della console. Puoi anche accedere alle informazioni dall'API AWS CLI o dall'API IAM.

Per individuare le password inutilizzate (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Se necessario, aggiungere la colonna Console last sign-in (Ultimo accesso alla console) nella tabella degli utenti:
   a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ( ).
   b. In Select visible columns (Seleziona colonne visibili), seleziona Console last sign-in (Ultimo accesso alla console).

c. Seleziona Confirm (Conferma) per tornare all'elenco degli utenti.

4. La colonna Ultimo accesso alla console mostra la data dell'ultimo accesso dell'utente AWS tramite la console. Mediante queste informazioni puoi trovare gli utenti le cui password non sono state utilizzate per un determinato periodo di tempo. La colonna mostra Never (Mai) per gli utenti che non hanno mai usato le password per accedere. None (Nessuna) indica gli utenti senza password. Le password non utilizzate di recente potrebbero essere candidate ideali per la rimozione.

> **Important**
>
> A causa di un problema di servizio, i dati sull'ultimo utilizzo della password non includono il periodo compreso fra le 22.50 (PDT) del 3 maggio 2018 e le 14.08 (PDT) del 23 maggio 2018. Ciò influisce sulle date dell'ultimo accesso mostrate nella console IAM e sulle date dell'ultimo utilizzo della password nel rapporto sulle credenziali IAM e restituite dall'GetUser operazione API. Se un utente ha effettuato l'accesso durante il periodo interessato, l'ultima data di utilizzo di password visualizzata sarà quella relativa all'ultimo utilizzo prima del 3 maggio 2018. Per gli utenti che hanno effettuato l'accesso dopo le 14.08 PDT del 23 maggio 2018, la data indicata sarà accurata.
> Se utilizzi le informazioni relative all'ultima password utilizzata per identificare le credenziali non utilizzate da eliminare, ad esempio per eliminare gli utenti che non hanno effettuato l'accesso AWS negli ultimi 90 giorni, ti consigliamo di modificare la finestra di valutazione per includere date successive al 23 maggio 2018. In alternativa, se gli utenti utilizzano le chiavi di accesso per accedere a AWS livello di codice, è possibile fare riferimento alle ultime informazioni utilizzate sulla chiave di accesso, poiché sono accurate per tutte le date.

Per trovare le password inutilizzate scaricando il report delle credenziali (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione, selezionare Credential report (Rapporto credenziali).
3. Selezionare Download Report (Scarica report) per scaricare un file CSV denominato `status_reports_<date>T<time>.csv`. La quinta colonna contiene la colonna `password_last_used` con le date o uno dei seguenti messaggi:

   - N/A— Utenti a cui non è stata assegnata alcuna password.

- no_information: gli utenti che non hanno utilizzato la propria password da quando IAM ha iniziato a monitorarne l'utilizzo (20 ottobre 2014).

Per trovare le password non utilizzate (AWS CLI)

Per individuare le password non utilizzate, seguire il comando seguente:

- `aws iam list-users` restituisce un elenco di utenti, ciascuno con un valore `PasswordLastUsed`. Se il valore è mancante, l'utente non ha una password oppure la password non è stata utilizzata da quando IAM ha iniziato a monitorarne l'utilizzo (20 ottobre 2014).

Per trovare le password non utilizzate (AWS API)

Per individuare le password non utilizzate, richiamare la seguente operazione:

- `ListUsers` restituisce una raccolta di utenti, ciascuno delle quali ha un valore `<PasswordLastUsed>`. Se il valore è mancante, l'utente non ha una password oppure la password non è stata utilizzata da quando IAM ha iniziato a monitorarne l'utilizzo (20 ottobre 2014).

Per informazioni sui comandi per scaricare il report delle credenziali, consultare Ottenere rapporti sulle credenziali (AWS CLI).

Ricerca di chiavi di accesso inutilizzate

Puoi utilizzare il Console di gestione AWS per visualizzare le informazioni sull'utilizzo delle chiavi di accesso per i tuoi utenti. Se il numero di utenti è elevato, puoi utilizzare la console per scaricare un report delle credenziali per sapere quando gli utenti hanno utilizzato le loro chiavi di accesso. Puoi anche accedere alle informazioni dall'API AWS CLI o dall'API IAM.

Per trovare le chiavi di accesso inutilizzate (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Se necessario, aggiungere la colonna Access key last used (Ultimo utilizzo chiave di accesso) nella tabella degli utenti:

a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ().

b. In Select visible columns (Seleziona colonne visibili), seleziona Access key last used (Ultima chiave d'accesso utilizzata).

c. Seleziona Confirm (Conferma) per tornare all'elenco degli utenti.

4. La colonna Ultima chiave di accesso utilizzata mostra il numero di giorni trascorsi dall'ultimo accesso dell'utente a livello di AWS programmazione. Mediante queste informazioni puoi trovare le chiavi di accesso che non sono state utilizzate per un determinato periodo di tempo. Per gli utenti senza chiavi di accesso nella colonna è riportato –. Le chiavi di accesso non utilizzate di recente potrebbero essere candidate ideali per la rimozione.

Per trovare le chiavi di accesso inutilizzate scaricando il report delle credenziali (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione, selezionare Credential Report (Rapporto credenziali).
3. Selezionare Download Report (Scarica report) per scaricare un file CSV denominato `status_reports_<date>T<time>.csv`. Le colonne da 11 a 13 contengono la data di ultimo utilizzo, la regione e le informazioni di servizio per la chiave di accesso 1. Le colonne da 16 a 18 contengono le stesse informazioni per la chiave di accesso 2. Il valore è N/Ase l'utente non dispone di una chiave di accesso o non l'ha utilizzata da quando IAM ha iniziato a tracciare l'età della chiave di accesso il 22 aprile 2015.

Per trovare le chiavi di accesso non utilizzate (AWS CLI)

Per individuare le chiavi di accesso non utilizzate, eseguire i comandi seguenti:

- `aws iam list-access-keys` restituisce informazioni sulle chiavi di accesso di un utente, tra cui `AccessKeyID`.
- `aws iam get-access-key-last-used` prende un ID chiave di accesso e restituisce informazioni, tra cui `LastUsedDate`, `Region` in cui la chiave di accesso è stata utilizzata e il `ServiceName` dell'ultimo servizio richiesto. Se `LastUsedDate` è mancante, la chiave di accesso non è stata utilizzata da quando IAM ha iniziato a monitorarne l'utilizzo (22 aprile 2015).

Per trovare le chiavi di accesso non utilizzate (AWS API)

Per individuare le chiavi di accesso non utilizzate, richiamare le seguenti operazioni:

- `ListAccessKeys` restituisce un elenco di `AccessKeyID` valori per le chiavi di accesso associati all'utente specificato.
- `GetAccessKeyLastUsed` prende un ID chiave di accesso e restituisce una raccolta di valori. Sono incluse `LastUsedDate`, `Region` in cui la chiave di accesso è stato utilizzato e `ServiceName` dell'ultimo servizio richiesto. Se il valore è mancante, l'utente non dispone di una chiave di accesso oppure non l'ha utilizzata da quando IAM ha iniziato a monitorarne l'utilizzo (22 aprile 2015).

Per informazioni sui comandi per scaricare il report delle credenziali, consultare Ottenere rapporti sulle credenziali (AWS CLI).

Genera report sulle credenziali per i tuoi Account AWS

Puoi generare e scaricare un report delle credenziali che elenca tutti gli utenti presenti nel tuo account e lo stato delle loro diverse credenziali, tra cui le password, le chiavi di accesso e i dispositivi MFA. Puoi ottenere un rapporto sulle credenziali dagli AWS SDK e dagli strumenti a riga di comando o dall'API IAM. Console di gestione AWS

> **Note**
>
> Il rapporto sulle credenziali IAM include solo le seguenti IAM-managed credenziali: password, le prime due chiavi di accesso per utente, dispositivi MFA e certificati di firma. X.509 Il rapporto non include le credenziali specifiche del servizio (come CodeCommit password, chiavi API a lungo termine di Amazon Bedrock o chiavi API a lungo termine di Amazon CloudWatch Logs) o altre chiavi di accesso utente oltre alle prime due. Per una visibilità completa delle credenziali, utilizza le API and. ListServiceSpecificCredentials ListAccessKeys

Puoi utilizzare i report delle credenziali nei tuoi controlli e strategie di conformità. È possibile utilizzare il report per controllare gli effetti dei requisiti del ciclo di vita delle credenziali, come ad esempio la password e gli aggiornamenti della chiave di accesso. Puoi fornire il report a un revisore esterno o concedere le autorizzazioni a un'entità di controllo in modo che possa scaricare il report direttamente.

Puoi generare un report delle credenziali con una frequenza di una volta ogni quattro ore. Quando richiedi un report, IAM verifica innanzitutto se Account AWS è stato generato un report nelle ultime quattro ore. In questo caso, viene scaricato il report più recente. Se il report più recente per l'account è più vecchio di quattro ore oppure se non ci sono report precedenti per l'account, IAM genera e scarica un nuovo report.

Argomenti

- Autorizzazioni richieste
- Comprendere il formato del report
- Recupero dei report delle credenziali (Console)
- Ottenere rapporti sulle credenziali (AWS CLI)
- Ottenere rapporti sulle credenziali (AWS API)

Autorizzazioni richieste

Per creare e scaricare i report sono necessarie le seguenti autorizzazioni:

- Per creare un report delle credenziali: `iam:GenerateCredentialReport`
- Per scaricare il report: `iam:GetCredentialReport`

Comprendere il formato del report

I file dei report delle credenziali sono formattati con una virgola come separatore (CSV). Puoi aprire i file CSV con i comuni software per fogli di calcolo per eseguire l'analisi, oppure è possibile creare un'applicazione che riceva il file CSV in modo programmatico ed esegua un'analisi personalizzata.

Il file CSV contiene le seguenti colonne:

user

Il nome semplice dell'utente.

arn

L'Amazon Resource Name (ARN) dell'utente. Per ulteriori informazioni sugli ARN, consultare la pagina ARN IAM.

user_creation_time

La data e l'ora in cui l'utente è stato creato, nel formato data/ora ISO 8601.

password_enabled

Quando l'utente ha una password, questo valore è `TRUE`. In caso contrario è `FALSE`. Questo valore è `FALSE` per i nuovi account membri creati come parte dell'organizzazione, poiché per impostazione predefinita non dispongono delle credenziali dell'utente root.

password_last_used

La data e l'ora in cui la password dell'utente Utente root dell'account AWS o dell'utente è stata utilizzata l'ultima volta per accedere a un AWS sito Web, nel formato data-ora ISO 8601. AWS i siti Web che registrano l'ora dell'ultimo accesso di un utente sono i Console di gestione AWS Forum di AWS discussione e il Marketplace. AWS Quando una password è utilizzata più volte in un intervallo di 5 minuti, viene registrato in questo campo solo il primo utilizzo.

- Il valore in questo campo è `no_information` in questi casi:
  - La password dell'utente non è mai stata utilizzata.
  - Non sono previsti dati di accesso associati alla password, come ad esempio quando la password dell'utente non è stata utilizzata dopo che IAM ha iniziato a monitorare queste informazioni il 20 ottobre 2014.
- Il valore di questo campo è `N/A` (non applicabile) quando l'utente non ha una password.

> **Important**
>
> A causa di un problema di servizio, i dati sull'ultimo utilizzo della password non includono il periodo compreso fra le 22.50 (PDT) del 3 maggio 2018 e le 14.08 (PDT) del 23 maggio 2018. Ciò influisce sulle date dell'ultimo accesso visualizzate nella console IAM e sulle date dell'ultimo utilizzo della password nel rapporto sulle credenziali IAM e restituite dall'GetUser operazione API. Se un utente ha effettuato l'accesso durante il periodo interessato, l'ultima data di utilizzo di password visualizzata sarà quella relativa all'ultimo utilizzo prima del 3 maggio 2018. Per gli utenti che hanno effettuato l'accesso dopo le 14.08 PDT del 23 maggio 2018, la data indicata sarà accurata.
> Se utilizzi le informazioni sull'ultima password utilizzata per identificare le credenziali non utilizzate da eliminare, ad esempio per eliminare gli utenti che non hanno effettuato l'accesso AWS negli ultimi 90 giorni, ti consigliamo di modificare la finestra di valutazione per includere le date successive al 23 maggio 2018. In alternativa, se i tuoi utenti utilizzano le chiavi di accesso per accedere in modo AWS programmatico, puoi fare riferimento alle informazioni utilizzate per l'ultima chiave di accesso perché sono accurate per tutte le date.

password_last_changed

La data e ora in cui la password dell'utente è stata impostata per l'ultima volta nel formato data/ora ISO 8601. Se l'utente non ha una password, il valore di questo campo è `N/A` (non applicabile).

password_next_rotation

Quando l'account ha una policy della password che richiede la rotazione della password, questo campo contiene la data e l'ora, nel formato data/ora ISO 8601, quando all'utente è richiesto di impostare una nuova password. Il valore per Account AWS (root) è sempre. `not_supported`

mfa_active

Quando un dispositivo Multi-Factor Authentication (MFA) è stato abilitato per l'utente, il valore è `TRUE`. In caso contrario è `FALSE`.

access_key_1_active

Quando l'utente ha una chiave di accesso e lo stato della chiave di accesso è `Active`, questo valore è `TRUE`. In caso contrario è `FALSE`. Si applica sia all'utente root dell'account che agli utenti IAM.

access_key_1_last_rotated

La data e l'ora nel formato data/ora ISO 8601, della creazione della chiave di accesso dell'utente o dell'ultima modifica. Se l'utente non ha una chiave di accesso attiva, il valore in questo campo è `N/A` (non applicabile). Si applica sia all'utente root dell'account che agli utenti IAM.

access_key_1_last_used_date

La data e l'ora, nel formato data/ora ISO 8601, in cui la chiave di accesso dell'utente è stata recentemente utilizzata per firmare una richiesta API AWS . Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo. Si applica sia all'utente root dell'account che agli utenti IAM.

Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso.
- La chiave di accesso non è mai stata utilizzata.
- La chiave di accesso non è stata utilizzata dopo che IAM ha iniziato a monitorarne le informazioni (22 aprile 2015).

access_key_1_last_used_region

La regione AWS in cui la chiave di accesso è stata utilizzata più recentemente. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo. Si applica sia all'utente root dell'account che agli utenti IAM.

Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso.
- La chiave di accesso non è mai stata utilizzata.
- La chiave di accesso è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).
- L'ultimo servizio utilizzato non lo è Region-specific, ad esempio Amazon S3.

access_key_1_last_used_service

Il AWS servizio a cui è stato effettuato l'ultimo accesso con la chiave di accesso. Il valore in questo campo utilizza lo spazio dei nomi del servizio, ad esempio `s3` per Amazon S3 e `ec2` per Amazon EC2. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo. Si applica sia all'utente root dell'account che agli utenti IAM.

Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso.
- La chiave di accesso non è mai stata utilizzata.
- La chiave di accesso è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).

access_key_2_active

Quando l'utente ha una chiave di accesso secondaria e lo stato della chiave di accesso secondaria è `Active`, questo valore è `TRUE`. In caso contrario è `FALSE`. Si applica sia all'utente root dell'account che agli utenti IAM.

> **Note**
>
> Gli utenti possono avere fino a due chiavi di accesso: ciò semplifica la rotazione, consentendo di aggiornare prima la chiave e successivamente di eliminare la chiave precedente. Per ulteriori informazioni sull'aggiornamento delle chiavi di accesso, consulta la pagina Aggiornare le chiavi di accesso.

access_key_2_last_rotated

La data e l'ora, espresse in formato data/ora ISO 8601, di creazione o ultima modifica della seconda chiave di accesso dell'utente. Se l'utente non ha una chiave di accesso secondaria attiva, il valore in questo campo è `N/A` (non applicabile). Si applica sia all'utente root dell'account che agli utenti IAM.

access_key_2_last_used_date

La data e l'ora, nel formato data-ora ISO 8601, in cui la seconda chiave di accesso dell'utente è stata utilizzata più di recente per firmare una AWS richiesta API. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo. Si applica sia all'utente root dell'account che agli utenti IAM.

Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso secondaria.
- La chiave di accesso secondaria dell'utente non è mai stata utilizzata.
- La chiave di accesso secondaria dell'utente è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).

access_key_2_last_used_region

La regione AWS in cui la chiave di accesso secondaria è stata utilizzata più recentemente. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo. Si applica sia all'utente root dell'account che agli utenti IAM. Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso secondaria.
- La chiave di accesso secondaria dell'utente non è mai stata utilizzata.
- La chiave di accesso secondaria dell'utente è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).
- L'ultimo servizio utilizzato non lo è Region-specific, ad esempio Amazon S3.

access_key_2_last_used_service

Il AWS servizio a cui è stato effettuato l'ultimo accesso con la seconda chiave di accesso dell'utente. Il valore in questo campo utilizza lo spazio dei nomi del servizio, ad esempio `s3` per Amazon S3 e `ec2` per Amazon EC2. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo. Si applica sia

all'utente root dell'account che agli utenti IAM. Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso secondaria.
- La chiave di accesso secondaria dell'utente non è mai stata utilizzata.
- La chiave di accesso secondaria dell'utente è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).

cert_1_active

Quando l'utente dispone di un certificato di X.509 firma e lo stato di tale certificato è`Active`, questo valore è`TRUE`. In caso contrario è `FALSE`.

cert_1_last_rotated

La data e l'ora nel formato data/ora ISO 8601 della creazione del certificato di firma dell'utente o dell'ultima modifica. Se l'utente non ha un certificato di firma attivo, il valore in questo campo è `N/A` (non applicabile).

cert_2_active

Quando l'utente dispone di un secondo certificato di X.509 firma e lo stato di tale certificato è`Active`, questo valore è`TRUE`. In caso contrario è `FALSE`.

> **Note**
>
> Gli utenti possono avere fino a due certificati di X.509 firma, per facilitare la rotazione dei certificati.

cert_2_last_rotated

La data e l'ora nel formato data/ora ISO 8601 della creazione del certificato di firma secondario dell'utente o dell'ultima modifica. Se l'utente non ha un certificato di firma secondario attivo, il valore in questo campo è `N/A` (non applicabile).

additional_credentials_info

Quando l'utente dispone di più di due chiavi di accesso o certificati, questo valore è il numero di chiavi di accesso o certificati aggiuntivi e le azioni che è possibile utilizzare per elencare le chiavi di accesso o i certificati associati all'utente.

Recupero dei report delle credenziali (Console)

È possibile utilizzare il Console di gestione AWS per scaricare un rapporto sulle credenziali come file con valori separati da virgole (CSV).

Per scaricare un report delle credenziali (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione, selezionare Credential report (Rapporto credenziali).
3. Scegliere Download Report (Scarica report).

Ottenere rapporti sulle credenziali (AWS CLI)

Per scaricare un rapporto sulle credenziali (AWS CLI)

1. Genera un rapporto sulle credenziali. AWS memorizza un singolo report. Se esiste già un report, la generazione di un report sulle credenziali sovrascrive il report precedente. `aws iam generate-credential-report`
2. Visualizza l'ultimo report generato: `aws iam get-credential-report`

Ottenere rapporti sulle credenziali (AWS API)

Per scaricare un rapporto sulle credenziali (AWS API)

1. Genera un rapporto sulle credenziali. AWS memorizza un singolo report. Se esiste già un report, la generazione di un report sulle credenziali sovrascrive il report precedente. `GenerateCredentialReport`
2. Visualizza l'ultimo report generato: `GetCredentialReport`

Credenziali IAM per CodeCommit: credenziali Git, chiavi SSH e chiavi di accesso AWS

CodeCommit è un servizio di controllo delle versioni gestito che ospita repository Git privati nel AWS cloud. Per utilizzarlo CodeCommit, configuri il tuo client Git per comunicare con i CodeCommit

repository. Come parte di questa configurazione, fornisci credenziali IAM che CodeCommit possono essere utilizzate per autenticarti. IAM supporta tre tipi CodeCommit di credenziali:

- Credenziali Git, una coppia di nome utente e password generata da IAM che puoi usare per comunicare con i CodeCommit repository tramite HTTPS.
- Chiavi SSH, una coppia di chiavi pubblica-privata generata localmente che puoi associare al tuo utente IAM per comunicare con i CodeCommit repository tramite SSH.
- AWS chiavi di accesso, che puoi utilizzare con l'helper per le credenziali incluso nella confezione per comunicare con i AWS CLI repository tramite HTTPS. CodeCommit

> **Note**
>
> Non è possibile utilizzare le chiavi SSH o le credenziali Git per accedere ai repository in un altro account AWS . Per informazioni su come configurare l'accesso ai CodeCommit repository per utenti e gruppi IAM in un altro archivio Account AWS, consulta Configurare l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli nella Guida per l'utente.AWS CodeCommit

Per ulteriori informazioni su ciascuna opzione, consultare le sezioni seguenti.

Usa credenziali Git e HTTPS con CodeCommit (consigliato)

Con le credenziali Git, generi una coppia statica di nome utente e password per il tuo utente IAM e utilizzi tali credenziali per le connessioni HTTPS. Puoi utilizzare queste credenziali anche con qualsiasi strumento di terza parte o ambiente di sviluppo integrato (IDE) che supporta le credenziali Git statiche.

Poiché queste credenziali sono universali per tutti i sistemi operativi supportati e compatibili con la maggior parte dei sistemi di gestione delle credenziali, ambienti di sviluppo e altri strumenti di sviluppo software, questo è il metodo consigliato. Puoi reimpostare la password per le credenziali Git in qualsiasi momento. Puoi anche rendere le credenziali inattive o eliminarle se non ne hai più bisogno.

> **Note**
>
> Non è possibile selezionare il nome utente e la password per le credenziali Git. IAM genera queste credenziali per aiutarti a garantire che soddisfino gli standard di sicurezza AWS

e proteggano gli archivi in. CodeCommit Puoi scaricare le credenziali una sola volta, nel momento in cui vengono generate. Assicurati di salvare le credenziali in una posizione sicura. Se necessario, puoi reimpostare la password in qualsiasi momento, ma questa operazione invalida le connessioni configurate con la password precedente. Devi riconfigurare le connessioni in modo che utilizzino la nuova password per poterti connettere nuovamente.

Per ulteriori informazioni, consultare i seguenti argomenti:

- Per creare un utente IAM, consulta Crea un utente IAM nel tuo Account AWS.
- Per generare e utilizzare credenziali Git con CodeCommit, consulta Per gli utenti HTTPS che utilizzano le credenziali Git nella Guida per l'AWS CodeCommit utente.

> **Note**
>
> La modifica del nome di un utente IAM dopo la generazione delle credenziali Git non comporta la modifica del nome utente delle credenziali. Il nome utente e la password rimangono invariati e validi.

Per aggiornare le credenziali specifiche del servizio

1. Creare un secondo set di credenziali specifico del servizio in aggiunta al set attualmente in uso.
2. Aggiornare tutte le applicazioni in modo da utilizzare il nuovo set di credenziali e confermare che le applicazioni funzionino.
3. Cambiare lo stato delle credenziali originali in "Inactive" (Non attivo).
4. Verificare che tutte le applicazioni funzionino ancora.
5. Eliminare le credenziali specifiche del servizio non attive.

Usa chiavi SSH e SSH con CodeCommit

Con le connessioni SSH, crei file di chiave pubblici e privati sulla tua macchina locale che Git e Git CodeCommit utilizzano per l'autenticazione SSH. La chiave pubblica va associata all'utente IAM e la chiave privata va archiviata nel computer locale. Per ulteriori informazioni, consultare i seguenti argomenti:

- Per creare un utente IAM, consulta Crea un utente IAM nel tuo Account AWS.
- Per creare una chiave pubblica SSH e associarla a un utente IAM consulta Per le connessioni SSH su Linux, macOS o Unix oppure Per le connessioni SSH su Windows nella Guida per l'utente di AWS CodeCommit .

> **Note**
>
> La chiave pubblica deve essere codificata in formato ssh-rsa o formato PEM. La lunghezza in bit minima della chiave pubblica è di 2.048 bit e la lunghezza massima è di 16.384 bit. Questo valore è separato dalla dimensione del file da caricare. Ad esempio, è possibile generare una chiave a 2.048 bit e il file PEM risultante è lungo 1.679 byte. Se fornisci la chiave pubblica in un altro formato o dimensione, verrà visualizzato un messaggio di errore indicante che il formato non è valido.

Usa HTTPS con l'helper per le AWS CLI credenziali e CodeCommit

In alternativa alle connessioni HTTPS con credenziali Git, puoi consentire a Git di utilizzare una versione con firma crittografica delle tue credenziali utente IAM o del ruolo dell'istanza Amazon EC2 ogni volta che Git deve AWS autenticarsi per interagire con i repository. CodeCommit Questo è l'unico metodo di connessione per i CodeCommit repository che non richiede un utente IAM. Inoltre, questo è il solo metodo che funziona con l'accesso federato e le credenziali temporanee. Per ulteriori informazioni, consultare i seguenti argomenti:

- Per ulteriori informazioni sull'accesso federato, consultare Fornitori di identità e federazione in AWS e Accesso a utenti autenticati esternamente (federazione delle identità).
- Per ulteriori informazioni sulle credenziali provvisorie, consulta Credenziali di sicurezza temporanee in IAM e la pagina relativa all'accesso provvisorio ai repository CodeCommit.

L'helper per AWS CLI le credenziali non è compatibile con altri sistemi di supporto per le credenziali, come Keychain Access o Windows Credential Management. Quando configuri le connessioni HTTPS con l'assistente credenziali, devi tenere presenti ulteriori considerazioni. Per ulteriori informazioni, consulta Per le connessioni HTTPS su Linux, macOS o Unix con il AWS CLI Credential Helper o Connessioni HTTPS su Windows con il Credential Helper nella Guida per l' AWS CLI utente.AWS CodeCommit

Gestire i certificati server in IAM

Per abilitare le connessioni HTTPS al tuo sito Web o alla tua applicazione in AWS, hai bisogno di un certificato SSL/TLS server. Per i certificati in una regione supportata da AWS Certificate Manager (ACM), consigliamo di utilizzare ACM per effettuare il provisioning, la gestione e la distribuzione dei certificati server. Nelle regioni non supportate, è necessario utilizzare IAM come gestore di certificati. Per informazioni sulle regioni supportate da ACM, consulta Endpoint e quote di AWS Certificate Manager nella Riferimenti generali di AWS.

> **Important**
>
> ACM è lo strumento preferito per il provisioning, la gestione e la distribuzione dei certificati del server. Con ACM puoi richiedere un certificato o distribuire un certificato ACM o esterno esistente alle risorse. AWS I certificati forniti da ACM sono gratuiti e vengono automaticamente rinnovati. In una regione supportata è possibile utilizzare ACM per gestire i certificati server dalla console o a livello di programmazione. Per ulteriori informazioni sull'utilizzo di ACM, consulta la Guida per l'utente di AWS Certificate Manager . Per ulteriori informazioni su come richiedere un certificato ACM, consulta Richiesta di un certificato pubblico o Richiesta di un certificato privato nella Guida per l'utente di AWS Certificate Manager . Per ulteriori informazioni sull'importazione di certificati di terza parte in ACM, consulta Importazione di certificati nella Guida per l'utente di AWS Certificate Manager .

Utilizza IAM come gestore di certificati solo quando è necessario il supporto alle connessioni HTTPS in una regione che non è supportata da ACM. IAM crittografa in modo sicuro le chiavi private e archivia la versione crittografata nella memoria dei certificati SSL di IAM. IAM supporta la distribuzione di certificati server in tutte le regioni, ma è necessario ottenere il certificato da un provider esterno per utilizzarlo con. AWS Non è possibile caricare un certificato ACM in IAM. Inoltre, non è possibile gestire i certificati dalla console IAM.

Per ulteriori informazioni sul caricamento di certificati di terze parti in IAM, consulta i seguenti argomenti.

Argomenti

- Carica un certificato server (AWS API)
- AWS Operazioni API per i certificati del server
- Risolvere i problemi relativi ai certificati server

Carica un certificato server (AWS API)

Per caricare un certificato del server in IAM, è necessario fornire il certificato e la chiave privata corrispondente. Quando il certificato non è autofirmato, è necessario fornire anche una catena di certificati. (La catena di certificati non necessaria se si carica un certificato autofirmato). Prima di caricare un certificato, assicurarsi di disporre di tutti questi elementi e di soddisfare i seguenti criteri:

- Il certificato deve essere valido al momento del caricamento. Non è possibile caricare un certificato prima dell'inizio del periodo di validità `NotBefore` o dopo la data di scadenza (la data `NotAfter` del certificato).
- La chiave di accesso non deve essere crittografata. Non è possibile caricare una chiave di accesso privata protetta da password o da passphrase. Per informazioni sulla decodifica di una chiave privata crittografata, consultare Risolvere i problemi relativi ai certificati server.
- Il certificato, la chiave privata e la catena di certificati devono esserlo tutti PEM-encoded. Per informazioni sulla conversione di tali elementi in formato PEM, consultare Risolvere i problemi relativi ai certificati server.

Per utilizzare l'API IAM per caricare un certificato, invia una UploadServerCertificate richiesta. L'esempio seguente mostra come eseguire questa operazione con l'AWS Command Line Interface (AWS CLI). L'esempio presuppone quanto segue:

- Il PEM-encoded certificato è archiviato in un file denominato`Certificate.pem`.
- La catena PEM-encoded di certificati è archiviata in un file denominato`CertificateChain.pem`.
- La PEM-encoded chiave privata non crittografata viene archiviata in un file denominato`PrivateKey.pem`.
- (Facoltativo) Desideri applicare un tag al certificato del server con una coppia chiave-valore. Ad esempio, è possibile aggiungere la chiave tag `Department` e il valore tag `Engineering` per facilitare l'identificazione e l'organizzazione dei certificati.

Per utilizzare il seguente comando esemplificativo, sostituisci questi nomi di file con il tuo. Sostituiscilo *`ExampleCertificate`* con un nome per il certificato caricato. Se desideri etichettare il certificato, sostituisci la coppia chiave-valore *`ExampleKey`* e *`ExampleValue`* contrassegna con i tuoi valori. Digitare il comando su una linea continua. L'esempio seguente include interruzioni di linea e spazi aggiuntivi per agevolare la lettura.

```
aws iam upload-server-certificate --server-certificate-name ExampleCertificate
```

```
                                    --certificate-body file://Certificate.pem
                                    --certificate-chain file://CertificateChain.pem
                                    --private-key file://PrivateKey.pem
                                    --tags '{"Key": "ExampleKey", "Value":
"ExampleValue"}'
```

Quando il comando precedente viene completato, restituisce i metadati relativi al certificato caricati, tra cui il relativo Amazon Resource Name (ARN), il nome descrittivo, l'identificatore (ID), la data di scadenza, i tag e molte altre informazioni.

> **Note**
>
> Se stai caricando un certificato server da utilizzare con Amazon CloudFront, devi specificare un percorso utilizzando l'opzione. `--path` Il percorso deve iniziare con `/cloudfront` e devono includere una barra finale (ad esempio, `/cloudfront/test/`).

Per utilizzare il AWS Tools for Windows PowerShell per caricare un certificato, usa Publish-IAMServerCertificate.

AWS Operazioni API per i certificati del server

Utilizza i seguenti comandi per visualizzare, aggiungere tag, rinominare ed eliminare i certificati server.

- GetServerCertificateUsare per recuperare un certificato. Questa richiesta restituisce il certificato, la la catena di certificati (se ne è stata caricata una) e i metadati sul certificato.

  > **Note**
  >
  > Non è possibile scaricare o recuperare una chiave privata da IAM dopo averla caricata.

- Get-IAMServerCertificateDa utilizzare per recuperare un certificato.
- ListServerCertificatesUsalo per elencare i certificati del server caricati. La richiesta restituisce un elenco che contiene metadati relativi a ciascun certificato.
- Get-IAMServerCertificatesUsalo per elencare i certificati del server caricati.
- TagServerCertificateUsalo per contrassegnare un certificato server esistente.
- Si usa UntagServerCertificate per rimuovere i tag da un certificato del server.

- UpdateServerCertificateDa utilizzare per rinominare un certificato del server o aggiornarne il percorso.

  L'esempio seguente mostra come eseguire questa operazione con l' AWS CLI.

  Per utilizzare il seguente comando di esempio, sostituire i nomi dei certificati precedenti e nuovi e il percorso del certificato e digitare il comando su una riga continua. L'esempio seguente include interruzioni di linea e spazi aggiuntivi per agevolare la lettura.

  ```
  aws iam update-server-certificate --server-certificate-name ExampleCertificate
                                    --new-server-certificate-
  name CloudFrontCertificate
                                    --new-path /cloudfront/
  ```

  Per utilizzare il AWS Tools for Windows PowerShell per rinominare un certificato del server o aggiornarne il percorso, usa. Update-IAMServerCertificate

- DeleteServerCertificateUsare per eliminare un certificato del server.

  Per utilizzare il AWS Tools for Windows PowerShell per eliminare un certificato del server, utilizzare Remove-IAMServerCertificate.

Risolvere i problemi relativi ai certificati server

Prima di poter caricare un certificato su IAM, devi assicurarti che il certificato, la chiave privata e la catena di certificati siano tutti PEM-encoded. È inoltre necessario assicurarsi che la chiave privata non sia crittografata. Fare riferimento agli esempi riportati di seguito.

Example Esempio PEM-encoded di certificato

```
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

Example Esempio PEM-encoded, chiave privata non crittografata

```
-----BEGIN RSA PRIVATE KEY-----
Base64-encoded private key
-----END RSA PRIVATE KEY-----
```

Example Esempio di catena PEM-encoded di certificati

Una catena di certificati contiene uno o più certificati. Puoi utilizzare un editor di testo, il comando di copia in Windows, oppure il comando Linux cat per concatenare i tuoi file del certificato in una catena. Quando includi più certificati, ogni certificato deve certificare il certificato precedente. Puoi farlo concatenando i certificati, incluso il certificato CA radice per ultimo.

L'esempio seguente contiene tre certificati, ma la catena di certificati può contenerne un numero maggiore o minore di certificati.

```
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

Se questi elementi non sono nel formato corretto per il caricamento in IAM, puoi utilizzare OpenSSL per convertirli nel formato corretto.

Per convertire un certificato o una catena di certificati da DER a PEM

Utilizzare il comando OpenSSL x509, come nell'esempio seguente. Nel seguente comando di esempio, sostituisci *Certificate.der* con il nome del file che contiene il DER-encoded certificato. Sostituisci *Certificate.pem* con il nome preferito del file di output che contiene il PEM-encoded certificato.

```
openssl x509 -inform DER -in Certificate.der -outform PEM -out Certificate.pem
```

Per convertire una chiave privata da DER a PEM

Utilizzare il comando OpenSSL rsa, come nell'esempio seguente. Nel seguente comando di esempio, sostituisci *PrivateKey.der* con il nome del file che contiene la tua chiave DER-encoded privata. Sostituisci *PrivateKey.pem* con il nome preferito del file di output che contiene la chiave PEM-encoded privata.

```
openssl rsa -inform DER -in PrivateKey.der -outform PEM -out PrivateKey.pem
```

Per decrittografare una chiave privata crittografata (rimuovere la password o la passphrase)

Utilizzare il comando OpenSSL rsa, come nell'esempio seguente. Per utilizzare il seguente comando di esempio, sostituire *EncryptedPrivateKey.pem* con il nome del file che contiene la chiave privata crittografata. Sostituisci *PrivateKey.pem* con il nome preferito del file di output per contenere la chiave privata PEM-encoded non crittografata.

```
openssl rsa -in EncryptedPrivateKey.pem -out PrivateKey.pem
```

Per convertire un bundle di certificati da PKCS # 12 (PFX) a PEM

Utilizzare il comando OpenSSL pkcs12, come nell'esempio seguente. Nel seguente comando di esempio, sostituire *CertificateBundle.p12* con il nome del file che contiene il bundle di certificati con codifica PKCS#12. Sostituisci *CertificateBundle.pem* con il nome preferito del file di output per contenere il pacchetto di PEM-encoded certificati.

```
openssl pkcs12 -in CertificateBundle.p12 -out CertificateBundle.pem -nodes
```

Per convertire un bundle di certificati da PKCS#7 a PEM

Utilizzare il comando OpenSSL pkcs7, come nell'esempio seguente. Nel seguente comando di esempio, sostituire *CertificateBundle.p7b* con il nome del file che contiene il bundle di certificati con codifica PKCS#7. Sostituisci *CertificateBundle.pem* con il nome preferito del file di output per contenere il pacchetto di PEM-encoded certificati.

```
openssl pkcs7 -in CertificateBundle.p7b -print_certs -out CertificateBundle.pem
```

Gruppi di utenti IAM

Un gruppo di utenti IAM è una raccolta di utenti IAM. I gruppi di utenti consentono di specificare le autorizzazioni per più utenti e quindi la gestione delle autorizzazioni per quegli utenti può essere

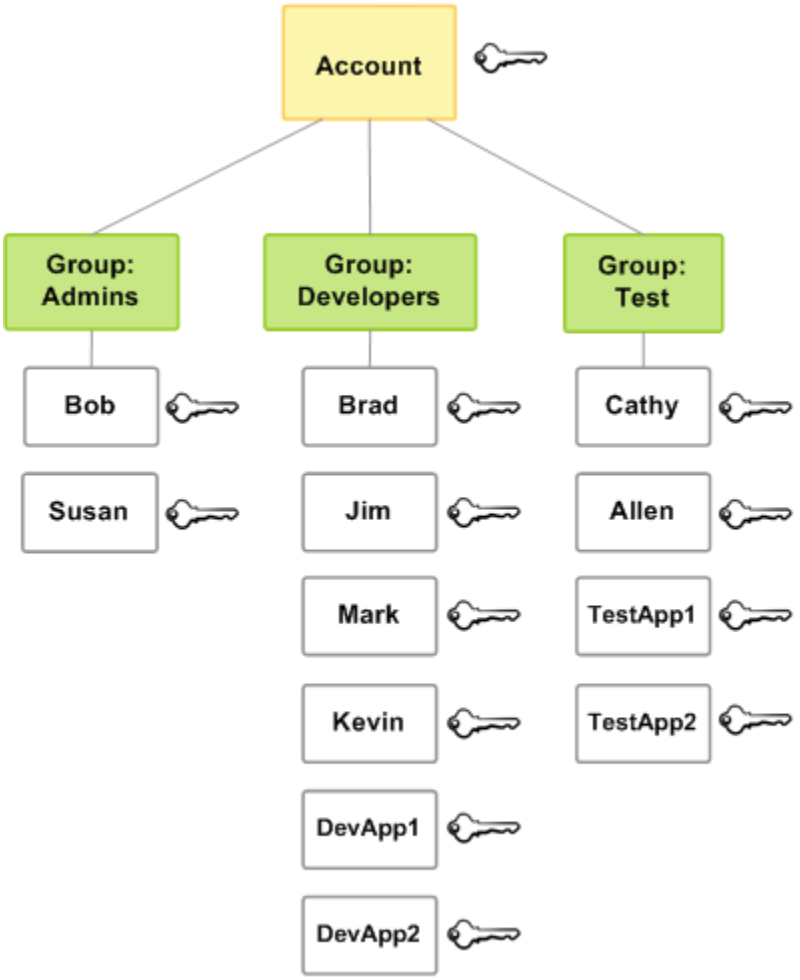
più facile. Ad esempio, potresti avere un gruppo di utenti chiamato Amministratori e concedere a tale gruppo di utenti le autorizzazioni tipiche degli amministratori. Qualsiasi utente all'interno di tale gruppo dispone automaticamente delle autorizzazioni del gruppo Amministratori. Se un nuovo utente entra a far parte dell'organizzazione e necessita dei privilegi di amministratore, puoi concedere le autorizzazioni appropriate aggiungendo l'utente al gruppo di utenti Amministratori. Se una persona cambia mansione all'interno dell'organizzazione, invece di modificare le autorizzazioni dell'utente puoi rimuoverlo dai vecchi gruppi IAM e aggiungerlo a nuovi gruppi IAM appropriati.

Puoi collegare una policy basata sull'identità a un gruppo di utenti in modo che tutti gli utenti del gruppo ricevano le autorizzazioni della policy. Non è possibile identificare un gruppo di utenti come `Principal` in una policy (ad esempio una policy basata sulle risorse) perché i gruppi si riferiscono alle autorizzazioni, non all'autenticazione, e i principali sono entità IAM autenticate. Per ulteriori informazioni sui tipi di policy, consulta Identity-based politiche e politiche basate sulle risorse.

Queste sono alcune delle funzionalità importanti dei gruppi IAM:

- Un gruppo di utenti può contenere molti utenti e un utente può appartenere a più gruppi di utenti.
- I gruppi di utenti non possono essere nidificati; possono contenere solo utenti, non altri gruppi IAM.
- Non esiste alcun gruppo di utenti predefinito che include automaticamente tutti gli utenti nell' Account AWS. Se desideri disporre di un gruppo di utenti di questo tipo, è necessario crearlo e assegnarvi ogni nuovo utente.
- Il numero e la dimensione delle risorse IAM in un Account AWS, ad esempio il numero di gruppi e il numero di gruppi di cui un utente può essere membro, sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Il diagramma seguente mostra un semplice esempio di una piccola azienda. Il proprietario dell'azienda crea un gruppo di utenti `Admins` perché gli utenti possano creare e gestire altri utenti mentre l'azienda cresce. Il gruppo di utenti `Admins` crea un gruppo di utenti `Developers` e un gruppo di utenti `Test`. Ciascuno di questi gruppi IAM è composto da utenti (umani e applicazioni) con cui interagiscono AWS (Jim, Brad, DevApp 1 e così via). Ogni utente dispone di un singolo set di credenziali di sicurezza. In questo esempio, ogni utente appartiene a un singolo gruppo. Tuttavia, gli utenti possono appartenere a più gruppi IAM.



Dopo aver creato un gruppo di utenti, puoiVisualizzare i gruppi IAM,Collegare una policy a un gruppo di utenti IAM, eRidenominare un gruppo di utenti IAM.

Creare gruppi IAM

> **Note**
>
> Come procedura consigliata, consigliamo di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee. Se segui le best practice, non gestisci utenti e gruppi IAM. Gli utenti e i gruppi sono invece gestiti all'esterno AWS e possono accedere alle AWS risorse come identità federata. Un'identità federata è un utente dell'elenco utenti aziendale, un provider di identità Web, il AWS Directory Service, la directory Identity Center o qualsiasi utente che accede AWS ai servizi utilizzando le credenziali fornite tramite un'origine di identità. Le identità

federate utilizzano i gruppi definiti dal rispettivo gestore di identità. Se lo utilizzi Centro identità AWS IAM, consulta Gestisci le identità in IAM Identity Center nella Guida per l'Centro identità AWS IAM utente per informazioni sulla creazione di utenti e gruppi in IAM Identity Center.

Crei gruppi IAM per gestire le autorizzazioni di accesso per più utenti con ruoli o responsabilità simili. Collegando le policy a questi gruppi, puoi concedere o revocare le autorizzazioni per interi insiemi di utenti. Ciò semplifica la manutenzione delle policy di sicurezza, poiché le modifiche apportate alle autorizzazioni di un gruppo vengono applicate automaticamente a tutti i membri di quel gruppo, garantendo un controllo uniforme degli accessi. Dopo aver creato il gruppo, assegna le autorizzazioni in base alle attività previste per gli utenti IAM che ne faranno parte, quindi aggiungi gli utenti IAM al gruppo.

Per informazioni sulle autorizzazioni richieste per la creazione di un gruppo IAM, consulta Autorizzazioni necessarie per accedere alle risorse IAM.

Per creare un gruppo IAM e collegare policy

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione seleziona Gruppi di utenti, quindi Crea gruppo.
3. In Nome gruppo di utenti, digita il nome del gruppo.

   > **Note**
   >
   > Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote. I nomi dei gruppi possono essere una combinazione di un massimo di 128 lettere, cifre e i seguenti caratteri: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), trattino basso (_) e trattino (-). I nomi devono essere univoci all'interno di un account. Non viene fatta distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare gruppi denominati sia **`ADMINS`** che **`admins`**.

4. Nell'elenco degli utenti, seleziona la casella di controllo per ogni utente che desideri aggiungere al gruppo.

5. Nell'elenco di tutte le policy, selezionare la casella di controllo per ogni policy che si desidera applicare a tutti i membri del gruppo.
6. Seleziona Crea gruppo.

AWS CLI

Esegui il comando seguente:

- aws iam create-group

API

Chiamare l'operazione seguente:

- CreateGroup

Visualizzare i gruppi IAM

Puoi elencare tutti i gruppi IAM nel tuo account, elencare gli utenti in un gruppo di utenti ed elencare i gruppi IAM a cui un utente appartiene. Se utilizzi la CLI o l'API, puoi elencare tutti i gruppi IAM con un determinato prefisso di percorso.

Console

Per elencare tutti i gruppi IAM nell'account:

- Nel pannello di navigazione, scegli Gruppi di utenti.

Per elencare gli utenti IAM in un determinato gruppo IAM:

- Nel pannello di navigazione, seleziona Gruppi di utenti. Quindi, scegli il nome del gruppo per aprire la pagina dei dettagli del gruppo. Consulta la scheda Utenti per vedere l'appartenenza al gruppo.

Per elencare tutti i gruppi IAM in cui si trova un utente:

- Nel pannello di navigazione, seleziona Utenti. Quindi seleziona il nome utente per aprire la pagina dei dettagli dell'utente. Scegli la scheda Gruppi per visualizzare un elenco dei gruppi a cui appartiene l'utente.

AWS CLI

Per elencare tutti i gruppi IAM nell'account:

- aws iam list-groups

Per elencare gli utenti in un determinato gruppo IAM:

- aws iam get-group

Per elencare tutti i gruppi IAM in cui si trova un utente:

- era aim list-groups-for-user

API

Per elencare tutti i gruppi IAM nell'account:

- ListGroups

Per elencare gli utenti in un determinato gruppo IAM:

- GetGroup

Per elencare tutti i gruppi IAM in cui si trova un utente:

- ListGroupsForUser

Modificare gli utenti nei gruppi IAM

Usa i gruppi IAM per applicare le stesse policy di autorizzazione a più utenti contemporaneamente. Potrai quindi aggiungere o rimuovere utenti da un gruppo IAM. Questa funzione è utile quando le persone arrivano e lasciano l'organizzazione.

Revisione delle policy di accesso

Prima di rimuovere un gruppo, utilizza la pagina dei dettagli del gruppo per esaminarne i membri (utenti IAM), la scheda Autorizzazioni per vedere le policy collegate e la scheda Ultimo accesso per esaminare le attività recenti a livello di servizio. Questo aiuta a evitare di revocare involontariamente l'accesso a un principale (persona o applicazione) che lo utilizza. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Aggiunta di un utente IAM a un gruppo IAM

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Gruppi di utenti, quindi scegli il nome del gruppo.
3. Seleziona la scheda Utenti, quindi scegli Aggiungi utenti. Selezionare la casella di controllo accanto agli utenti che si desidera aggiungere.
4. Scegli Aggiungi utenti.

AWS CLI

Esegui il comando seguente:

- `aws iam add-user-to-group`

API

Chiamare l'operazione seguente:

- `AddUserToGroup`

Rimozione di un utente IAM da un gruppo IAM

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione, scegli Gruppi di utenti, quindi scegli il nome del gruppo.
3. Scegli la scheda Users (Utenti); Seleziona la casella di controllo accanto agli utenti che desideri rimuovere e quindi scegli Rimuovi utenti.

AWS CLI

Esegui il comando seguente:

- `aws iam remove-user-from-group`

API

Chiamare l'operazione seguente:

- `RemoveUserFromGroup`

Collegare una policy a un gruppo di utenti IAM

È possibile allegare una politica AWS gestita, ovvero una politica prescritta fornita da, AWS a un gruppo di utenti, come spiegato nei passaggi seguenti. Per collegare una policy gestita dal cliente, ovvero una policy con autorizzazioni personalizzate da te creata, è prima necessario creare la policy. Per ulteriori informazioni sulla creazione di policy gestite dal cliente, consulta Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente.

Per ulteriori informazioni sulle autorizzazioni e sulle policy, consulta Gestione degli accessi per AWS risorse.

Per collegare una policy a un gruppo IAM

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Gruppi di utenti, quindi scegli il nome del gruppo.
3. Scegli la scheda Autorizzazioni.
4. Seleziona Aggiungi autorizzazioni, quindi seleziona Collega policy.
5. Le policy correnti collegate al gruppo di utenti vengono visualizzate nell'elenco Policy di autorizzazione correnti. Nell'elenco Altre policy di autorizzazioni, seleziona la casella di

controllo accanto al nome delle policy da collegare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy in base al tipo e al nome della policy.

6. Seleziona la policy che desideri collegare al tuo gruppo IAM e scegli Collega policy.

AWS CLI

Esegui il comando seguente:

- `aws iam attach-group-policy`

API

Chiamare l'operazione seguente:

- `AttachGroupPolicy`

Ridenominare un gruppo di utenti IAM

Quando modifichi il nome o il percorso di un gruppo di utenti, si verificano i seguenti eventi:

- Qualsiasi policy associata al gruppo di utenti resta con il gruppo con il nuovo nome.
- Il gruppo di utenti mantiene tutti i suoi utenti con il nuovo nome.
- L'ID univoco del gruppo di utenti rimane invariato. Per ulteriori informazioni su unique IDs, consultaIdentificatori univoci.

IAM non aggiorna automaticamente le policy che fanno riferimento al gruppo di utenti come risorsa per l'utilizzo del nuovo nome. Pertanto, è necessario prestare attenzione quando si rinomina un gruppo di utenti. Prima di rinominare il gruppo di utenti, è necessario verificare manualmente tutte le policy per individuare quelle in cui tale gruppo viene menzionato in base al nome. Supponiamo ad esempio che Bob sia il responsabile dell'area test dell'organizzazione. Bob dispone di una policy associata alla sua entità utente IAM che gli permette di aggiungere e rimuovere utenti dal gruppo di utenti Test. Se un amministratore cambia il nome del gruppo di utenti (o modifica il percorso del gruppo), deve anche aggiornare la policy associata a Bob per l'utilizzo del nuovo nome o percorso. In caso contrario Bob non potrà aggiungere o rimuovere gli utenti dal gruppo di utenti.

Per trovare le policy che fanno riferimento a un gruppo IAM come risorsa:

1. Nel pannello di navigazione della console IAM, scegli Policy.
2. Ordina in base alla colonna Tipo per individuare le tue policy personalizzate gestite dal cliente.
3. Scegli il nome della policy da modificare.
4. Scegli la scheda Autorizzazioni e quindi Riepilogo.
5. Seleziona IAM dall'elenco di servizi, se disponibile.
6. Cerca il nome del gruppo di utenti nella colonna Risorsa.
7. Seleziona Modifica per modificare il nome del gruppo di utenti nella policy.

Come modificare il nome di un gruppo di utenti IAM

Console

1. Nel pannello di navigazione, scegli Gruppi utenti e poi scegli il nome del gruppo.
2. Scegli Modifica. Digita il nuovo nome del gruppo di utenti e scegli Salva modifiche.

AWS CLI

Esegui il comando seguente:

- aws iam update-group

API

Chiamare l'operazione seguente:

- UpdateGroup

Eliminazione di un gruppo IAM

Quando elimini un gruppo IAM nella console, questa rimuove automaticamente tutti i membri del gruppo, scollega tutte le policy gestite collegate ed elimina tutte le policy inline. Tuttavia, poiché IAM non elimina automaticamente le policy che fanno riferimento al gruppo IAM come risorsa, è necessario prestare attenzione quando si elimina un gruppo IAM. Prima di poter eliminare il gruppo IAM, è necessario verificare manualmente le policy per individuare quelle in cui tale gruppo

viene menzionato per nome. Ad esempio, John, il responsabile del team di test, dispone di una policy collegata alla sua entità utente IAM che gli consente di aggiungere e rimuovere utenti dal gruppo Test. Se un amministratore elimina il gruppo, deve eliminare anche la policy collegata a John. Altrimenti, se l'amministratore ricrea il gruppo eliminato e gli assegna lo stesso nome, le autorizzazioni di John rimangono valide, anche se ha lasciato il Test Team.

Al contrario, quando si utilizza la CLI, l'SDK o l'API per eliminare un gruppo di utenti, è necessario prima rimuovere gli utenti del gruppo. Quindi, si eliminano le policy inline eventualmente incorporate nel gruppo IAM. A questo punto è necessario scollegare le policy gestite collegate al gruppo. Una volta fatto questo è possibile eliminare il gruppo IAM.

Console

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Gruppi di utenti.
3. Nell'elenco dei gruppi IAM, seleziona la casella di controllo accanto ai nomi dei gruppi IAM da eliminare. Puoi utilizzare la casella di ricerca per filtrare l'elenco dei gruppi IAM in base al tipo, alle autorizzazioni e al nome del gruppo.
4. Scegli Elimina.
5. Nella casella di conferma, se desideri eliminare un singolo gruppo, digita il nome del gruppo e scegli Elimina. Se desideri eliminare più gruppi, digita il numero di gruppi IAM da eliminare seguito da **`user groups`** e scegli Elimina. Ad esempio, se desideri eliminare tre gruppi, digita **`3 user groups`**.

AWS CLI

1. Rimuovi tutti gli utenti dal gruppo IAM.
   - aws iam get-group (per ottenere l'elenco degli utenti nel gruppo IAM) e aws iam remove-user-from-group (per rimuovere un utente dal gruppo IAM)
2. Elimina tutte le policy inline incorporate nel gruppo IAM.
   - aws iam list-group-policies (per ottenere un elenco delle politiche in linea del gruppo IAM) e aws iam delete-group-policy (per eliminare le politiche in linea del gruppo IAM)
3. Scollega tutte le policy gestite collegate al gruppo IAM.

- aws iam list-attached-group-policies (per ottenere un elenco delle politiche gestite allegate al gruppo IAM) e aws iam detach-group-policy (per scollegare una politica gestita dal gruppo IAM)

4. Elimina il gruppo IAM.

   - aws iam delete-group

API

1. Rimuovi tutti gli utenti dal gruppo IAM.

   - GetGroup(per ottenere l'elenco degli utenti nel gruppo IAM) e RemoveUserFromGroup(per rimuovere un utente dal gruppo IAM)

2. Elimina tutte le policy inline incorporate nel gruppo IAM.

   - ListGroupPolicies(per ottenere un elenco delle politiche in linea del gruppo IAM) e DeleteGroupPolicy(per eliminare le politiche in linea del gruppo IAM)

3. Scollega tutte le policy gestite collegate al gruppo IAM.

   - ListAttachedGroupPolicies(per ottenere un elenco delle politiche gestite allegate al gruppo IAM) e DetachGroupPolicy(per scollegare una politica gestita dal gruppo IAM)

4. Elimina il gruppo IAM.

   - DeleteGroup

Ruoli IAM

Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. Un ruolo IAM è simile a un utente IAM, in quanto è un'identità AWS con policy di autorizzazioni che determinano ciò che l'identità può e non può fare in AWS. Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque. Inoltre, un ruolo non ha credenziali a lungo termine standard associate (password o chiavi di accesso). Invece, quando assumi un ruolo, ti fornisce credenziali di sicurezza temporanee per la tua sessione di ruolo.

È possibile utilizzare i ruoli per delegare l'accesso a utenti, applicazioni o servizi che normalmente non hanno accesso alle AWS risorse. Ad esempio, potresti voler concedere agli utenti del tuo AWS

account l'accesso a risorse di cui normalmente non dispongono o concedere agli utenti di un Account AWS accesso alle risorse di un altro account. Oppure potresti voler consentire a un'app mobile di utilizzare AWS le risorse, ma non incorporare AWS le chiavi all'interno dell'app (dove possono essere difficili da aggiornare e dove gli utenti possono potenzialmente estrarle). A volte è necessario concedere AWS l'accesso agli utenti che hanno già identità definite all'esterno AWS, ad esempio nella directory aziendale. In alternativa, è possibile concedere l'accesso all'account a terze parti in modo che possano eseguire un controllo sulle proprie risorse.

Per questi scenari, puoi delegare l'accesso alle AWS risorse utilizzando un ruolo IAM. Questa sezione introduce i ruoli e i diversi modi in cui è possibile utilizzarli, quando e come selezionare gli approcci, come creare, gestire, cambiare (o assumere) ed eliminare i ruoli.

Account access manager è una funzionalità IAM che consente di assegnare centralmente i ruoli IAM tra gli account dell'organizzazione agli utenti e ai gruppi di IAM Identity Center. Account Access Manager offre alla tua forza lavoro l'accesso tramite i ruoli IAM con il loro set completo di funzionalità. Puoi usarlo insieme ai set di autorizzazioni di IAM Identity Center o da solo.

> **Note**
>
> Quando crei il tuo per la prima volta Account AWS, per impostazione predefinita non viene creato alcun ruolo. Man mano che aggiungi servizi al tuo account, questi possono aggiungere ruoli collegati ai servizi per supportarne i casi d'uso.
> Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo di eseguire un'azione per conto dell'utente. Service-linked i ruoli vengono visualizzati nell'utente Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.
> Prima di poter eliminare i ruoli collegati ai servizi, devi eliminare le risorse associate. Questa procedura protegge le risorse poiché impedisce la rimozione involontaria dell'autorizzazione ad accedere alle risorse.
> Per informazioni sui servizi che supportano l'utilizzo di ruoli collegati ai servizi, vedi AWS servizi supportati da IAM e cerca i servizi che hanno Sì nella colonna Service-Linked Ruolo. Scegliere Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Argomenti

- Quando creare un utente IAM invece di un ruolo

- Termini e concetti dei ruoli
- Risorse aggiuntive
- Problema del "confused deputy"
- Scenari comuni per i ruoli IAM
- Creazione di ruoli IAM
- Gestione del ruolo IAM
- Metodi per assumere un ruolo

Quando creare un utente IAM invece di un ruolo

Ti consigliamo di utilizzare gli utenti IAM solo per casi d'uso non supportati dalla federazione delle identità. Alcuni dei casi d'uso sono i seguenti:

- Carichi di lavoro che non possono utilizzare ruoli IAM: è possibile eseguire un carico di lavoro da una posizione che deve accedere a AWS. In alcune situazioni, non puoi utilizzare i ruoli IAM per fornire credenziali temporanee, ad esempio per i plugin. WordPress In queste situazioni, per autenticarti a AWS usa le chiavi di accesso a lungo termine dell'utente IAM per quel carico di lavoro.
- Third-party AWS clienti: se utilizzi strumenti che non supportano l'accesso con IAM Identity Center, come AWS client o fornitori di terze parti che non sono ospitati su IAM AWS, utilizza le chiavi di accesso a lungo termine degli utenti IAM.
- AWS CodeCommit accesso: se utilizzi CodeCommit per archiviare il tuo codice, puoi utilizzare un utente IAM con chiavi SSH o credenziali specifiche del servizio CodeCommit per l'autenticazione nei tuoi repository. Si consiglia di eseguire questa operazione oltre a utilizzare un utente di IAM Identity Center per l'autenticazione normale. Gli utenti di IAM Identity Center sono le persone della tua forza lavoro che hanno bisogno di accedere alle tue o alle tue applicazioni cloud. Account AWS Per consentire agli utenti di accedere ai tuoi CodeCommit repository senza configurare gli utenti IAM, puoi configurare l'utilità. git-remote-codecommit Per ulteriori informazioni su IAM e CodeCommit, consulta. Credenziali IAM per CodeCommit: credenziali Git, chiavi SSH e chiavi di accesso AWS Per ulteriori informazioni sulla configurazione dell'git-remote-codecommitutilità, consulta Connessione ai AWS CodeCommit repository con credenziali rotanti nella Guida per l'utente. AWS CodeCommit
- Accesso ad Amazon Keyspaces (per Apache Cassandra): in una situazione in cui non è possibile utilizzare gli utenti in IAM Identity Center, ad esempio per scopi di test per la compatibilità con

Cassandra, puoi utilizzare un utente IAM con credenziali specifiche del servizio per l'autenticazione con Amazon Keyspaces. Gli utenti di IAM Identity Center sono le persone della tua forza lavoro che hanno bisogno di accedere alle tue Account AWS o alle tue applicazioni cloud. Puoi anche connetterti ad Amazon Keyspaces utilizzando credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di credenziali temporanee per connettersi ad Amazon Keyspaces utilizzando un ruolo IAM e il plugin SIGv4 nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).

- Accesso di emergenza: in una situazione in cui non puoi accedere al tuo provider di identità e devi intervenire nel tuo Account AWS. Stabilire l'accesso di emergenza per gli utenti IAM può far parte del tuo piano di resilienza. Si consiglia di controllare e proteggere le credenziali degli utenti di emergenza con l'autenticazione a più fattori (MFA).

Termini e concetti dei ruoli

Di seguito sono elencati alcuni termini di base per aiutarti a iniziare a utilizzare i ruoli.

Ruolo

UN'identità IAM che puoi creare nell'account che ha le autorizzazioni specifiche. Un ruolo IAM presenta alcune analogie con un utente IAM. Ruoli e utenti sono entrambi identità AWS con policy di autorizzazioni che determinano ciò che l'identità può o non può fare in AWS. Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque. Inoltre, un ruolo non ha credenziali a lungo termine standard associate (password o chiavi di accesso). Invece, quando assumi un ruolo, ti fornisce credenziali di sicurezza temporanee per la tua sessione di ruolo.

I ruoli possono essere assunti da:

- Un utente IAM nello stesso Account AWS o in un altro Account AWS
- Ruoli IAM nello stesso account
- Principali del servizio, da utilizzare con AWS servizi e funzionalità come:
  - Servizi che consentono di eseguire codice su servizi di calcolo, come Amazon EC2 o AWS Lambda
  - Funzionalità che eseguono azioni sulle tue risorse per tuo conto, come la replica di oggetti Amazon S3
  - Servizi che forniscono credenziali di sicurezza temporanee alle applicazioni eseguite all'esterno AWS, come IAM Roles Anywhere o Amazon ECS Anywhere

- Un utente esterno autenticato da un gestore dell'identità digitale (IdP) compatibile con SAML 2.0 o OpenID Connect

AWS ruolo del servizio

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

AWS ruolo collegato al servizio

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo di eseguire un'azione per conto dell'utente. Service-linked i ruoli vengono visualizzati nell'utente Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

> **Note**
>
> Se stai già utilizzando un servizio quando inizia a supportare i ruoli collegati al servizio, potresti ricevere un'e-mail che annuncia un nuovo ruolo nel tuo account. In questo caso, il servizio ha creato automaticamente il ruolo collegato al servizio nel tuo account. Non è necessario compiere alcuna operazione per supportare questo ruolo e non è necessario eliminarlo manualmente. Per ulteriori informazioni, consulta È apparso un nuovo ruolo nel mio AWS account.

Per informazioni sui servizi che supportano l'utilizzo di ruoli collegati ai servizi, vedi AWS servizi supportati da IAM e cerca i servizi che hanno Sì nella colonna Service-Linked Ruolo. Scegliere Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio. Per ulteriori informazioni, consulta Creare un ruolo collegato ai servizi.

Concatenazione del ruolo

La concatenazione dei ruoli si verifica quando si utilizza un ruolo per assumere un secondo ruolo. È possibile eseguire il concatenamento dei ruoli Console di gestione AWS tramite il cambio di ruolo AWS CLI, l'API o l'API. Ad esempio, `RoleA` dispone dell'autorizzazione per assumere il ruolo `RoleB`. È possibile consentire a User1 di assumere `RoleA` utilizzando le proprie credenziali utente a lungo termine nell' AssumeRoleoperazione API. Questa restituisce le credenziali a breve termine del ruolo `RoleA`. Con la concatenazione del ruolo, puoi utilizzare le credenziali a breve termine del ruolo `RoleA` per abilitare l'Utente1 ad assumere il ruolo `RoleB`.

Quando assumi un ruolo, puoi passare un tag di sessione e impostare il tag come transitivo. I tag di sessione transitivi vengono passati a tutte le sessioni successive in una concatenazione del ruolo. Per ulteriori informazioni sui tag di sessione, consulta Passa i tag della sessione in AWS STS.

Il concatenamento dei ruoli limita la tua Console di gestione AWS sessione di ruolo AWS CLI o quella dell' AWS API a un massimo di un'ora. Si applica indipendentemente dalla durata massima della sessione configurata per i singoli ruoli. Quando utilizzi l'operazione AssumeRole API per assumere un ruolo, puoi specificare la durata della sessione di ruolo con il `DurationSeconds` parametro. Puoi specificare un valore di parametro fino a 43200 secondi (12 ore), che dipende dall'impostazione della durata massima della sessione per il tuo ruolo. Tuttavia, se assumi un ruolo utilizzando la concatenazione dei ruoli e fornisci un valore del parametro `DurationSeconds` maggiore di un'ora, l'operazione ha esito negativo.

Per informazioni sul passaggio a un ruolo in Console di gestione AWS, vederePassare da un utente a un ruolo IAM (console).

Delega

La concessione delle autorizzazioni a un altro utente per permettere l'accesso alle risorse di controllo. La delega comporta la configurazione di un trust tra due account. Il primo è l'account proprietario della risorsa (l'account che concede fiducia). Il secondo è l'account che contiene gli utenti che devono accedere alla risorsa (l'account attendibile). L'account a cui viene concessa fiducia e l'account che concede fiducia possono essere uno dei seguenti:

- Lo stesso account.
- Account diversi che sono comunque sotto il controllo della tua organizzazione.
- Due account di proprietà di organizzazioni diverse.

Per delegare l'autorizzazione per accedere a una risorsa, crea un ruolo IAM nell'account che concede fiducia che ha due policy collegate. Le policy di autorizzazioni concedono all'utente del ruolo le autorizzazioni necessarie per eseguire le attività previste sulla risorsa. La policy di attendibilità specifica quali membri degli account a cui viene concessa fiducia sono autorizzati ad assumere il ruolo.

Quando si crea una policy di fiducia, non è possibile specificare un carattere jolly (*) come parte di un ARN nell'elemento principale. La policy di affidabilità è associata al ruolo nell'account che concede fiducia e rappresenta una metà delle autorizzazioni. L'altra metà è una policy delle autorizzazioni collegata all'utente nell'account a cui viene concessa fiducia che consente a

quell'utente di passare al ruolo o di assumerlo. Un utente che assume un ruolo temporaneamente cede le proprie autorizzazioni e ottiene le autorizzazioni del ruolo. Quando l'utente esce o termina l'utilizzo del ruolo, le autorizzazioni originali dell'utente vengono ripristinate. Un parametro aggiuntivo chiamato external ID contribuisce a garantire sicuro l'uso dei ruoli tra gli account che non vengono controllati dalla stessa organizzazione.

Policy di trust

Documento di policy JSON in cui si definiscono i principali considerati attendibili per assumere il ruolo. Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo in IAM. I principali che è possibile specificare nella policy di attendibilità includono utenti, ruoli, account e servizi. Per ulteriori informazioni, consulta How to use trust policies in IAM roles nel Blog sulla sicurezza di AWS .

Ruolo per l'accesso tra account

Un ruolo che concede l'accesso alle risorse in un account a un principale affidabile in un diverso account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, alcuni AWS servizi consentono di allegare una policy direttamente a una risorsa (invece di utilizzare un ruolo come proxy). Si chiamano policy basate sulle risorse e puoi utilizzarle per concedere ai mandanti un altro Account AWS accesso alla risorsa. Alcune di queste risorse includono bucket Amazon Simple Storage Service (S3), vault Amazon Glacier, argomenti Amazon Simple Notification Service (SNS) e code Amazon Simple Queue Service (SQS). Per informazioni su quali servizi supportano le policy basate su risorse, consulta AWS servizi supportati da IAM. Per ulteriori informazioni sulle policy basate sulle risorse, consulta Accesso alle risorse multi-account in IAM.

Risorse aggiuntive

Le seguenti risorse possono rivelarsi utili per saperne di più sulla terminologia di IAM relativa ai ruoli IAM.

- I committenti sono entità in grado di eseguire azioni e AWS accedere alle risorse. Un principale può essere un Utente root dell'account AWS utente IAM o un ruolo. Un principal che rappresenta l'identità di un AWS servizio è un service principal. Utilizza l'elemento Principal nelle policy di attendibilità per il ruolo per definire i principali attendibili per assumere il ruolo.

  Per ulteriori informazioni ed esempi di principali a cui è possibile consentire l'assunzione di un ruolo, consulta AWS Elementi della policy JSON: Principale.

- La federazione delle identità crea una relazione di fiducia tra un provider di identità esterno e AWS. Puoi utilizzare il tuo provider OpenID Connect (OIDC) o Security Assertion Markup Language (SAML) 2.0 esistente per gestire chi può accedere alle risorse AWS . Quando si utilizzano OIDC e SAML 2.0 per configurare una relazione di fiducia tra questi provider di identità esterni AWS , all'utente viene assegnato un ruolo IAM. e riceve credenziali provvisorie che gli consentono di accedere alle tue risorse AWS .

  Per ulteriori informazioni sui principali federati, consulta Fornitori di identità e federazione in AWS.

- I federated principal sono identità esistenti provenienti dalla directory degli utenti aziendali o da Directory Service un provider OIDC. AWS assegna un ruolo a un principale federato quando l'accesso viene richiesto tramite un provider di identità. ???

  Per ulteriori informazioni sui principali federati SAML e OIDC, consulta Sessioni e ruoli dell'utente federato.

- Le policy di autorizzazione sono policy basate sull'identità che definiscono le azioni e le risorse che il ruolo può utilizzare. Il documento è scritto in base alle regole del linguaggio della policy IAM.

  Per ulteriori informazioni, consulta Riferimento alla policy JSON IAM.

- Il limite delle autorizzazioni è una funzione avanzata in cui le policy vengono utilizzate per limitare il numero massimo di autorizzazioni che una policy basata su identità può concedere a un ruolo. Non è possibile applicare un limite delle autorizzazioni a un ruolo collegato al servizio.

  Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM.

Problema del "confused deputy"

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Per evitare che ciò accada, AWS fornisce strumenti che ti aiutano a proteggere il tuo account se fornisci a terzi (i cosiddetti cross-account) o ad altri AWS servizi (noti come cross-service) l'accesso alle risorse del tuo account.

A volte, potresti dover concedere a terzi l'accesso alle tue AWS risorse (accesso delegato). Ad esempio, decidete di assumere una società terza chiamata Example Corp per monitorare Account AWS e ottimizzare i costi. Per tenere traccia delle vostre spese giornaliere, Example Corp deve accedere alle vostre AWS risorse. Example Corp controlla anche molti altri Account AWS per altri clienti. Puoi utilizzare un ruolo IAM per stabilire una relazione di fiducia tra il tuo account

Account AWS e quello di Example Corp. Un aspetto importante di questo scenario è l'ID esterno, un identificativo facoltativo che è possibile utilizzare in una policy di attendibilità del ruolo IAM per indicare chi può assumere il ruolo. La funzione principale dell'ID esterno è quella di risolvere e prevenire il problema del "confused deputy" (delegato confuso).

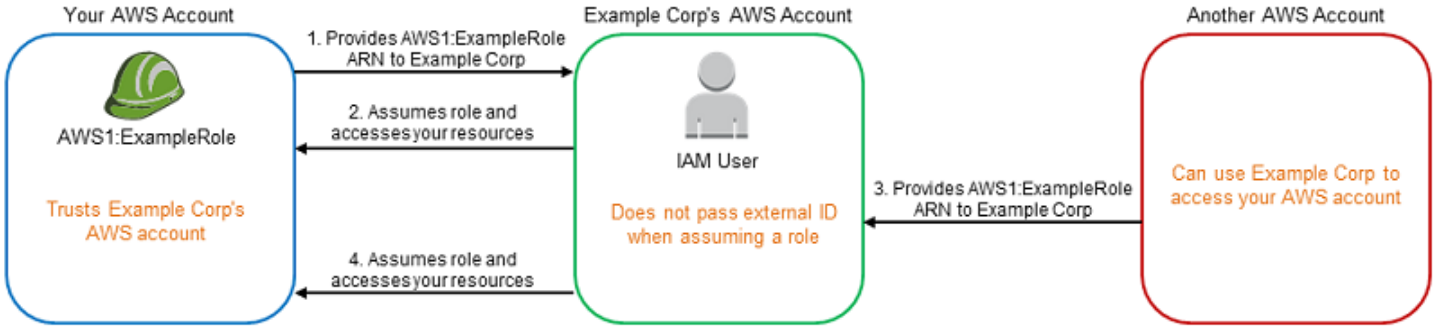
Alcuni AWS servizi (servizi di chiamata) utilizzano il proprio AWS service principal per accedere alle AWS risorse di altri AWS servizi (chiamati servizi). In alcune di queste interazioni di servizio è possibile configurare i servizi di chiamata in modo che comunichino con le risorse di un servizio chiamato in un altro modo Account AWS. Un esempio di ciò è la configurazione AWS CloudTrail per la scrittura su un bucket Amazon S3 centrale che si trova in un altro. Account AWS Al servizio di chiamata viene concesso l'accesso al bucket S3 utilizzando la policy del bucket S3 aggiungendo un'istruzione allow for. CloudTrail `cloudtrail.amazonaws.com`

Quando un responsabile di AWS servizio di un servizio chiamante accede a una risorsa da un servizio chiamato, la politica delle risorse del servizio chiamato autorizza solo il responsabile del servizio e non l'attore che ha configurato il AWS servizio chiamante. Ad esempio, un bucket S3 che si fida del responsabile del CloudTrail servizio senza condizioni potrebbe ricevere CloudTrail i log configurati da un amministratore fidato, ma anche CloudTrail i log da un attore non autorizzato Account AWS che lo utilizza Account AWS, se conosce il nome del bucket S3.

Il confuso problema del vicedirettore sorge quando un attore sfrutta la fiducia del responsabile del AWS servizio per accedere a risorse a cui non è destinato ad avere accesso.

Cross-account confusa prevenzione sostitutiva

Il seguente diagramma illustra il problema "confused deputy" tra account.



In questo scenario sono validi i requisiti riportati di seguito:

- AWS 1 è tuo Account AWS.

- AWS 1: ExampleRole è un ruolo nel tuo account. La policy di affidabilità di questo ruolo considera attendibile Example Corp specificando l'account AWS di Example Corp come account che può assumere il ruolo.

Ecco che cosa succede:

1. Quando inizi a utilizzare il servizio di Example Corp, fornisci l'ARN AWS di 1 ExampleRole: a Example Corp.
2. Example Corp utilizza quel ruolo ARN per ottenere credenziali di sicurezza temporanee per accedere alle risorse del tuo. Account AWS In questo modo, l'Utente A considera Example Corp come "deputy" attendibile che può agire per conto dell'Utente A stesso.
3. Anche un altro AWS cliente inizia a utilizzare il servizio di Example Corp e fornisce anche l'ARN AWS di 1 ExampleRole: for Example Corp da utilizzare. Presumibilmente l'altro cliente ha imparato o indovinato il numero AWS 1: ExampleRole, che non è un segreto.
4. Quando l'altro cliente chiede a Example Corp di accedere alle AWS risorse del suo account (quello che afferma di essere), Example Corp utilizza AWS 1: ExampleRole per accedere alle risorse del tuo account.

Questo è il modo in cui altri clienti possono ottenere l'accesso non autorizzato alle risorse di un utente, in questo caso dell'Utente A. Poiché il cliente Utente B è stato in grado di ingannare Example Corp e lo ha indotto ad agire involontariamente sulle risorse, Example Corp è ora un "confused deputy".

Example Corp può risolvere il problema "confused deputy" chiedendo di includere la condizione di verifica `ExternalId` nella policy di affidabilità del ruolo. Example Corp genera un valore `ExternalId` univoco per ogni cliente e lo utilizza nella sua richiesta per assumere il ruolo. Il valore `ExternalId` deve essere univoco tra i clienti di Example Corp e controllato da Example Corp, non dai suoi clienti. Questo è il motivo per cui i clienti lo ricevono da Example Corp e non lo creano in autonomia. In questo modo si evita che Example Corp si comporti in modo confuso e consenta l'accesso alle risorse di un altro account. AWS

In questo scenario, immagina che l'ID univoco di Example Corp per te sia 12345 e quello per l'altro cliente sia 67890. Questi ID sono semplificati per comodità in questo scenario. In genere, questi identificatori sono GUID. Supponendo che questi identificatori siano univoci tra i clienti di Example Corp, sono valori sensibili da utilizzare per l'ID esterno.

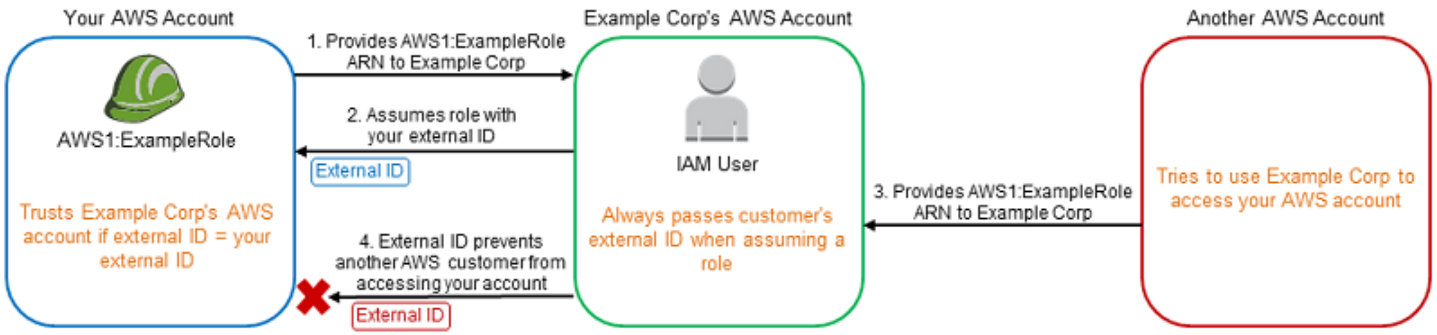
Example Corp ti fornisce il valore ID esterno 12345. È necessario aggiungere un elemento `Condition` alla policy di attendibilità del ruolo che richieda che il valore `sts:ExternalId` sia 12345, come segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {
      "AWS": "Example Corp's AWS Account ID"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "sts:ExternalId": "12345"
      }
    }
  }
}
```

L'elemento Condition di questa politica consente a Example Corp di assumere il ruolo solo quando la chiamata AssumeRole API include il valore ID esterno 12345. Example Corp si assicura che ogni volta che assume un ruolo per conto di un cliente, includa sempre il valore dell'ID esterno del cliente nella chiamata. AssumeRole Anche se un altro cliente fornisce a Example Corp il tuo ARN, non può controllare l'ID esterno che Example Corp include nella sua richiesta. AWS In questo modo è possibile evitare che un cliente non autorizzato acceda alle tue risorse.

Il diagramma seguente illustra tale processo.

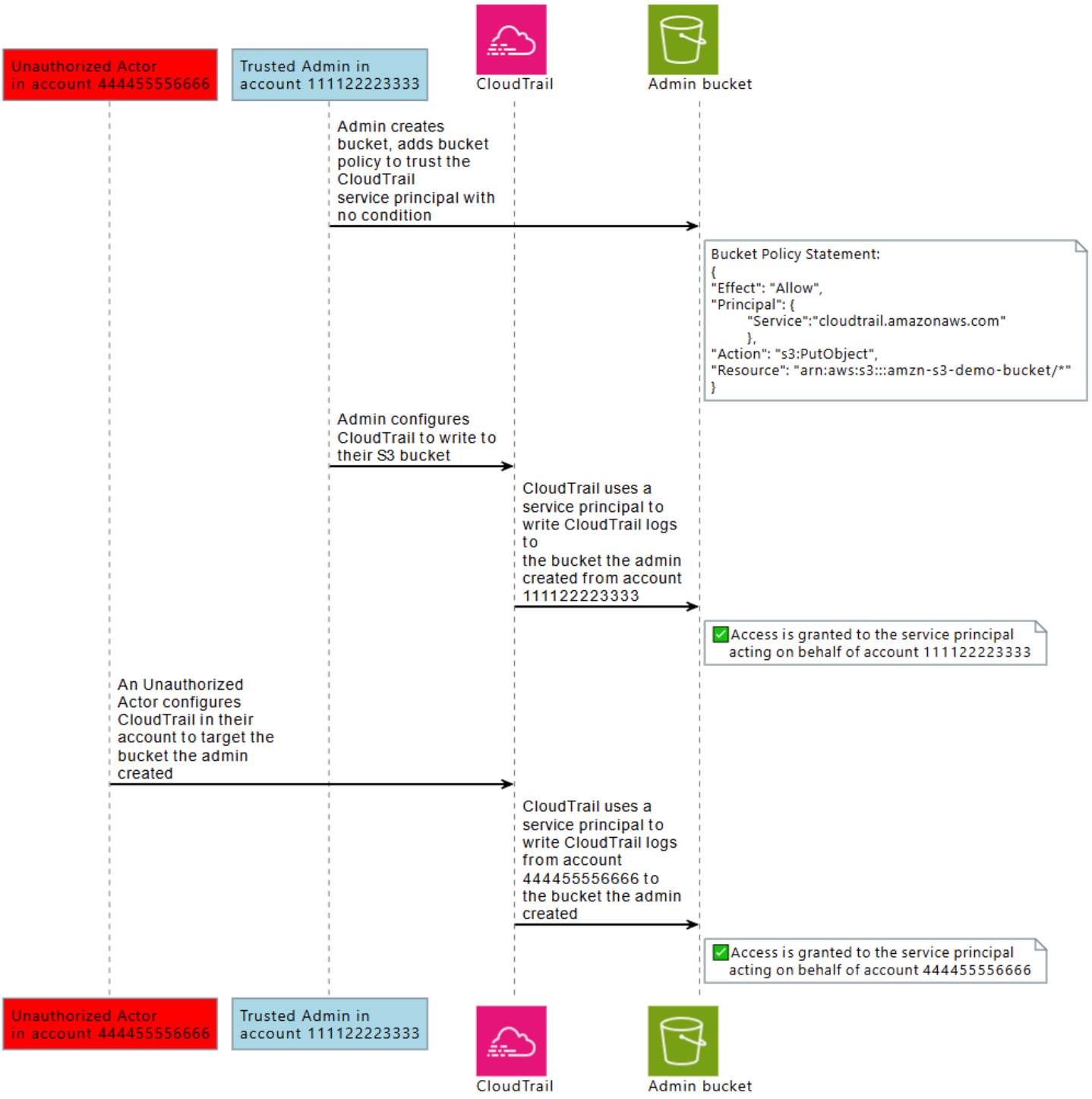


1. Come in precedenza, quando si inizia a utilizzare il servizio di Example Corp, si fornisce l'ARN AWS di 1 ExampleRole: a Example Corp.
2. Quando Example Corp utilizza quel ruolo ARN per assumere il AWS ruolo 1 ExampleRole:, Example Corp include l'ID esterno (12345) nella chiamata API. AssumeRole L'ID esterno corrisponde alla politica di fiducia del ruolo, quindi la chiamata AssumeRole API ha esito positivo e Example Corp ottiene le credenziali di sicurezza temporanee per accedere alle risorse del tuo. Account AWS
3. Anche un altro AWS cliente inizia a utilizzare il servizio di Example Corp e, come in precedenza, fornisce anche l'ARN AWS di 1 ExampleRole: for Example Corp da utilizzare.
4. Ma questa volta, quando Example Corp tenta di assumere il ruolo AWS 1: ExampleRole, fornisce l'ID esterno associato all'altro cliente (67890). L'altro cliente non ha modo di modificare questa operazione. Example Corp opera in questo modo perché la richiesta di utilizzare il ruolo proviene dall'altro cliente, pertanto 67890 indica la circostanza in cui Example Corp sta operando. Poiché hai aggiunto una condizione con il tuo ID esterno (12345) alla politica di fiducia AWS 1: ExampleRole, la AssumeRole chiamata API ha esito negativo. All'altro cliente viene impedito di ottenere l'accesso non autorizzato alle risorse nel tuo account (indicato dalla "X" rossa nel diagramma).

L'ID esterno consente di impedire a qualsiasi altro cliente di ingannare Example Corp e indurre l'azienda ad accedere involontariamente alle risorse.

Cross-service confusa prevenzione sostitutiva

Il diagramma seguente illustra il problema dell'assistente confuso tra servizi utilizzando l' CloudTrail esempio di interazione con Amazon S3, in cui un attore non autorizzato scrive i log su un bucket Amazon S3 CloudTrail a cui non è autorizzato ad accedere.



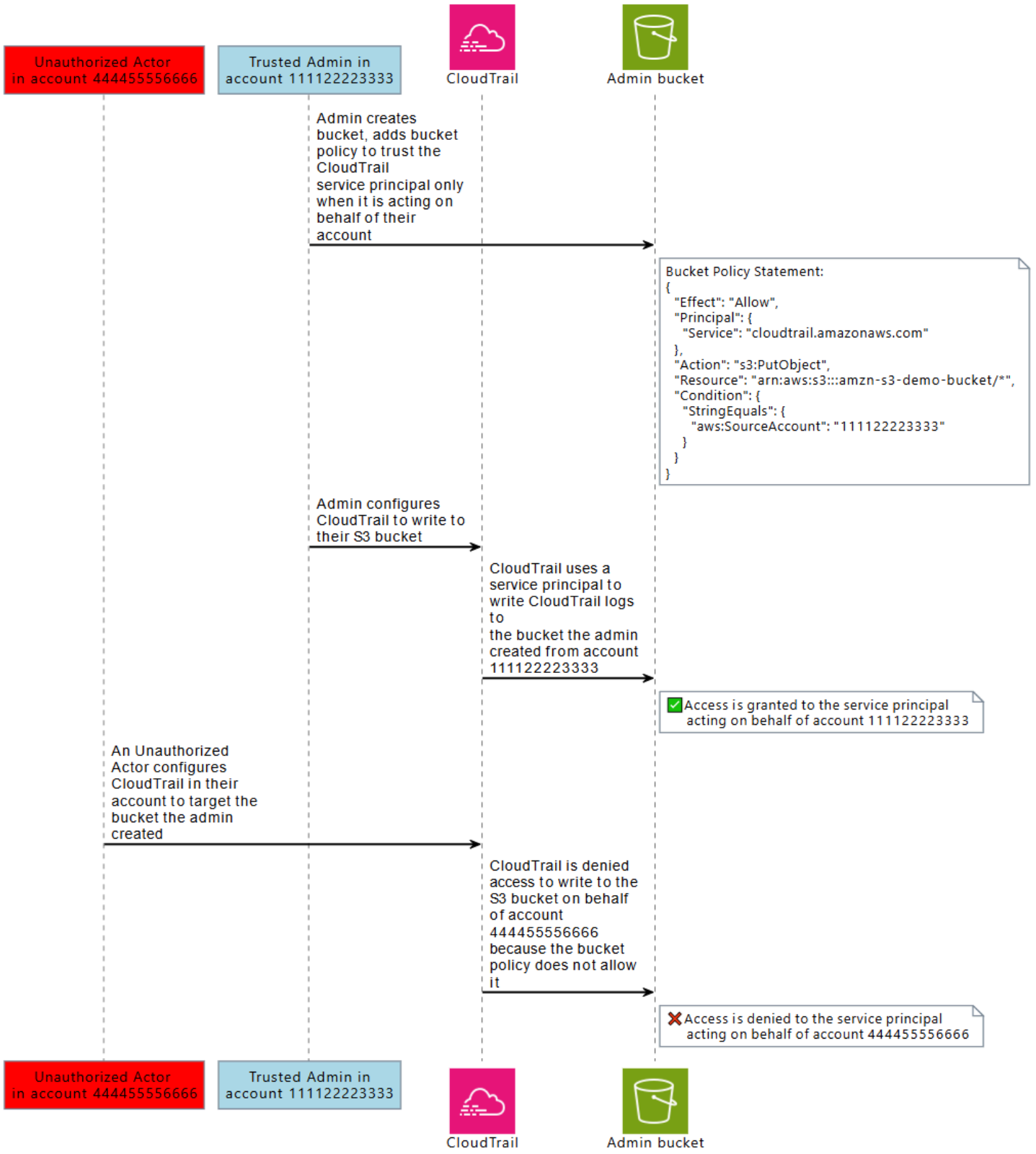
Per evitare che un attore non autorizzato utilizzi la fiducia di un AWS responsabile per accedere alle tue risorse, i responsabili del AWS servizio includono informazioni sulla AWS risorsa e sull' AWS organizzazione per cui agiscono. Account AWS

Queste informazioni sono disponibili in valori chiave globali che possono essere utilizzati in una politica delle risorse o in una politica di controllo delle risorse per le richieste effettuate dai

responsabili del AWS servizio. Ti consigliamo di utilizzareleggi: SourceArn, era: SourceAccountaws: ID SourceOrg, o leggi: SourceOrgPaths nelle politiche relative alle risorse laddove al responsabile del AWS servizio sia concessa l'autorizzazione ad accedere a una delle tue risorse. Queste chiavi di condizione consentono di verificare, nell'ambito delle politiche relative alle risorse o alle politiche di controllo delle risorse, che i responsabili dei AWS servizi che accedono alle risorse lo facciano per conto delle AWS risorse Account AWS, o come previsto dall' AWS Organizations utente.

- `aws:SourceArn`Da utilizzare per consentire a un responsabile del AWS servizio di accedere alle risorse per conto di una risorsa specifica, ad esempio un AWS CloudTrail percorso o una AppStream flotta specifici.
- `aws:SourceAccount`Da utilizzare per consentire a un responsabile del AWS servizio di accedere alle risorse per conto di una determinata persona Account AWS.
- `aws:SourceOrgID`Da utilizzare per consentire a un responsabile AWS del servizio di accedere alle risorse dell'utente per conto di una persona specifica AWS Organizations.
- `aws:SourceOrgPaths`Da utilizzare per consentire al responsabile del AWS servizio di accedere alle risorse per conto di un AWS Organizations percorso specifico.

Il diagramma seguente illustra lo scenario sostitutivo confuso tra diversi servizi in cui una risorsa viene configurata con la chiave di contesto della condizione `aws:SourceAccount` globale e un attore non autorizzato di un altro account tenta di accedere a AWS risorse a cui non dovrebbe avere accesso.



L'utilizzo delle chiavi di condizione globali `aws:SourceArn`, `aws:SourceAccount`, `aws:SourceOrgID` e `aws:SourceOrgPaths` in una policy ti aiuta a garantire che i principali

del servizio accedano alle tue risorse per tuo conto. Ti consigliamo di utilizzare queste chiavi di condizione ogni volta che l'accesso a una delle tue risorse viene concesso a un AWS responsabile del servizio.

> **Note**
>
> Alcune Servizio AWS interazioni prevedono controlli aggiuntivi che aiutano a proteggersi da problemi amministrativi confusi tra diversi servizi che mettono alla prova l'accesso degli utenti a una risorsa. Ad esempio, quando la concessione di una chiave KMS viene concessa a un utente Servizio AWS, AWS KMS utilizza il contesto di crittografia associato alla risorsa e la concessione della chiave per contribuire alla protezione da problemi connessi alla confusione tra i vari servizi.
> Consulta la documentazione dei servizi in uso per ulteriori informazioni sui meccanismi specifici del servizio che possono aiutare a evitare rischi derivanti dal problema "confused deputy" tra servizi diversi e per scoprire se `aws:SourceArn`, `aws:SourceAccount`, `aws:SourceOrgID` e `aws:SourceOrgPaths` sono supportati.

Cross-service ha confuso la protezione dei deputati con le politiche basate sulle risorse

La seguente politica di esempio concede l'`cloudtrail.amazonaws.com`accesso principale del servizio al bucket Amazon S3, arn:aws:s3: ::amzn-s3-demo-bucket1, solo quando il responsabile del servizio agisce per conto di 111122223333. Account AWS

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CloudTrailAclCheck",
            "Effect": "Allow",
            "Principal": {"Service": "cloudtrail.amazonaws.com"},
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
```

```
            }
        },
        {
            "Sid": "AWSCloudTrailWrite",
            "Effect": "Allow",
            "Principal": {"Service": "cloudtrail.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/[optionalPrefix]/Logs/
myAccountID/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Questa policy bucket di esempio concede al servizio l'`appstream.amazonaws.com`accesso principale allo script powershell examplefile.psh all'interno di s3://amzn-s3-demo-bucket2 solo quando agisce per conto della AppStream flotta Amazon specificata, specificando la flotta con cui arn. `aws:SourceArn`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "appstream.amazonaws.com"
                ]
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket2/examplefile.psh",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:appstream:us-
east-1:111122223333:fleet/ExampleFleetName"
```

```
                }
            }
        }
    ]
}
```

Cross-service ha confuso la protezione dei deputati con le politiche di controllo delle risorse

È possibile utilizzare le policy di controllo delle risorse (RCP) per applicare controlli "confused deputy" tra diversi servizi alle risorse dei Servizi AWS supportati. Le RCP consentono di applicare centralmente alle risorse controlli "confused deputy" tra servizi diversi. È possibile utilizzare chiavi di condizione analoghe `aws:SourceOrgId` e `aws:SourceOrgPaths` con gli RCP allegati alle AWS Organizations unità organizzative (OU) o Account AWS all'interno dell'organizzazione senza aggiungere dichiarazioni a politiche specifiche basate sulle risorse. Per ulteriori informazioni sulle RCP e sui servizi supportati, consulta Resource control policies (RCPs) nella Guida per l'utente di AWS Organizations .

L'esempio seguente RCP nega ai responsabili AWS del servizio l'accesso ai bucket Amazon S3 nei tuoi account membro quando non `aws:SourceOrgID` è uguale a o-. ExampleOrg Un'autorizzazione corrispondente deve essere presente nella policy basata sulle risorse del bucket S3 per consentire i principali con un valore pari a o-. Servizio AWS `SourceOrgID` ExampleOrg

Questa policy applica il controllo solo alle richieste dei principali del servizio (`"Bool": {"aws:PrincipalIsAWSService": "true"}`) che hanno la chiave `aws:SourceAccount` (`"Null": {"aws:SourceAccount": "false"}`), in modo che le integrazioni di servizi che non richiedono l'uso di questa chiave di condizione e le chiamate da parte dei principali non vengano influenzate. Se la chiave di condizione `aws:SourceAccount` è presente nel contesto della richiesta, la condizione Null verrà valutata come true, determinando l'applicazione della chiave `aws:SourceOrgID`. Utilizziamo invece `aws:SourceAccount` al posto di `aws:SourceOrgID` nell'operatore di condizione Null in modo che il controllo si applichi ancora se la richiesta proviene da un account che non appartiene a un'organizzazione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Sid": "RCPEnforceConfusedDeputyProtectionForS3",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceOrgID": "o-ExampleOrg"
        },
        "Null": {
          "aws:SourceAccount": "false"
        },
        "Bool": {
          "aws:PrincipalIsAWSService": "true"
        }
      }
    }
  ]
}
```

Scenari comuni per i ruoli IAM

Come per la maggior parte delle AWS funzionalità, in genere hai due modi per utilizzare un ruolo: in modo interattivo nella console IAM o a livello di codice con gli AWS CLI strumenti per Windows PowerShell o l'API.

- Gli utenti IAM nell'account che utilizza la console IAM possono passare a un ruolo per utilizzare temporaneamente le autorizzazioni del ruolo nella console. Gli utenti abbandonano le loro autorizzazioni originali e assumono le autorizzazioni assegnate al ruolo. Quando gli utenti escono dal ruolo, le autorizzazioni originali vengono ripristinate.
- Un'applicazione o un servizio offerto da AWS (come Amazon EC2) può assumere un ruolo richiedendo credenziali di sicurezza temporanee per un ruolo a cui effettuare richieste programmatiche. AWSÈ possibile utilizzare un ruolo in questo modo per non dover condividere o gestire le credenziali di sicurezza a lungo termine (ad esempio creando un utente IAM) per ogni entità che richiede l'accesso a una risorsa.

> **Note**
>
> In questa guida le frasi passare a un ruolo e assumere un ruolo vengono utilizzate in modo intercambiabile.

Il modo più semplice per utilizzare i ruoli è quello di concedere agli utenti IAM le autorizzazioni per passare ai ruoli creati da te all'interno del tuo o di un altro Account AWS. È possibile passare da un ruolo all'altro facilmente utilizzando la console IAM per utilizzare le autorizzazioni che non si desidera abbiano normalmente e uscire dal ruolo per cedere a tali autorizzazioni. Ciò può aiutare a impedire l'accesso accidentale alle risorse sensibili o la loro modifica.

Per utilizzi più complessi di ruoli, ad esempio la concessione di accesso alle applicazioni e servizi, o gli utenti federati esterni, è possibile richiamare l'API `AssumeRole`. Questa chiamata API restituisce un set di credenziali temporanee che l'applicazione può utilizzare in successive chiamate API. Le operazioni tentate con le credenziali temporanee dispongono solo delle autorizzazioni concesse dal ruolo associato. Un'applicazione non deve "uscire" dal ruolo nello stesso modo di un utente nella console, ma l'applicazione smette semplicemente di utilizzare le credenziali temporanee e riprende le chiamate con le credenziali originali.

Gli utenti federati accedono utilizzando le credenziali di un provider di identità (IdP). AWS fornisce quindi credenziali temporanee all'IdP affidabile da trasmettere all'utente per includerle nelle AWS successive richieste di risorse. Queste credenziali forniscono le autorizzazioni concesse al ruolo assegnato.

Questa sezione fornisce una panoramica dei seguenti scenari:

- Fornisci l'accesso a un utente IAM in uno Account AWS di tua proprietà per accedere alle risorse di un altro account di tua proprietà
- Fornire l'accesso a carichi di lavoro non AWS
- Fornire l'accesso agli utenti IAM negli Account AWS di proprietà di terze parti
- Fornisci l'accesso ai servizi offerti dalle AWSAWS risorse
- Fornire l'accesso agli utenti autenticati esternamente (federazione delle identità)

Accesso per un utente IAM in un altro Account AWS che possiedi

Puoi concedere ai tuoi utenti IAM il permesso di passare a ruoli all'interno del tuo Account AWS o a ruoli definiti in altri Account AWS ruoli di tua proprietà.

> **Note**
>
> Se desideri concedere l'accesso a un account che non possiedi né controlli, consulta Accesso a Account AWS di proprietà di terzi più avanti in questo argomento.

Immaginiamo di avere delle istanze Amazon EC2 critiche per la tua organizzazione. Invece di concedere direttamente agli utenti l'autorizzazione a terminare le istanze, è possibile creare un ruolo con tali privilegi. Quindi consentire agli amministratori di passare al ruolo quando è necessario terminare un'istanza. In questo modo si aggiungono i seguenti livelli di protezione alle istanze:

- È necessario concedere esplicitamente agli utenti il permesso di assumere quel ruolo.
- I tuoi utenti devono passare attivamente al ruolo utilizzando Console di gestione AWS o assumere il ruolo utilizzando l' AWS API AWS CLI or.
- È possibile aggiungere una Multi-Factor Authentication (MFA) al ruolo, in modo che solo gli utenti che accedono con un dispositivo MFA possano assumere quel ruolo. Per ulteriori informazioni su come configurare un ruolo in modo che gli utenti che assumono il ruolo debbano essere prima autenticati utilizzando l'autenticazione a più fattori (MFA), consulta Accesso sicuro alle API con MFA.

Consigliamo di utilizzare questo approccio per applicare il principio di privilegio minimo. Ciò significa limitare l'uso di autorizzazioni elevate unicamente a quelle volte in cui sono necessarie per operazioni specifiche. Con i ruoli puoi prevenire modifiche accidentali agli ambienti sensibili, soprattutto se li combini con il controllo per assicurarti che i ruoli vengano utilizzati solo quando necessario.

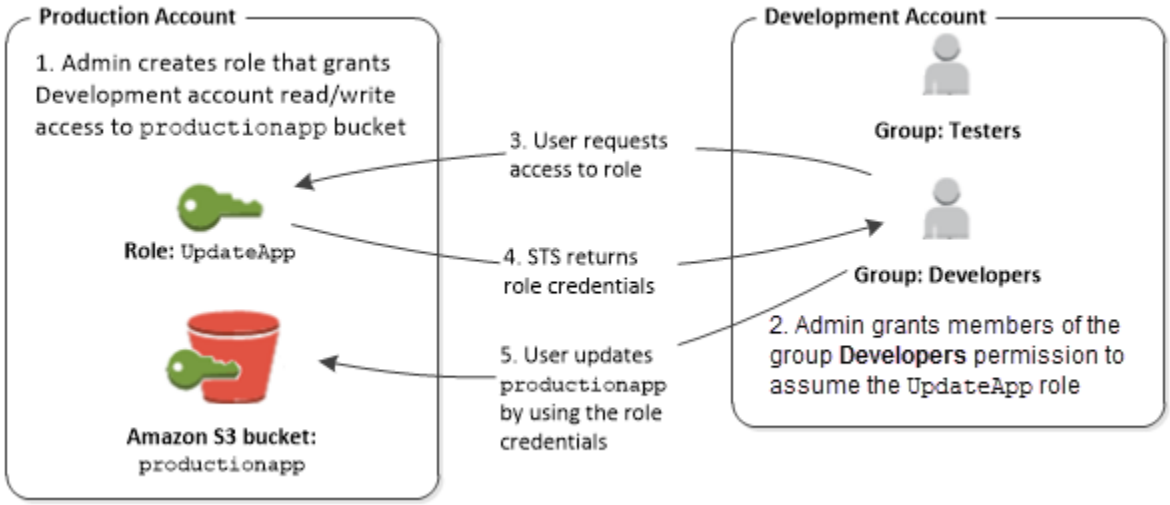
Quando si crea un ruolo per questo scopo, è necessario specificare l'ID degli account da cui gli utenti devono accedere nell'elemento `Principal` della policy di affidabilità del ruolo. È quindi possibile concedere agli utenti specifici in tali altri account le autorizzazioni per passare al ruolo. Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Un utente in un account può passare a un ruolo dello stesso o di un altro account. Mentre si usa il ruolo, l'utente è in grado di eseguire solo le azioni e accedere solo alle risorse consentite dal ruolo;

le loro autorizzazioni utente originali sono sospese. Quando l'utente esce dal ruolo, le autorizzazioni utente originali vengono ripristinate.

Esempio di uno scenario in cui si utilizzano account di sviluppo e produzione separati

Immagina che la tua organizzazione ne abbia diversi Account AWS per isolare un ambiente di sviluppo da un ambiente di produzione. Gli utenti nell'account di sviluppo potrebbero occasionalmente aver bisogno di accedere alle risorse nell'account di produzione. Ad esempio, potrebbe essere necessario l'accesso a più account quando si sta richiedendo un aggiornamento dall'ambiente di sviluppo all'ambiente di produzione. Anche se è possibile creare identità separate (e password) per gli utenti che lavorano con entrambi gli account, la gestione delle credenziali per più account complica la gestione delle identità. Nell'illustrazione seguente, tutti gli utenti vengono gestiti nell'account di sviluppo, ma per alcuni sviluppatori è necessario un accesso limitato all'account di produzione. L'account di sviluppo dispone di due gruppi: collaudatori e sviluppatori, e ciascun gruppo ha la propria policy.



1. Nell'account di produzione, un amministratore utilizza IAM per creare il ruolo `UpdateApp` in tale account. Nel ruolo, l'amministratore definisce una policy di affidabilità che specifica l'account di sviluppo come `Principal`; in tal modo gli utenti autorizzati dall'account di sviluppo possono utilizzare il ruolo `UpdateApp`. L'amministratore definisce inoltre una policy delle autorizzazioni per il ruolo che specifica le autorizzazioni in lettura e scrittura per il bucket Amazon S3 denominato `productionapp`.

   L'amministratore quindi condivide le informazioni appropriate con chiunque debba assumere quel ruolo. Tali informazioni sono il numero di account e il nome del ruolo (per gli utenti della AWS console) o l'Amazon Resource Name (ARN) (per AWS CLI l'accesso all' AWS API). L'ARN del

ruolo può essere simile a `arn:aws:iam::123456789012:role/UpdateApp`, dove il ruolo è denominato `UpdateApp` ed è stato creato nel numero di account 123456789012.

> **Note**
>
> L'amministratore può eventualmente configurare il ruolo in modo che gli utenti che assumono il ruolo debbano essere prima autenticati utilizzando l'autenticazione a più fattori (MFA). Per ulteriori informazioni, consulta Accesso sicuro alle API con MFA.

2. Nell'account di sviluppo, un amministratore concede ai membri del gruppo di sviluppatori l'autorizzazione a cambiare il ruolo. Ciò avviene concedendo al gruppo di sviluppatori l'autorizzazione a chiamare l'`AssumeRole`API Servizio di token di sicurezza AWS (AWS STS) per il `UpdateApp` ruolo. Qualsiasi utente IAM che appartiene al gruppo Sviluppatori nell'account di sviluppo può ora passare al ruolo `UpdateApp` nell'account di produzione. Gli altri utenti che non appartengono al gruppo di sviluppatori non hanno il permesso di passare al ruolo e pertanto non sono in grado di accedere al bucket S3 nell'account di produzione.
3. L'utente richiede di passare al ruolo:
   - AWS console: l'utente sceglie il nome dell'account nella barra di navigazione e sceglie Cambia ruolo. L'utente specifica l'ID account (o alias) e il nome del ruolo. In alternativa, l'utente può scegliere un link inviato via e-mail dall'amministratore. Il link indirizza l'utente alla pagina Switch Role (Cambia ruolo) con i dettagli già compilati.
   - AWS API/AWS CLI: un utente del gruppo Developers dell'account di sviluppo chiama la `AssumeRole` funzione per ottenere le credenziali per il `UpdateApp` ruolo. L'utente specifica l'ARN del ruolo `UpdateApp` come parte della chiamata. Se un utente nel gruppo di collaudatori inoltra la stessa richiesta, la richiesta ha esito negativo perché i collaudatori non hanno l'autorizzazione a chiamare `AssumeRole` per il `UpdateApp` ruolo ARN.
4. AWS STS restituisce credenziali temporanee:
   - AWS console: AWS STS verifica la richiesta con la politica di fiducia del ruolo per assicurarsi che la richiesta provenga da un'entità attendibile (che è: l'account di sviluppo). Dopo la verifica, AWS STS restituisce le credenziali di sicurezza temporanee alla AWS console.
   - API/CLI: AWS STS verifica la richiesta rispetto alla politica di fiducia del ruolo per assicurarsi che provenga da un'entità attendibile (che è: l'account di sviluppo). Dopo la verifica, AWS STS restituisce le credenziali di sicurezza temporanee all'applicazione.
5. Le credenziali temporanee consentono l'accesso alla AWS risorsa:

- AWS console: la AWS console utilizza le credenziali temporanee per conto dell'utente per tutte le successive azioni della console, in questo caso, di lettura e scrittura nel `productionapp` bucket. La console non è in grado di accedere ad altre risorse nell'account di produzione. Quando l'utente esce dal ruolo, le autorizzazioni dell'utente tornano a quelle originali detenute prima di cambiare il ruolo.
- API/CLI: l'applicazione utilizza le credenziali di sicurezza temporanee per aggiornare il bucket. `productionapp` Con le credenziali di sicurezza provvisorie, l'applicazione può solo leggere e scrivere al bucket `productionapp` e non è in grado di accedere a qualsiasi altra risorsa nell'account di produzione. L'applicazione non deve uscire dal ruolo, bensì cessa di utilizzare le credenziali provvisorie e utilizza le credenziali originali nelle successive chiamate API.

Risorse aggiuntive

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Tutorial IAM: Delega l'accesso attraverso AWS account che utilizzano ruoli IAM

Accesso per carichi non AWS di lavoro

Un ruolo IAM è un oggetto in AWS Identity and Access Management (IAM) a cui vengono assegnate autorizzazioni. Quando assumi quel ruolo utilizzando un'identità IAM o un'identità esterna a AWS, ti vengono fornite credenziali di sicurezza temporanee per la tua sessione di ruolo. Potresti avere carichi di lavoro in esecuzione nel tuo data center o in un'altra infrastruttura esterna AWS che deve accedere alle tue AWS risorse. Invece di creare, distribuire e gestire chiavi di accesso a lungo termine, puoi utilizzare AWS Identity and Access Management Roles Anywhere (IAM Roles Anywhere) per autenticare i tuoi carichi non di lavoro.AWS IAM Roles Anywhere utilizza i certificati X.509 dell'autorità di certificazione (CA) per autenticare le identità e fornire l'accesso in modo sicuro alle credenziali temporanee fornite da Servizi AWS un ruolo IAM.

Per utilizzare IAM Roles Anywhere

1. Configura una CA utilizzando AWS Autorità di certificazione privata o utilizza una CA dalla propria infrastruttura PKI.
2. Dopo aver impostato una CA, viene creato un oggetto in IAM Roles Anywhere chiamato ancoraggio di fiducia. Questo ancoraggio stabilisce la fiducia tra IAM Roles Anywhere e la tua CA per l'autenticazione.

3. Puoi quindi configurare i ruoli IAM esistenti o creare nuovi ruoli che si fidino del servizio IAM Roles Anywhere.
4. Autentica i tuoi AWS carichi non di lavoro con IAM Roles Anywhere utilizzando il trust anchor. AWS concede le credenziali temporanee non relative ai AWS carichi di lavoro al ruolo IAM che ha accesso alle tue risorse. AWS

Risorse aggiuntive

Le risorse seguenti possono rivelarsi utili per fornire l'accesso a carichi di lavoro non AWS .

- Per ulteriori informazioni sulla configurazione di Ruoli IAM Anywhere, consulta l'argomento Cos'è AWS Identity and Access Management Ruoli Anywhere nella Guida dell'utente di IAM Roles Anywhere.
- Per scoprire come configurare un'infrastruttura a chiave pubblica (PKI) per IAM Roles Anywhere, consulta IAM Roles Anywhere con un'autorità di certificazione esterna nel Blog sulla sicurezza AWS .

Accesso a Account AWS di proprietà di terzi

Quando terze parti richiedono l'accesso alle AWS risorse della tua organizzazione, puoi utilizzare i ruoli per delegare l'accesso ad esse. Ad esempio, una terza parte potrebbe fornire un servizio per la gestione delle risorse AWS . Con i ruoli IAM, puoi concedere a queste terze parti l'accesso alle tue AWS risorse senza condividere le tue credenziali AWS di sicurezza. Invece, la terza parte può accedere alle tue AWS risorse assumendo un ruolo che hai creato nel tuo. Account AWS Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Le terze parti devono fornirti le informazioni seguenti per permetterti di creare un ruolo che possa essere da loro assunto:

- L' Account AWS ID della terza parte. Puoi specificare il loro ID dell' Account AWS come entità principale quando definisci la policy di affidabilità per il ruolo.
- Un ID esterno da associare in modo univoco con il ruolo. L'ID esterno può essere qualsiasi identificatore noto a te e alla terza parte. Puoi ad esempio usare un ID di fattura tra te e la terza parte, ma non devi usare qualcosa che sia possibile indovinare, ad esempio il nome o il numero di telefono della terza parte. Devi specificare questo ID quando definisci la policy di affidabilità per il ruolo. La terza parte deve fornire questo ID quando assume il ruolo.

- Le autorizzazioni di cui la terza parte necessita per usare le risorse AWS Devi specificare queste autorizzazioni quando definisci la policy di autorizzazione del ruolo. Questa policy definisce le operazioni consentite e le risorse a cui è possibile accedere.

Dopo aver creato il ruolo, devi fornire l'Amazon Resource Name (ARN) del ruolo alla terza parte. L'ARN del ruolo è necessario per assumere il ruolo.

> **Important**
>
> Quando concedi a terze parti l'accesso alle tue AWS risorse, queste possono accedere a qualsiasi risorsa specificata nella politica. Le risorse usate dalla terza parte vengono fatturate a te. Assicurati di limitare l'uso delle risorse in modo appropriato.

ID esterni per l'accesso di terze parti

Un ID esterno consente all'utente che sta assumendo il ruolo di dichiarare le circostanze in cui sta operando. Fornisce inoltre un modo per il proprietario dell'account di consentire che il ruolo venga assunto solo in circostanze specifiche. La funzione principale dell'ID esterno è quella di risolvere e prevenire il Problema del "confused deputy".

> **Important**
>
> AWS non considera l'ID esterno come un segreto. Dopo aver creato un segreto, ad esempio una coppia di chiavi di accesso o una password AWS, non è possibile visualizzarlo nuovamente. L'ID esterno di un ruolo può essere visto da chiunque disponga dell'autorizzazione a visualizzare il ruolo.

Quando si deve usare l'ID esterno?

Utilizzare un ID esterno nelle seguenti situazioni:

- Sei un Account AWS proprietario e hai configurato un ruolo per una terza parte che accede ad un'altra persona oltre Account AWS al tuo. È opportuno chiedere alla terza parte un ID esterno da includere quando assume il ruolo fornito alla terza parte. Quindi verificare l'ID esterno tramite la policy di affidabilità del ruolo fornito alla terza parte. Ciò garantisce che la parte esterna possa assumere il tuo ruolo solo quando agisce per conto del proprietario.

- Ci si trova in una posizione che comporta l'assunzione di ruoli per conto di diversi clienti in modo analogo a Example Corp nello scenario precedente. È opportuno assegnare un ID esterno univoco a ciascun cliente e fornire indicazioni per aggiungere l'ID esterno alla policy di affidabilità creata per il ruolo da fornire. È quindi necessario assicurarsi di includere sempre l'ID esterno corretto nelle richieste di assunzione dei ruoli.

  Probabilmente si dispone già di un identificativo univoco per ogni cliente e questo ID univoco è sufficiente per l'utilizzo come ID esterno. L'ID esterno non è un valore speciale da creare in modo esplicito o monitorare separatamente, solo per questo scopo.

  Si deve sempre specificare l'ID esterno nelle chiamate API `AssumeRole`. Inoltre, quando un cliente assegna un ARN del ruolo, verificare se è possibile assumere il ruolo con e senza l'ID esterno corretto. Se è possibile assumere il ruolo senza l'ID esterno corretto, non memorizzare l'ARN del ruolo del cliente nel sistema. Attendere fino a quando il cliente non ha aggiornato la policy di affidabilità del ruolo per richiedere l'ID esterno corretto. In questo modo è possibile aiutare i clienti a operare nel modo corretto e pertanto a garantire la sicurezza di entrambi rispetto al problema "confused deputy".

Scenario di esempio che utilizza un ID esterno

Ad esempio, supponiamo che tu decida di assumere una società terza chiamata Example Corp per monitorare Account AWS e ottimizzare i costi. Per tenere traccia delle spese giornaliere, Example Corp deve accedere alle tue AWS risorse. Example Corp controlla anche molti altri account AWS per altri clienti.

Non fornire l'accesso a Example Corp a un utente IAM e le relative credenziali a lungo termine nell'account AWS . Utilizza invece un ruolo IAM e le credenziali di sicurezza temporanee. Un ruolo IAM fornisce un meccanismo per consentire a una terza parte di accedere alle tue AWS risorse senza dover condividere credenziali a lungo termine (come una chiave di accesso utente IAM).

Puoi utilizzare un ruolo IAM per stabilire una relazione di attendibilità tra il tuo Account AWS e l'account di Example Corp. Una volta stabilita questa relazione, un membro dell'account Example Corp può chiamare l' Servizio di token di sicurezza AWS AssumeRoleAPI per ottenere credenziali di sicurezza temporanee. I membri di Example Corp possono quindi utilizzare le credenziali per accedere alle AWS risorse del tuo account.

> **Note**
>
> Per ulteriori informazioni su questa AssumeRole e altre operazioni AWS API che puoi chiamare per ottenere credenziali di sicurezza temporanee, consulta. Confronta le AWS STS credenziali

Di seguito è illustrata un'analisi più dettagliata di questo scenario.

1. L'Utente A affida un incarico a Example Corp, che crea un identificatore univoco per l'Utente A. Ti forniscono questo ID cliente univoco e il relativo Account AWS numero. Queste informazioni sono necessarie per creare un ruolo IAM nella fase successiva.

   > **Note**
   >
   > Example Corp può utilizzare qualsiasi valore di stringa desiderato per il ExternalId, purché sia univoco per ogni cliente. È possibile che si tratti di un numero di account cliente o addirittura di una stringa di caratteri casuale, purché non esistano due clienti con lo stesso valore. Non si tratta di un "segreto". Example Corp deve fornire il ExternalId valore a ciascun cliente. L'aspetto cruciale è che l'ID deve essere generato da Example Corp e non dai clienti affinché ogni ID esterno sia univoco.

2. Accedi AWS e crei un ruolo IAM che consenta a Example Corp di accedere alle tue risorse. Come per qualsiasi ruolo IAM, il ruolo dispone di due tipi di policy: una policy di autorizzazione e una policy di attendibilità. La policy di affidabilità del ruolo specifica chi può assumere il ruolo. Nel nostro scenario di esempio, la policy specifica il Account AWS numero di Example Corp come. `Principal` Ciò consente alle identità di tale account di assumere il ruolo. Inoltre, viene aggiunto un elemento `Condition` alla policy di attendibilità. Questo elemento `Condition` verifica la chiave di contesto `ExternalId` per assicurarsi che corrisponda all'ID cliente univoco di Example Corp. Ad esempio:

   ```
      "Principal": {"AWS": "Example Corp's Account AWS ID"},
      "Condition": {"StringEquals": {"sts:ExternalId": "Unique ID Assigned by Example
   Corp"}}
   ```

3. La policy di autorizzazione per il ruolo specifica le operazioni che il ruolo consente di effettuare a un utente. Ad esempio, puoi specificare che il ruolo deve permettere agli utenti di gestire solo le risorse Amazon RDS e Amazon EC2, ma non gli utenti o i gruppi IAM. In questo scenario di

esempio, si utilizza la policy di autorizzazione per fornire l'accesso in sola lettura per Example Corp a tutte le risorse nell'account dell'Utente A.

4. Dopo aver creato il ruolo, è necessario fornire l'Amazon Resource Name (ARN) del ruolo a Example Corp.
5. Quando Example Corp ha bisogno di accedere alle tue AWS risorse, qualcuno dell'azienda chiama l'API. AWS `sts:AssumeRole` La chiamata include l'ARN del ruolo da assumere e il ExternalId parametro che corrisponde all'ID del cliente.

Se la richiesta proviene da qualcuno che utilizza Example Corp's Account AWS e se l'ARN del ruolo e l'ID esterno sono corretti, la richiesta ha esito positivo. Fornisce quindi credenziali di sicurezza temporanee che Example Corp può utilizzare per accedere alle AWS risorse consentite dal ruolo dell'utente.

In altre parole, quando una policy di ruolo include un ID esterno, chiunque desideri assumere il ruolo deve essere un entità principale nel ruolo e deve includere l'ID esterno corretto.

Punti chiave per gli ID esterni

- In un ambiente multi-tenant in cui si supportano più clienti con AWS account diversi, si consiglia di utilizzare un ID esterno per. Account AWS Questo ID dovrebbe essere una stringa casuale generata dalla terza parte.
- Per richiedere che la terza parte fornisca un ID esterno quando si assume un ruolo, aggiorna la policy di attendibilità del ruolo con l'ID esterno scelto.
- Per fornire un ID esterno quando assumi un ruolo, utilizza l' AWS API AWS CLI o per assumere quel ruolo. Per ulteriori informazioni, vedete l'operazione AssumeRole API STS o l'operazione CLI assume-role di STS.
- Il valore `ExternalId` deve avere un minimo di 2 caratteri e un massimo di 1.224 caratteri. Il valore deve essere alfanumerico senza spazi. Può anche includere i seguenti simboli: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), due punti (:), barra (/) e trattino (-).

Risorse aggiuntive

Le risorse seguenti possono rivelarsi utili per fornire l'accesso a Account AWS di proprietà di terze parti.

- Per sapere come consentire ad altri di eseguire azioni nel tuo, consulta. Account AWSCreare un ruolo utilizzando policy di attendibilità personalizzate

- Per informazioni su come concedere l'autorizzazione per passare a un ruolo, consulta Concedere le autorizzazioni agli utenti per cambiare ruoli
- Per informazioni su come creare e fornire agli utenti attendibili credenziali di sicurezza temporanee, consultaAutorizzazioni per le credenziali di sicurezza temporanee.

Accesso a un AWS servizio

Molti AWS servizi richiedono l'utilizzo di ruoli per controllare a cosa può accedere il servizio. Un ruolo che un servizio assume per eseguire operazioni a tuo nome viene chiamato ruolo del servizio. Quando un ruolo fornisce uno scopo specializzato per un servizio, questo può essere categorizzato come ruolo collegato al servizio. Consulta la documentazione AWS di ciascun servizio per verificare se utilizza ruoli e per ulteriori informazioni su come assegnare un ruolo per il servizio da utilizzare.

Per informazioni dettagliate sulla creazione di un ruolo per delegare l'accesso a un servizio offerto da AWS, consultaCrea un ruolo per delegare le autorizzazioni a un AWS service.

Accesso a utenti autenticati esternamente (federazione delle identità)

I tuoi utenti potrebbero già avere identità esterne AWS, ad esempio nella tua directory aziendale. Se tali utenti devono utilizzare AWS le risorse (o lavorare con applicazioni che accedono a tali risorse), devono anche creare credenziali AWS di sicurezza. È possibile utilizzare un ruolo IAM per specificare le autorizzazioni per gli utenti la cui identità è federata dalla propria organizzazione o da un provider di identità di terze parti (IdP).

> **Note**
>
> Come best practice di sicurezza, ti consigliamo di gestire l'accesso degli utenti in Centro identità IAM con la federazione delle identità anziché creare utenti IAM. Per informazioni su situazioni specifiche in cui è richiesto un utente IAM, consulta la sezione Quando creare un utente IAM invece di un ruolo.

I set di autorizzazioni IAM Identity Center forniscono modelli di ruolo IAM gestiti da IAM Identity Center. Account access manager , una funzionalità IAM, consente di assegnare i propri ruoli IAM personalizzati agli utenti e ai gruppi di IAM Identity Center, abilitando funzionalità come politiche di fiducia personalizzate, tag di ruolo e percorsi di ruolo configurabili. Puoi usarlo insieme ai set di autorizzazioni o da solo.

Federazione di utenti di una applicazione per dispositivi mobili o basata sul Web con Amazon Cognito

Se crei un'app mobile o basata sul Web che accede alle AWS risorse, l'app necessita delle credenziali di sicurezza per effettuare richieste programmatiche. AWS Per la maggior parte degli scenari relativi alle applicazioni per dispositivi mobili, consigliamo di utilizzare Amazon Cognito. Puoi utilizzare questo servizio con AWS Mobile SDK per iOS e Mobile SDK per Android e Fire OS per creare identità univoche per gli utenti e autenticarli per un accesso sicuro alle tue risorse. AWS AWS Amazon Cognito supporta gli stessi provider di identità come quelli elencati nella sezione successiva e supporta anche identità autenticate dallo sviluppatore e accesso non autenticato (ospite). Amazon Cognito fornisce inoltre operazioni API per la sincronizzazione dei dati utente in modo che vengano conservati quando gli utenti passano da un dispositivo all'altro. Per ulteriori informazioni, consulta Amazon Cognito per applicazioni per dispositivi mobili.

Federazione degli utenti con provider di servizi di identità pubblica o OpenID Connect

Quando possibile, utilizza Amazon Cognito per scenari di applicazioni per dispositivi mobili o basate sul Web. Amazon Cognito lavora principalmente dietro le quinte con servizi di provider di identità pubblica per tuo conto. Lavora con gli stessi servizi di terze parti e supporta anche gli accessi anonimi. Tuttavia, per ulteriori scenari avanzati, è possibile lavorare direttamente con un servizio di terze parti, ad esempio Login with Amazon, Facebook, Google o qualsiasi IdP compatibile con OpenID Connect (OIDC). Per ulteriori informazioni sull'utilizzo della federazione OIDC utilizzando uno di questi servizi, consulta Federazione OIDC.

Federazione degli utenti con SAML 2.0

Se la tua organizzazione utilizza già un pacchetto software per provider di identità che supporta SAML 2.0 (Security Assertion Markup Language 2.0), puoi creare fiducia tra la tua organizzazione come provider di identità (IdP) e come fornitore di servizi. AWS Puoi quindi utilizzare SAML per fornire ai tuoi utenti un single sign-on (SSO) federato o un accesso federato per chiamare le Console di gestione AWS operazioni API. AWS Ad esempio, se la tua azienda utilizza Microsoft Active Directory e Active Directory Federation Services, puoi effettuare la federazione utilizzando SAML 2.0. Per ulteriori informazioni sulla federazione degli utenti con SAML 2.0, consulta Federazione SAML 2.0.

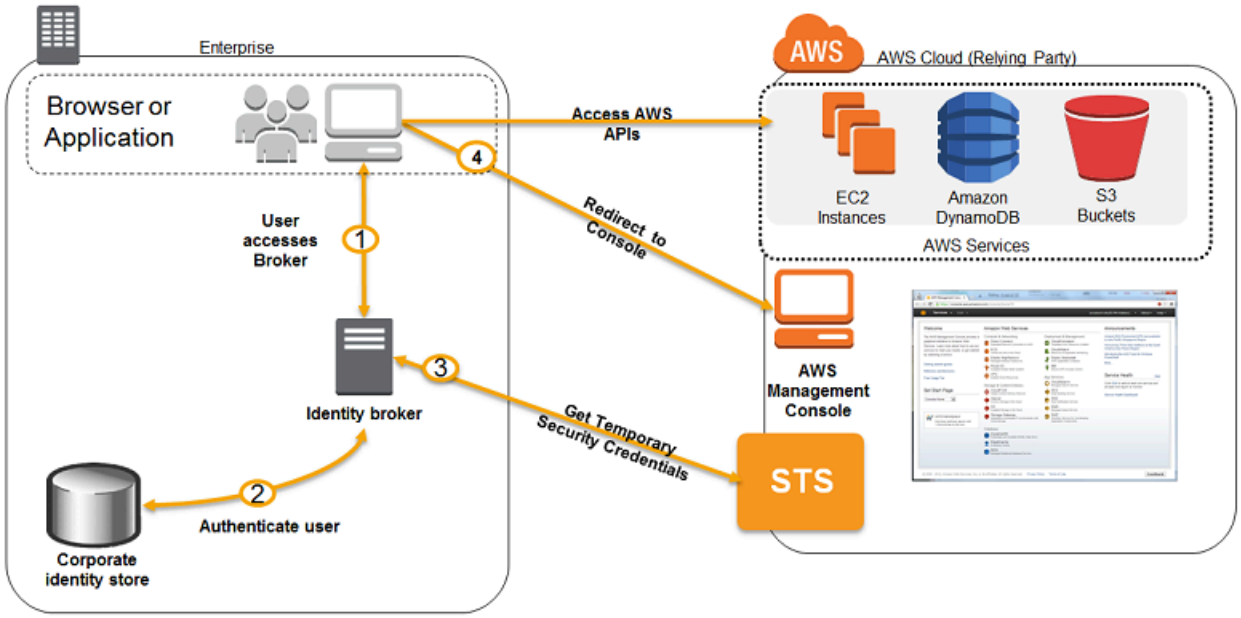
Federazione degli utenti creando un'applicazione personalizzata per la gestione di identità

Se il proprio archivio identità non è compatibile con SAML 2.0, è possibile creare un'applicazione personalizzata per la gestione di identità per eseguire una funzione simile. L'applicazione broker autentica gli utenti, richiede credenziali temporanee per gli utenti e quindi le fornisce all'utente per accedere alle AWS risorse. AWS

Ad esempio, Example Corp. ha molti dipendenti che devono eseguire applicazioni interne che accedono alle risorse dell'azienda. AWS I dipendenti hanno già identità nel sistema di identità e autenticazione dell'azienda ed Example Corp. non desidera creare un utente IAM separato per ogni dipendente dell'azienda.

Bob è uno sviluppatore presso Example Corp. Per consentire alle applicazioni interne di Example Corp. di accedere alle AWS risorse dell'azienda, Bob sviluppa un'applicazione di identity broker personalizzata. L'applicazione verifica che i dipendenti abbiano effettuato l'accesso nel sistema di identità e autenticazione esistente, che potrebbe utilizzare LDAP, Active Directory o un altro sistema. L'applicazione del gestore identità quindi ottiene le credenziali di sicurezza provvisorie per i dipendenti. Questo scenario è simile a quello precedente (un'app mobile che utilizza un sistema di autenticazione personalizzato), tranne per il fatto che le applicazioni che devono accedere alle AWS risorse vengono eseguite tutte all'interno della rete aziendale e l'azienda dispone di un sistema di autenticazione esistente.

Per ottenere le credenziali di sicurezza provvisorie, l'applicazione del gestore identità chiama `AssumeRole` o `GetFederationToken` per ottenere le credenziali di sicurezza provvisorie, a seconda di come Bob desidera gestire le policy per gli utenti e quando scadono le credenziali provvisorie. (Per ulteriori informazioni sulle differenze tra queste operazioni API, consultare Credenziali di sicurezza temporanee in IAM e Autorizzazioni per le credenziali di sicurezza temporanee.) La chiamata restituisce credenziali di sicurezza temporanee costituite da un ID della chiave di AWS accesso, una chiave di accesso segreta e un token di sessione. L'applicazione del gestore identità rende tali credenziali di sicurezza provvisorie disponibili all'applicazione aziendale interna. L'applicazione può quindi utilizzare le credenziali provvisorie per effettuare chiamate a AWS direttamente. L'app memorizza le credenziali finché non scadono e in seguito richiede un nuovo set di credenziali temporanee. L'immagine seguente illustra questo scenario.



Questo scenario ha i seguenti attributi:

- L'applicazione del gestore identità ha le autorizzazioni per accedere all'API di servizio token IAM (STS) per creare le credenziali di sicurezza temporanee.
- L'applicazione del gestore identità è in grado di verificare che i dipendenti siano autenticati nel sistema di autenticazione esistente.
- Gli utenti possono ottenere un URL temporaneo che consente loro di accedere alla console di AWS gestione (denominata single sign-on).

Per ulteriori informazioni sulla creazione di credenziali di sicurezza provvisorie, consultare Confronta le AWS STS credenziali. Per ulteriori informazioni sugli amministratori federati SAML che possono accedere alla Console di gestione, vedere. AWS Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS

Creazione di ruoli IAM

Per creare un ruolo, puoi utilizzare gli Console di gestione AWS strumenti per Windows o PowerShell l'API IAM. AWS CLI

Se utilizzi il Console di gestione AWS, una procedura guidata ti guida attraverso i passaggi per la creazione di un ruolo. La procedura guidata prevede passaggi leggermente diversi a seconda che tu stia creando un ruolo per un AWS servizio, per un responsabile Account AWS federato SAML o OIDC.

Ruoli per gli utenti IAM

Crea questo ruolo per delegare le autorizzazioni all'interno del tuo Account AWS o a ruoli definiti in altri di tua proprietà. Account AWS Un utente in un account può passare a un ruolo dello stesso o di un altro account. Mentre si usa il ruolo, l'utente è in grado di eseguire solo le azioni e accedere solo alle risorse consentite dal ruolo; le loro autorizzazioni utente originali sono sospese. Quando l'utente esce dal ruolo, le autorizzazioni utente originali vengono ripristinate.

Per ulteriori informazioni, consulta Creazione di un ruolo per fornire le autorizzazioni a un utente IAM.

Per ulteriori informazioni sulla creazione di ruoli per l'accesso multi-account, consulta Creare un ruolo utilizzando policy di attendibilità personalizzate.

Ruoli per AWS services

Creare questo ruolo per delegare le autorizzazioni per un servizio che può eseguire operazioni per tuo conto Un ruolo di servizio che passi a un servizio deve avere una policy IAM con le autorizzazioni che consentano al servizio di eseguire azioni associate a quel servizio. Sono necessarie autorizzazioni diverse per ciascuno dei servizi AWS .

Per ulteriori informazioni sulla creazione dei ruoli di servizio, consulta Crea un ruolo per delegare le autorizzazioni a un AWS service.

Per ulteriori informazioni sulla creazione di ruoli collegati al servizio, consulta Creare un ruolo collegato ai servizi.

Ruoli per la federazione delle identità

Crea questo ruolo per delegare le autorizzazioni agli utenti che hanno già identità esterne a AWS. Quando utilizzi un provider di identità, non devi creare un codice di accesso personalizzato né gestire le tue identità utente. Gli utenti esterni accedono tramite un IdP e tu puoi concedere a tali identità esterne le autorizzazioni per utilizzare le AWS risorse del tuo account. I provider di identità aiutano a proteggere il tuo AWS account perché non devi distribuire o incorporare credenziali di sicurezza a lungo termine, come le chiavi di accesso, nell'applicazione.

Per ulteriori informazioni, consulta Creazione di un ruolo per un provider di identità di terze parti.

Creazione di un ruolo per fornire le autorizzazioni a un utente IAM

Puoi utilizzare i ruoli IAM per fornire l'accesso alle tue AWS risorse. Con i ruoli IAM, puoi stabilire relazioni di fiducia tra il tuo account fiduciario e altri account AWS affidabili. L'account che concede fiducia possiede la risorsa alla quale accedere e l'account affidabile contiene gli utenti che devono

accedere alla risorsa. Tuttavia, è possibile che un altro account sia proprietario di una risorsa nell'account in uso. L'account che concede fiducia potrebbe infatti consentire all'account attendibile di creare nuove risorse, ad esempio creando nuovi oggetti in un bucket Amazon S3. In tal caso, l'account che crea la risorsa ne è proprietario e controlla chi può accedervi.

Dopo aver creato la relazione di fiducia, un utente IAM o un'applicazione dell'account affidabile può utilizzare l'operazione `AssumeRole`API Servizio di token di sicurezza AWS (AWS STS). Questa operazione fornisce credenziali di sicurezza temporanee che consentono l'accesso alle AWS risorse del tuo account.

Gli account possono essere controllati da sé stessi oppure l'account con gli utenti può essere controllato da terze parti. Se l'altro account con gli utenti è un account Account AWS che non controlli, puoi utilizzare l'`externalId`attributo. L'ID esterno può essere qualsiasi parola o numero concordato tra l'utente e l'amministratore dell'account di terze parti. Questa opzione aggiunge automaticamente una condizione alla policy di affidabilità che consente all'utente di assumere il ruolo solo se la richiesta include il corretto `sts:ExternalID`. Per ulteriori informazioni, consulta Accesso a Account AWS di proprietà di terzi.

Per informazioni su come utilizzare i ruoli per delegare le autorizzazioni, consultare Termini e concetti dei ruoli. Per informazioni sull'utilizzo di un ruolo di servizio per consentire l'accesso a risorse nel proprio account, consultare Crea un ruolo per delegare le autorizzazioni a un AWS service.

Creazione di un ruolo IAM (console)

Puoi utilizzare il Console di gestione AWS per creare un ruolo che un utente IAM può assumere. Ad esempio, supponiamo che l'organizzazione disponga di più Account AWS elementi per isolare un ambiente di sviluppo da un ambiente di produzione. Per informazioni di alto livello sulla creazione di un ruolo che consenta agli utenti nell'account di sviluppo di accedere alle risorse nell'account di produzione, consulta la sezione Esempio di uno scenario in cui si utilizzano account di sviluppo e produzione separati.

Autorizzazioni minime

Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:

- `access-analyzer:ValidatePolicy`
- `iam:AttachRolePolicy`
- `iam:CreatePolicy`

- `iam:CreateRole`
- `iam:GetAccountSummary`
- `iam:GetPolicy`
- `iam:GetPolicyVersion`
- `iam:GetRole`
- `iam:ListAccountAliases`
- `iam:ListAttachedRolePolicies`
- `iam:ListOpenIDConnectProviders`
- `iam:ListPolicies`
- `iam:ListRolePolicies`
- `iam:ListRoles`
- `iam:ListRoleTags`
- `iam:ListSAMLProviders`

Console

1. Accedi a Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, selezionare Ruoli e Crea ruolo.
3. Scegli il tipo di ruolo Account AWS.
4. Per creare un ruolo per il tuo account, scegli This account (Questo account). Per creare un ruolo per un altro account, scegli Altro Account AWS e inserisci l'ID account ID al quale desideri concedere l'accesso alle risorse.

   L'amministratore dell'account specificato può concedere l'autorizzazione di assumere questo ruolo a qualsiasi utente IAM in tale account. Per eseguire questa operazione, l'amministratore collega una policy all'utente o al gruppo che garantisce l'autorizzazione per l'operazione `sts:AssumeRole`. Tale policy deve specificare il nome ARN del ruolo `Resource`.
5. Per concedere le autorizzazioni agli utenti da un account di cui non hai il controllo e se tali utenti assumeranno il ruolo a livello di programmazione, seleziona Require external ID (Richiedi ID esterno). L'ID esterno può essere qualsiasi parola o numero concordato tra l'utente e l'amministratore dell'account di terze parti. Questa opzione aggiunge automaticamente una condizione alla policy di affidabilità che consente all'utente di assumere

il ruolo solo se la richiesta include il corretto `sts:ExternalID`. Per ulteriori informazioni, consulta Accesso a Account AWS di proprietà di terzi.

> **Important**
>
> La scelta di questa opzione limita l'accesso al ruolo solo tramite Tools for Windows PowerShell o l' AWS API. AWS CLI Questo perché non è possibile utilizzare la AWS console per passare a un ruolo che presenta una `externalId` condizione nella politica di attendibilità. Tuttavia, è possibile creare questo tipo di accesso a livello di codice scrivendo uno script o un'applicazione utilizzando il kit SDK rilevante. Per ulteriori informazioni e uno script di esempio, consulta Come abilitare l'accesso tra account alla Console di gestione AWS nel Blog sulla sicurezza di AWS .

6. Se si desidera limitare il ruolo agli utenti che accedono con la multi-factor authentication (MFA), selezionare Require MFA (Richiedi MFA). Questa opzione aggiunge una condizione alla policy di affidabilità del ruolo che controlla un accesso MFA. Un utente che desidera assumere il ruolo deve accedere con una password monouso temporanea da un dispositivo MFA configurato. Gli utenti senza autenticazione MFA non possono assumere il ruolo. Per ulteriori informazioni sulla funzionalità MFA, consultare AWS Multi-factor autenticazione in IAM.
7. Scegli Next (Successivo).
8. IAM include un elenco delle politiche AWS gestite e gestite dai clienti nel tuo account. Selezionare la policy delle autorizzazioni da utilizzare o scegliere Crea policy per aprire una nuova scheda del browser e creare una nuova policy da zero. Per ulteriori informazioni, consulta Creazione di policy IAM. Una volta creata la policy, chiudere la scheda e tornare alla scheda originale. Selezionare la casella di controllo accanto alle policy di autorizzazione da assegnare a chiunque assuma il ruolo. È anche possibile non selezionare le policy ora e collegarle al ruolo in un secondo momento. Per default, un ruolo non dispone di autorizzazioni.
9. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata.

   Aprire la sezione Set permissions boundary (Imposta limite delle autorizzazioni) e selezionare Use a permissions boundary to control the maximum role permissions (Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo). Selezionare la policy da utilizzare per il limite delle autorizzazioni.
10. Scegli Next (Successivo).

11. In Nome ruolo, immetti un nome per il ruolo. I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS. Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole. Quando un nome di ruolo viene visualizzato ai clienti nella console, ad esempio durante la procedura di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole. Poiché varie entità possono fare riferimento al ruolo, non puoi modificare il nome del ruolo dopo averlo creato.
12. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
13. Scegli Edit (Modifica) nelle sezioni Step 1: Select trusted entities (Fase 1: seleziona le entità attendibili) o Step 2: Add permissions (Fase 2: aggiungi autorizzazioni) per modificare i casi d'uso e le autorizzazioni per il ruolo. Verrai reindirizzato alle pagine precedenti per apportare le modifiche.
14. (Facoltativo) Aggiungere metadati al ruolo collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.
15. Rivedere il ruolo e scegliere Crea ruolo.

> **Important**
>
> Ricordare che questa è solo la prima metà della configurazione obbligatoria. È inoltre necessario fornire ai singoli utenti nell'account attendibile l'autorizzazione a passare al ruolo nella console o ad assumere il ruolo a livello di codice. Per ulteriori informazioni su questa fase, consultare Concedere le autorizzazioni agli utenti per cambiare ruoli.

Creazione di un ruolo IAM (AWS CLI)

La creazione di un ruolo da a AWS CLI comporta più passaggi. Quando si utilizza la console per creare un ruolo, molti passaggi vengono eseguiti automaticamente, ma con la console è AWS CLI necessario eseguire ogni passaggio in modo esplicito e autonomo. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Per creare un ruolo per accesso tra account (AWS CLI)

1. Creare un ruolo: aws iam create-role
2. Allega una politica di autorizzazioni gestite al ruolo: aws iam attach-role-policy

or

Crea una politica di autorizzazioni in linea per il ruolo: aws iam put-role-policy

3. (Facoltativo) Aggiungere attributi personalizzati al ruolo collegando tag: aws iam tag-role

   Per ulteriori informazioni, consulta Gestione dei tag sui ruoli IAM (AWS CLI or AWS API).

4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: aws iam put-role-permissions-boundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

L'esempio seguente mostra i primi due passaggi, più comuni, per la creazione di un ruolo per più account in un ambiente semplice. Questo esempio permette agli utenti dell'account `123456789012` di assumere il ruolo e visualizzare il bucket `example_bucket` di Amazon S3. L'esempio presuppone inoltre l'uso un computer client con Windows e che l'interfaccia a riga di comando sia già configurata con le credenziali dell'account e la regione. Per ulteriori informazioni, vedere Configurazione dell'interfaccia a AWS riga di comando.

In questo esempio, è necessario includere la seguente policy di attendibilità nel primo comando al momento della creazione del ruolo. Questa policy di attendibilità consente agli utenti dell'account `123456789012` di assumere il ruolo tramite l'operazione `AssumeRole`, ma solo se l'utente fornisce l'autenticazione MFA utilizzando i parametri `SerialNumber` e `TokenCode`. Per ulteriori informazioni sulla funzionalità MFA, consultare AWS Multi-factor autenticazione in IAM.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
          "Action": "sts:AssumeRole",
          "Condition": { "Bool": { "aws:MultiFactorAuthPresent": "true" } }
      }
  ]
}
```

> **Important**
>
> Se l'elemento `Principal` contiene l'ARN per un determinato utente o ruolo IAM, quando la policy viene salvata l'ARN viene trasformato in un ID principale univoco. Ciò aiuta a mitigare il rischio che qualcuno aumenti le proprie autorizzazioni rimuovendo e ricreando il ruolo o l'utente. Questo ID non è normalmente presente nella console, perché avviene anche una trasformazione inversa nell'ARN quando la policy di affidabilità viene visualizzata. Tuttavia, se si elimina il ruolo o l'utente, l'ID principale viene visualizzato nella console perché non è più AWS possibile mapparlo su un ARN. Pertanto, se si elimina e crea nuovamente un utente o un ruolo a cui viene fatto riferimento in un elemento `Principal` della policy di attendibilità, è necessario modificare il ruolo per sostituire l'ARN.

Quando si utilizza il secondo comando, è necessario collegare una policy gestita esistente al ruolo. La policy delle autorizzazioni seguente consente agli utenti che assumono il ruolo di eseguire solo l'operazione `ListBucket` sul bucket `example_bucket` di Amazon S3.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": "s3:ListBucket",
          "Resource": "arn:aws:s3:::example_bucket"
      }
  ]
}
```

Per creare questo ruolo `Test-UserAccess-Role`, è prima necessario salvare la policy di attendibilità precedente con il nome `trustpolicyforacct123456789012.json` nella cartella `policies` dell'unità `C:` locale. Quindi salva la precedente politica di autorizzazione come politica gestita dai clienti nel tuo account Account AWS con il nome. `PolicyForRole` È quindi possibile utilizzare i comandi seguenti per creare il ruolo e collegare la policy gestita.

```
# Create the role and attach the trust policy file that allows users in the specified
 account to assume the role.
```

```
$ aws iam create-role --role-name Test-UserAccess-Role --assume-role-policy-document
 file://C:\policies\trustpolicyforacct123456789012.json

# Attach the permissions policy (in this example a managed policy) to the role to
 specify what it is allowed to do.
$ aws iam attach-role-policy --role-name Test-UserAccess-Role --policy-arn
 arn:aws:iam::123456789012:policy/PolicyForRole
```

Important

Ricordare che questa è solo la prima metà della configurazione obbligatoria. È inoltre necessario fornire a singoli utenti nell'account affidabile le autorizzazioni per passare al ruolo. Per ulteriori informazioni su questa fase, consultare Concedere le autorizzazioni agli utenti per cambiare ruoli.

Dopo aver creato il ruolo e avergli concesso le autorizzazioni per eseguire AWS attività o accedere alle AWS risorse, qualsiasi utente dell'123456789012account può assumere il ruolo. Per ulteriori informazioni, consulta Passa a un ruolo IAM (AWS CLI).

Creazione di un ruolo IAM (AWS API)

La creazione di un ruolo dall' AWS API prevede diversi passaggi. Quando si usa la console per creare un ruolo, molti dei passaggi vengono eseguiti automaticamente, ma con l'API ogni passaggio deve essere eseguito esplicitamente dall'utente. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Creare un ruolo nel codice (AWS API)

1. Crea un ruolo: CreateRole

   Per la policy di affidabilità del ruolo, è possibile specificare una posizione del file.

2. Allega una politica di autorizzazione gestita al ruolo: AttachRolePolicy

   or

   Crea una politica di autorizzazione in linea per il ruolo: PutRolePolicy

> **Important**
>
> Ricordare che questa è solo la prima metà della configurazione obbligatoria. È inoltre necessario fornire a singoli utenti nell'account affidabile le autorizzazioni per passare al ruolo. Per ulteriori informazioni su questa fase, consultare Concedere le autorizzazioni agli utenti per cambiare ruoli.

3. (Facoltativo) Aggiungi attributi personalizzati al ruolo allegando tag: TagRole

   Per ulteriori informazioni, consulta Gestione dei tag sui ruoli IAM (AWS CLI or AWS API).

4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: PutRolePermissionsBoundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

Dopo aver creato il ruolo e avergli concesso le autorizzazioni per eseguire AWS attività o accedere alle AWS risorse, è necessario concedere le autorizzazioni agli utenti dell'account per consentire loro di assumere il ruolo. Per ulteriori informazioni sull'assunzione di un ruolo, consulta Passa a un ruolo IAM (AWS API).

Creazione di un ruolo IAM (AWS CloudFormation)

Per informazioni sulla creazione di un ruolo IAM in AWS CloudFormation, consulta il riferimento alle risorse e alle proprietà e gli esempi nella Guida per l'AWS CloudFormation utente.

Per ulteriori informazioni sui modelli IAM in AWS CloudFormation, consulta gli snippet di AWS Identity and Access Management modello nella Guida per l'AWS CloudFormation utente.

Crea un ruolo per delegare le autorizzazioni a un AWS service

Molti AWS servizi richiedono l'utilizzo di ruoli per consentire al servizio di accedere alle risorse di altri servizi per conto dell'utente. Un ruolo che un servizio assume per eseguire operazioni a tuo nome viene chiamato ruolo del servizio. Quando un ruolo fornisce uno scopo specializzato per un servizio, questo può essere categorizzato come ruolo collegato al servizio. Per visualizzare i servizi che supportano ruoli collegati ai servizi, oppure se un servizio supporta qualsiasi forma di credenziali provvisorie, consulta AWS servizi supportati da IAM. Per apprendere come un singolo servizio utilizza i ruoli, scegli il nome del servizio nella tabella e visualizza la documentazione relativa a tale servizio.

Quando imposti l'autorizzazione `PassRole`, devi assicurarti che un utente non invii un ruolo dove il ruolo dispone di più autorizzazioni di quelle che desideri che l'utente abbia. Ad esempio, Alice potrebbe non essere autorizzata a eseguire alcune operazioni su Amazon S3. Se Alice potesse trasferire un ruolo a un servizio che consente le azioni di Amazon S3, il servizio potrebbe eseguire azioni Amazon S3 per conto di Alice durante l'esecuzione del processo.

Per informazioni su come i ruoli aiutano a delegare le autorizzazioni, consulta Termini e concetti dei ruoli.

Autorizzazioni del ruolo del servizio

Per consentire a una entità IAM (utente o ruolo) di creare o modificare un ruolo di servizio, occorre configurare le autorizzazioni.

> **Note**
>
> L'ARN per un ruolo collegato ai servizi include un principale del servizio, indicata nelle policy seguenti come *`SERVICE-NAME`*`.amazonaws.com`. Non tentare di indovinare il principale del servizio, perché fa distinzione tra maiuscole e minuscole e il formato può variare tra i servizi AWS . Per visualizzare l'entità principale di un servizio, consulta la relativa documentazione del ruolo collegato al servizio.

Come consentire a un'entità IAM di creare un ruolo di servizio specifico

Aggiungi la policy seguente all'entità IAM che deve creare il ruolo di servizio. Questa policy ti permette di creare un ruolo del servizio per il servizio specificato e utilizzando un nome specifico. Puoi quindi collegare le policy gestite o inline a tale ruolo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:CreateRole",
```

```
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME"
        }
    ]
}
```

Come consentire a un'entità IAM di creare un qualsiasi ruolo di servizio

AWS consiglia di consentire solo agli utenti amministrativi di creare qualsiasi ruolo di servizio. Una persona con autorizzazioni per creare un ruolo e allegare qualsiasi policy può eseguire l'escalation delle proprie autorizzazioni. Invece, crea una policy che consenta a questa persona di creare solo i ruoli di cui hanno bisogno o lascia che un amministratore crei il ruolo di servizio per suo conto.

Per allegare una policy che consenta a un amministratore di accedere all'intero account Account AWS, utilizza la policy AdministratorAccess AWS gestita.

Come consentire a un'entità IAM di modificare un ruolo di servizio

Aggiungi la policy seguente all'entità IAM che deve modificare il ruolo di servizio.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EditSpecificServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:ListAttachedRolePolicies",
                "iam:ListRolePolicies",
                "iam:PutRolePolicy",
                "iam:UpdateRole",
                "iam:UpdateRoleDescription"
            ],
```

```
            "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME"
        },
        {
            "Sid": "ViewRolesAndPolicies",
            "Effect": "Allow",
            "Action": [
                "iam:GetPolicy",
                "iam:ListRoles"
            ],
            "Resource": "*"
        }
    ]
}
```

Come consentire a un'entità IAM di eliminare un ruolo di servizio specifico

Aggiungi l'istruzione seguente alla policy delle autorizzazioni per l'entità IAM che deve eliminare il ruolo di servizio specificato.

```
{
    "Effect": "Allow",
    "Action": "iam:DeleteRole",
    "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME"
}
```

Come consentire a un'entità IAM di eliminare qualunque ruolo di servizio

AWS consiglia di consentire solo agli utenti amministrativi di eliminare qualsiasi ruolo di servizio. Invece, crea una policy che consenta loro di eliminare solo i ruoli di cui hanno bisogno o lascia che un amministratore elimini il ruolo di servizio per suo conto.

Per allegare una policy che consenta a un amministratore di accedere all'intero account Account AWS, utilizza la policy AdministratorAccess AWS gestita.

Creare un ruolo per un AWS servizio (console)

È possibile utilizzare Console di gestione AWS per creare un ruolo per un servizio. Dal momento che alcuni servizi supportano più ruoli del servizio, consulta la documentazione AWS relativa al servizio per determinare quale caso d'uso selezionare. È possibile apprendere come assegnare le necessarie policy di affidabilità e autorizzazioni al ruolo, in modo che il servizio possa assumere quel ruolo per

conto dell'utente. Le operazioni che è possibile utilizzare per controllare le autorizzazioni per il tuo ruolo possono variare, a seconda del modo in cui il servizio definisce i casi d'uso e della creazione o meno di un ruolo collegato ai servizi.

Console

Per creare un ruolo per un Servizio AWS (console IAM)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Tipo di entità attendibile, seleziona Servizio AWS.
4. Per Servizio o caso d'uso, scegli un servizio, quindi scegli il caso d'uso. I casi d'uso sono definiti dal servizio per includere la politica di fiducia richiesta dal servizio.
5. Scegli Next (Successivo).
6. Per Policy di autorizzazione, le opzioni dipendono dal caso d'uso selezionato:
   - Se il servizio definisce le autorizzazioni per il ruolo, le policy di autorizzazioni non possono essere selezionate.
   - Seleziona una policy da un set limitato di policy di autorizzazione.
   - Seleziona una policy tra tutte le policy di autorizzazione.
   - Non selezionare policy di autorizzazioni, crea le policy dopo la creazione del ruolo e quindi collegale al ruolo.
7. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una funzionalità avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.
   a. Apri la sezione Imposta limite delle autorizzazioni e seleziona Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo.

      IAM include un elenco delle politiche AWS gestite e gestite dal cliente nel tuo account.
   b. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
8. Scegli Next (Successivo).
9. Per Nome del ruolo, le opzioni dipendono dal servizio:
   - Se il servizio definisce il nome del ruolo, non puoi modificarlo.
   - Se il servizio definisce un prefisso per il nome del ruolo, puoi inserire un suffisso facoltativo.

- Se il servizio non definisce il nome del ruolo, puoi assegnare un nome al ruolo.

> **Important**
>
> Quando assegni un nome a un ruolo, tieni presente quanto segue:
>
> - I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS e non possono essere resi univoci in base alle maiuscole e minuscole.
>
> Ad esempio, non creare ruoli denominati **`PRODROLE`** e **`prodrole`**. Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole, tuttavia quando un nome di ruolo viene visualizzato ai clienti nella console, ad esempio durante il processo di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole.
> - Non è possibile modificare il nome del ruolo dopo averlo creato, in quanto altre entità possono fare riferimento al ruolo.

10. (Facoltativo) In Descrizione, inserisci una descrizione per il ruolo.
11. (Facoltativo) Per modificare i casi d'uso e le autorizzazioni per il ruolo, in Fase 1: seleziona le entità attendibili o Fase 2: aggiungi autorizzazioni seleziona Modifica.
12. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, aggiungi i tag come coppie chiave-valore. Per ulteriori informazioni sull'uso dei tag in IAM, consulta Tag for AWS Identity and Access Management resources nella IAM User Guide.
13. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

Creazione di un ruolo per un servizio (AWS CLI)

La creazione di un ruolo da AWS CLI richiede più passaggi. Quando usi la console per creare un ruolo, molti passaggi vengono eseguiti automaticamente, ma con essi AWS CLI devi eseguire esplicitamente ogni passaggio tu stesso. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Se il servizio in uso è Amazon EC2, è necessario creare anche un profilo dell'istanza e aggiungervi il ruolo. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Per creare un ruolo per un AWS servizio di AWS CLI

1. Il `create-role` comando seguente crea un ruolo denominato Test-Role e gli allega una politica di fiducia:

```
aws iam create-role --role-name Test-Role --assume-role-policy-document
file://Test-Role-Trust-Policy.json
```

2. Collega una policy delle autorizzazioni gestita al ruolo: aws iam attach-role-policy

   Ad esempio, il seguente comando `attach-role-policy` allega la policy gestita AWS denominata `ReadOnlyAccess` al ruolo IAM denominato `ReadOnlyRole`:

   ```
   aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
   ReadOnlyAccess --role-name ReadOnlyRole
   ```

   or

   Creare una policy delle autorizzazioni inline per il ruolo: aws iam put-role-policy

   Per aggiungere una policy di autorizzazioni in linea, consulta l'esempio seguente:

   ```
   aws iam put-role-policy --role-name Test-Role --policy-name
   ExamplePolicy --policy-document file://AdminPolicy.json
   ```

3. (Facoltativo) Aggiungere attributi personalizzati al ruolo collegando tag: aws iam tag-role

   Per ulteriori informazioni, consulta Gestione dei tag sui ruoli IAM (AWS CLI or AWS API).

4. (Facoltativo) Impostare il limite delle autorizzazioni per il ruolo: aws iam put-role-permissions-boundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata AWS .

Se intendi utilizzare il ruolo con Amazon EC2 o un altro AWS servizio che utilizza Amazon EC2, devi memorizzare il ruolo in un profilo di istanza. Un profilo dell'istanza è un container per un ruolo che può essere associato a un'istanza Amazon EC2 quando viene avviato. Un profilo dell'istanza può contenere un solo ruolo e tale limite non può essere aumentato. Se crei il ruolo utilizzando il Console di gestione AWS, il profilo di istanza viene creato automaticamente con lo stesso nome del ruolo. Per ulteriori informazioni sui profili delle istanze, consulta Usare profili dell'istanza. Per informazioni su come avviare un'istanza EC2 con un ruolo, consulta Controllo dell'accesso alle risorse Amazon EC2 nella Guida per l'utente di Amazon EC2.

Per creare un profilo di istanza e memorizzare il ruolo in esso (AWS CLI)

1. Creare un profilo dell'istanza: aws iam create-instance-profile
2. Aggiungere il ruolo al profilo dell'istanza: aws iam add-role-to-instance-profile

Il set di comandi di AWS CLI esempio riportato di seguito illustra i primi due passaggi per creare un ruolo e assegnare le autorizzazioni. Mostra inoltre i due passaggi necessari per creare un profilo dell'istanza e aggiungere il ruolo al profilo. Questa policy di attendibilità di esempio permette al servizio Amazon EC2 di assumere il ruolo e visualizzare il bucket `example_bucket` di Amazon S3. L'esempio presuppone inoltre l'uso un computer client con Windows e che l'interfaccia a riga di comando sia già configurata con le credenziali dell'account e la regione. Per ulteriori informazioni, vedere Configurazione dell'interfaccia a riga di comando. AWS

In questo esempio, è necessario includere la seguente policy di attendibilità nel primo comando al momento della creazione del ruolo. La policy di attendibilità consente al servizio Amazon EC2 di assumere il ruolo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"Service": "ec2.amazonaws.com"},
    "Action": "sts:AssumeRole"
  }
}
```

Quando si utilizza il secondo comando, è necessario collegare una policy di autorizzazione al ruolo. L'esempio di policy di autorizzazione seguente consente al ruolo di eseguire solo l'operazione `ListBucket` sul bucket `example_bucket` di Amazon S3.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
```

```
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::example_bucket"
  }
}
```

Per creare questo ruolo `Test-Role-for-EC2`, è innanzitutto necessario salvare la policy di attendibilità precedente con il nome `trustpolicyforec2.json` e la policy di autorizzazione precedente con il nome `permissionspolicyforec2.json` nella directory `policies` dell'unità `C:` locale. È quindi possibile utilizzare i comandi seguenti per creare il ruolo, collegare la policy, creare il profilo dell'istanza e aggiungere il ruolo al profilo dell'istanza.

```
# Create the role and attach the trust policy that allows EC2 to assume this role.
$ aws iam create-role --role-name Test-Role-for-EC2 --assume-role-policy-document
 file://C:\policies\trustpolicyforec2.json

# Embed the permissions policy (in this example an inline policy) to the role to
 specify what it is allowed to do.
$ aws iam put-role-policy --role-name Test-Role-for-EC2 --policy-name Permissions-
Policy-For-Ec2 --policy-document file://C:\policies\permissionspolicyforec2.json

# Create the instance profile required by EC2 to contain the role
$ aws iam create-instance-profile --instance-profile-name EC2-ListBucket-S3

# Finally, add the role to the instance profile
$ aws iam add-role-to-instance-profile --instance-profile-name EC2-ListBucket-S3 --
role-name Test-Role-for-EC2
```

Quando avvii l'istanza EC2, specifica il nome del profilo dell'istanza nella pagina Configura i dettagli dell'istanza se utilizzi la AWS console. Se utilizzi il comando della CLI `aws ec2 run-instances`, specifica il parametro `--iam-instance-profile`.

Creazione di un ruolo per un servizio (AWS API)

La creazione di un ruolo dall' AWS API prevede diversi passaggi. Quando si usa la console per creare un ruolo, molti dei passaggi vengono eseguiti automaticamente, ma con l'API ogni passaggio deve essere eseguito esplicitamente dall'utente. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Se il servizio in uso è Amazon EC2, è necessario creare anche un profilo dell'istanza e aggiungervi il ruolo. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Creare un ruolo per un AWS servizio (AWS API)

1. Crea un ruolo: CreateRole

   Per la policy di affidabilità del ruolo, è possibile specificare una posizione del file.
2. Allega una politica di autorizzazioni gestite al ruolo: AttachRolePolicy

   or

   Crea una politica di autorizzazioni in linea per il ruolo: PutRolePolicy
3. (Facoltativo) Aggiungi attributi personalizzati al ruolo allegando tag: TagRole

   Per ulteriori informazioni, consulta Gestione dei tag sui ruoli IAM (AWS CLI or AWS API).
4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: PutRolePermissionsBoundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

Se intendi utilizzare il ruolo con Amazon EC2 o un altro AWS servizio che utilizza Amazon EC2, devi memorizzare il ruolo in un profilo di istanza. Un profilo dell'istanza è un container per un ruolo. Ogni profilo dell'istanza può contenere un solo ruolo e tale limite non può essere superato. Se crei il ruolo in Console di gestione AWS, il profilo dell'istanza viene creato per te con lo stesso nome del ruolo. Per ulteriori informazioni sui profili delle istanze, consulta Usare profili dell'istanza. Per informazioni su come avviare un'istanza Amazon EC2 con un ruolo, consulta Controllo dell'accesso alle risorse Amazon EC2 nella Guida per l'utente di Amazon EC2.

Per creare un profilo di istanza e memorizzare il ruolo in esso (AWS API)

1. Crea un profilo di istanza: CreateInstanceProfile
2. Aggiungi il ruolo al profilo dell'istanza: AddRoleToInstanceProfile

Creare un ruolo collegato ai servizi

Un ruolo collegato ai servizi è un tipo unico di ruolo IAM collegato direttamente a un AWS servizio. Service-linked i ruoli sono predefiniti dal servizio e includono tutte le autorizzazioni necessarie al servizio per chiamare altri AWS servizi per conto dell'utente. Il servizio collegato definisce anche le modalità di creazione, modifica ed eliminazione di un ruolo collegato al servizio. Un servizio può creare o eliminare automaticamente il ruolo. È possibile che ti permetta di creare, modificare o

eliminare il ruolo come parte di una procedura guidata o un processo nel servizio. Oppure potrebbe richiedere l'utilizzo di IAM per creare o eliminare il ruolo. Indipendentemente dal metodo, i ruoli collegati ai servizi semplificano la procedura di configurazione di un servizio poiché non dovrai più aggiungere manualmente le autorizzazioni necessarie ai servizi per completare le operazioni per tuo conto.

> **Note**
>
> Ricorda che i ruoli di servizio sono diversi dai ruoli collegati ai servizi. Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM. Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo di eseguire un'azione per conto dell'utente. Service-linked i ruoli vengono visualizzati nell'utente Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Il servizio collegato definisce le autorizzazioni dei relativi ruoli collegati ai servizi e, a meno che non sia stato stabilito diversamente, solo quel servizio può assumere i ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni. Una policy delle autorizzazioni specifica non può essere collegata a un'altra entità IAM.

Prima di poter eliminare i ruoli, devi eliminare le risorse associate. Questo ti aiuta a evitare di rimuovere involontariamente l'autorizzazione ad accedere alle risorse.

> **Tip**
>
> Per informazioni sui servizi che supportano l'utilizzo di ruoli collegati ai servizi, vedi AWS servizi supportati da IAM e cerca i servizi che hanno Sì nella colonna Service-Linked Ruolo. Scegliere Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Service-linked autorizzazioni relative ai ruoli

Per consentire a un utente o un ruolo di creare o modificare un ruolo collegato ai servizi, devi configurare le autorizzazioni per un'entità IAM (utente o ruolo).

> **Note**
>
> L'ARN per un ruolo collegato ai servizi include un'entità principale del servizio, indicata nelle policy seguenti come *SERVICE-NAME*.amazonaws.com. Non cercate di indovinare l'entità del servizio, perché fa distinzione tra maiuscole e minuscole e il formato può variare tra AWS i servizi. Per visualizzare l'entità principale di un servizio, consulta la relativa documentazione del ruolo collegato al servizio.

Per consentire a un'entità IAM di creare un ruolo specifico collegato ai servizi

Aggiungi la policy seguente a un'entità IAM che deve creare il ruolo collegato ai servizi.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-
NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*",
            "Condition": {"StringLike": {"iam:AWSServiceName": "SERVICE-
NAME.amazonaws.com"}}
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-
NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*"
        }
    ]
}
```

Come consentire a un'entità IAM di creare qualunque ruolo collegato ai servizi

Aggiungi la seguente istruzione alla policy delle autorizzazioni per l'entità IAM che deve creare un ruolo collegato ai servizi o qualunque ruolo di servizio che include le policy di cui ha bisogno. Questa istruzione della policy non consente all'entità IAM di collegare una policy al ruolo.

```
{
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

Come consentire a un'entità IAM di modificare la descrizione di qualunque ruolo di servizio

Aggiungi la seguente istruzione alla policy delle autorizzazioni per l'entità IAM che deve modificare la descrizione di un ruolo collegato ai servizi o qualunque ruolo di servizio.

```
{
    "Effect": "Allow",
    "Action": "iam:UpdateRoleDescription",
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

Come consentire a un'entità IAM di eliminare un ruolo collegato ai servizi specifico

Aggiungi la seguente istruzione alla policy delle autorizzazioni per l'entità IAM che deve eliminare il ruolo collegato ai servizi.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-
NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*"
}
```

Come consentire a un'entità IAM di eliminare qualunque ruolo collegato ai servizi

Aggiungi la seguente istruzione alla policy delle autorizzazioni per l'entità IAM che deve eliminare un ruolo collegato ai servizi ma non il ruolo di servizio.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

Come consentire a un'entità IAM di passare un ruolo esistente al servizio

Alcuni AWS servizi consentono di passare un ruolo esistente al servizio, invece di creare un nuovo ruolo collegato al servizio. Per eseguire questa operazione, un utente deve disporre delle autorizzazioni per passare il ruolo al servizio. Aggiungi l'istruzione seguente alla policy delle autorizzazioni per l'entità IAM che deve passare un ruolo. Questa istruzione della policy consente anche all'entità di visualizzare un elenco di ruoli da cui è possibile scegliere il ruolo da passare. Per ulteriori informazioni, consulta Concedi a un utente le autorizzazioni per passare un ruolo a un AWS service.

```
{
  "Sid": "PolicyStatementToAllowUserToListRoles",
  "Effect": "Allow",
  "Action": ["iam:ListRoles"],
  "Resource": "*"
},
{
  "Sid": "PolicyStatementToAllowUserToPassOneSpecificRole",
  "Effect": "Allow",
  "Action": [ "iam:PassRole" ],
  "Resource": "arn:aws:iam::account-id:role/my-role-for-XYZ"
}
```

Autorizzazioni indirette con ruoli collegati al servizio

Le autorizzazioni concesse da un ruolo collegato ai servizi possono essere indirettamente trasferite ad altri utenti e ruoli. Quando un ruolo collegato al servizio viene utilizzato da un AWS servizio, quel ruolo collegato al servizio può utilizzare le proprie autorizzazioni per chiamare altri servizi. AWS Ciò significa che gli utenti e i ruoli con le autorizzazioni per chiamare un servizio che utilizza un ruolo collegato al servizio possono avere accesso indiretto ai servizi a cui può accedere quel ruolo collegato al servizio.

Ad esempio, quando crei un'istanza database Amazon RDS, un ruolo collegato ai servizi per RDS viene creato automaticamente se non ne esiste già uno. Questo ruolo collegato al servizio consente a RDS di chiamare Amazon EC2, Amazon SNS, Amazon Logs e Amazon Kinesis per tuo conto. CloudWatch Se consenti agli utenti e ai ruoli del tuo account di modificare o creare database RDS, potrebbero interagire indirettamente con Amazon EC2, Amazon SNS, i CloudWatch log di Amazon Logs e le risorse Amazon Kinesis chiamando RDS, poiché RDS utilizzerebbe il suo ruolo collegato ai servizi per accedere a tali risorse.

Metodi per creare un ruolo collegato ai servizi

Il metodo utilizzato per creare un ruolo collegato ai servizi dipende dal servizio. In alcuni casi, non devi creare manualmente un ruolo collegato ai servizi. Ad esempio, quando completi un'azione specifica (ad esempio la creazione di una risorsa) nel servizio, il servizio potrebbe creare il ruolo collegato ai servizi per te. O se stavi utilizzando un servizio prima di iniziare il supporto ai ruoli collegati ai servizi, allora il servizio potrebbe aver creato automaticamente il ruolo nel tuo account. Per ulteriori informazioni, consulta È apparso un nuovo ruolo nel mio AWS account.

In altri casi, il servizio può supportare la creazione di un ruolo collegato ai servizi manualmente utilizzando la console di servizio, le API o la CLI. Per informazioni sui servizi che supportano l'utilizzo di ruoli collegati ai servizi, consulta AWS servizi supportati da IAM e cerca i servizi con Sì nella colonna Ruolo. Service-Linked Per scoprire se il servizio supporta la creazione del ruolo collegato ai servizi, selezionare il link Sì per visualizzare il ruolo collegato ai servizi per quel servizio.

Se il servizio non supporta la creazione del ruolo, è possibile utilizzare IAM per creare il ruolo collegato ai servizi.

> **Important**
>
> Service-linked i ruoli vengono conteggiati ai fini dei ruoli IAM Account AWS entro un limite, ma se hai raggiunto il limite, puoi comunque creare ruoli collegati ai servizi nel tuo account. Solo i ruoli collegati ai servizi possono superare il limite.

Creazione di un ruolo collegato ai servizi (console)

Prima di creare un ruolo collegato ai servizi in IAM, scopri se il servizio collegato crea automaticamente i ruoli collegati ai servizi; inoltre, scopri se è possibile creare il ruolo dalla console del servizio, dall'API o dalla CLI.

Come creare un ruolo collegato ai servizi (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel pannello di navigazione della console IAM seleziona Ruoli. Quindi seleziona Create role (Crea ruolo).
3. Scegli il tipo di ruolo di servizio AWS .
4. Scegli il caso d'uso per il servizio. I casi d'uso sono definiti dal servizio in modo da includere la policy di attendibilità richiesta dal servizio. Quindi, seleziona Successivo.
5. Scegli una o più policy di autorizzazione da collegare al ruolo. A seconda del caso d'uso selezionato, il servizio può eseguire una di queste operazioni:

   - Definire le autorizzazioni utilizzate dal ruolo.
   - Consentire di scegliere tra un set limitato di autorizzazioni.
   - Consentire di scegliere qualsiasi autorizzazione.
   - Ti consente di non selezionare policy in questo momento, creare le policy successivamente e quindi collegarle al ruolo.

   Seleziona la casella di controllo accanto alla policy che assegna le autorizzazioni desiderate per il ruolo, quindi scegli Successivo.

   > **Note**
   >
   > Le autorizzazioni specificate sono disponibili per qualsiasi entità che utilizza il ruolo. Per default, un ruolo non dispone di autorizzazioni.

6. Il grado di personalizzazione per Nome ruolo viene definito dal servizio. Se il servizio definisce il nome del ruolo, allora questa opzione non può essere modificata. In altri casi, il servizio può definire un prefisso per il ruolo e consentirti di inserire un suffisso opzionale.

   Se possibile, inserisci il suffisso del nome del ruolo da aggiungere al nome predefinito. Il suffisso consente di identificare lo scopo del ruolo. I nomi dei ruoli devono essere univoci all'interno dell'account AWS . Non si distinguono per maiuscole e minuscole. Ad esempio, non è possibile creare ruoli denominati sia **`<service-linked-role-name>_SAMPLE`** che **`<service-linked-role-name>_sample`**. Poiché varie entità possono fare riferimento al ruolo, non è possibile modificare il nome del ruolo dopo averlo creato.

7. (Facoltativo) In Description (Descrizione), modifica la descrizione per il nuovo ruolo collegato ai servizi.
8. Non è possibile collegare tag ai ruoli collegati ai servizi durante la creazione. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.
9. Rivedere il ruolo e scegliere Crea ruolo.

Creazione di un ruolo collegato al servizio (AWS CLI)

Prima di creare un ruolo collegato ai servizi in IAM, scopri se il servizio collegato crea automaticamente i ruoli collegati ai servizi e se è possibile creare il ruolo dalla CLI del servizio. Se la CLI del servizio non è supportata, puoi usare i comandi IAM per creare un ruolo collegato ai servizi con la policy di attendibilità e le policy in linea che il servizio richiede per assumere il ruolo.

Per creare un ruolo collegato ai servizi (AWS CLI)

Esegui il comando seguente:

```
aws iam create-service-linked-role --aws-service-name SERVICE-NAME.amazonaws.com
```

Creazione di un ruolo collegato al servizio (AWS API)

Prima di creare un ruolo collegato ai servizi in IAM, scopri se il servizio collegato crea automaticamente i ruoli collegati ai servizi e scopri se è possibile creare il ruolo dalle API del servizio. Se l'API del servizio non è supportata, puoi utilizzare l' AWS API per creare un ruolo collegato al servizio con la politica di fiducia e le politiche in linea necessarie al servizio per assumere il ruolo.

Per creare un ruolo collegato al servizio (API)AWS

Utilizzare la chiamata API CreateServiceLinkedRole. Nella richiesta, specificare un nome del servizio di *SERVICE_NAME_URL*`.amazonaws.com`.

Ad esempio, per creare il ruolo collegato ai servizi Lex Bots (Bot di Lex), utilizzare `lex.amazonaws.com`.

Creazione di un ruolo per un provider di identità di terze parti

Puoi utilizzare provider di identità invece di creare utenti IAM nel tuo Account AWS. Con un provider di identità (IdP), puoi gestire le tue identità utente all'esterno AWS e concedere a queste identità utente esterne le autorizzazioni per accedere AWS alle risorse del tuo account. Per ulteriori

informazioni sulla federazione e sui provider di identità, consultare Fornitori di identità e federazione in AWS.

Creazione di un ruolo per i principali federati OIDC e SAML (console)

Le procedure per la creazione di un ruolo dipendono dalla scelta dei provider di terze parti:

- Per OpenID Connect (OIDC), consulta Creare un ruolo per la federazione OpenID Connect (console).
- Per SAML 2.0, consulta Creare un ruolo per una federazione SAML 2.0 (console).

Creazione di un ruolo per l'accesso federato (AWS CLI)

Le procedure per creare un ruolo per il provider di identità supportato (OIDC o SAML) dalla AWS CLI sono identiche. La differenza consiste nel contenuto della policy di affidabilità creata in passaggi preliminari. Inizia seguendo le fasi descritte nella sezione dei prerequisiti per il tipo di provider in uso:

- Per un provider OIDC, consulta Prerequisiti per la creazione di un ruolo per OIDC.
- Per un provider SAML, consulta Prerequisiti per la creazione di un ruolo per SAML.

La creazione di un ruolo da richiede diversi passaggi. AWS CLI Quando si utilizza la console per creare un ruolo, molti passaggi vengono eseguiti automaticamente, ma con la console AWS CLI è necessario eseguire esplicitamente ogni passaggio da soli. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Per creare un ruolo (AWS CLI)

1. Creare un ruolo: aws iam create-role
2. Allega una politica di autorizzazioni al ruolo: aws iam attach-role-policy

   or

   Crea una politica di autorizzazioni in linea per il ruolo: aws iam put-role-policy
3. (Facoltativo) Aggiungere attributi personalizzati al ruolo collegando tag: aws iam tag-role

   Per ulteriori informazioni, consulta Gestione dei tag sui ruoli IAM (AWS CLI or AWS API).
4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: aws iam put-role-permissions-boundary

Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

L'esempio seguente mostra i primi due passaggi, più comuni, per la creazione di un ruolo del provider di identità in un ambiente semplice. Questo esempio permette agli utenti dell'account `123456789012` di assumere il ruolo e visualizzare il bucket `example_bucket` di Amazon S3. Questo esempio presuppone inoltre che tu stia eseguendo Windows AWS CLI su un computer che esegue Windows e che lo abbia già configurato AWS CLI con le tue credenziali. Per ulteriori informazioni, consultare la pagina relativa alla configurazione di AWS Command Line Interface.

La policy di attendibilità di esempio riportata di seguito è progettata per un'app per dispositivi mobili in cui l'utente accede tramite Amazon Cognito. In questo esempio, *`us-east:12345678-ffff-ffff-ffff-123456`* rappresenta l'ID del pool di identità assegnato da Amazon Cognito.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Sid": "RoleForCognito",
        "Effect": "Allow",
        "Principal": {"Federated": "cognito-identity.amazonaws.com"},
        "Action": "sts:AssumeRoleWithWebIdentity",
        "Condition": {"StringEquals": {"cognito-identity.amazonaws.com:aud": "us-
east:12345678-ffff-ffff-ffff-123456"}}
    }
}
```

La policy delle autorizzazioni seguente consente agli utenti che assumono il ruolo di eseguire solo l'operazione `ListBucket` sul bucket `example_bucket` di Amazon S3.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
```

```
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::example_bucket"
  }
}
```

Per creare questo ruolo `Test-Cognito-Role`, è prima necessario salvare la policy di attendibilità precedente con il nome `trustpolicyforcognitofederation.json` e la policy di autorizzazione precedente con il nome `permspolicyforcognitofederation.json` nella cartella `policies` dell'unità `C:` locale. È quindi possibile utilizzare i comandi seguenti per creare il ruolo e collegare la policy inline.

```
# Create the role and attach the trust policy that enables users in an account to
 assume the role.
$ aws iam create-role --role-name Test-Cognito-Role --assume-role-policy-document
 file://C:\policies\trustpolicyforcognitofederation.json

# Attach the permissions policy to the role to specify what it is allowed to do.
aws iam put-role-policy --role-name Test-Cognito-Role --policy-name
 Perms-Policy-For-CognitoFederation --policy-document file://C:\policies
\permspolicyforcognitofederation.json
```

Creazione di un ruolo per l'accesso federato (API)AWS

Le procedure per creare un ruolo per il provider di identità supportato (OIDC o SAML) dalla AWS CLI sono identiche. La differenza consiste nel contenuto della policy di affidabilità creata in passaggi preliminari. Inizia seguendo le fasi descritte nella sezione dei prerequisiti per il tipo di provider in uso:

- Per un provider OIDC, consulta Prerequisiti per la creazione di un ruolo per OIDC.
- Per un provider SAML, consulta Prerequisiti per la creazione di un ruolo per SAML.

Per creare un ruolo (AWS API)

1. Crea un ruolo: CreateRole
2. Allega una politica di autorizzazioni al ruolo: AttachRolePolicy

   or

   Crea una politica di autorizzazioni in linea per il ruolo: PutRolePolicy
3. (Facoltativo) Aggiungi attributi personalizzati al ruolo allegando tag: TagRole

Per ulteriori informazioni, consulta Gestione dei tag sui ruoli IAM (AWS CLI or AWS API).

4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: PutRolePermissionsBoundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

Creare un ruolo per la federazione OpenID Connect (console)

Puoi utilizzare i provider di identità federati OpenID Connect (OIDC) invece di creare AWS Identity and Access Management utenti nel tuo. Account AWS Con un provider di identità (IdP), puoi gestire le tue identità utente all'esterno AWS e concedere a queste identità utente esterne le autorizzazioni per accedere alle risorse del tuo account. AWS Per ulteriori informazioni sulla federazione e, vedere. IdPs Fornitori di identità e federazione in AWS

Prerequisiti per la creazione di un ruolo per OIDC

Prima di poter creare un ruolo per la federazione OIDC, devi completare i seguenti passaggi obbligatori.

Per prepararsi alla creazione di un ruolo per la federazione OIDC

1. Registrati con uno o più servizi che offrono l'identità OIDC federata. Se stai creando un'app che richiede l'accesso alle tue AWS risorse, configura anche l'app con le informazioni del provider. Al momento della registrazione, il gestore fornisce un ID applicazione o destinatario univoco per l'app. Provider diversi utilizzano una terminologia diversa per questo processo. Questa guida utilizza il termineconfigurare per il processo di identificazione dell'applicazione con il provider. È possibile configurare più app con ogni provider o più provider con una sola app. Consulta le informazioni sull'utilizzo degli IdP specificate di seguito:

   - Centro Sviluppatori di Login with Amazon
   - Aggiunta dell'accesso a Facebook a un'app o a un sito Web sul sito degli sviluppatori di Facebook.
   - Utilizzo di OAuth 2.0 per l'accesso (OpenID Connect) sul sito degli sviluppatori di Google.
2. Dopo aver ricevuto le informazioni richieste dall'IdP, crea un IdP in IAM. Per ulteriori informazioni, consulta Creare un provider di identità OpenID Connect (OIDC) in IAM.

> **Important**
>
> Se utilizzi un IdP OIDC di Google, Facebook o Amazon Cognito, non occorre creare un IdP IAM separato nella Console di gestione AWS. Questi provider di identità OIDC sono già integrati AWS e sono disponibili per l'uso. Ignora questa fase e vai alla fase successiva per creare nuovi ruoli utilizzando l'IdP.

3. Prepara le politiche per il ruolo che assumeranno gli IdP-authenticated utenti. Come qualsiasi altro ruolo, anche il ruolo per un'app per dispositivi mobili include due policy. Una è la policy di affidabilità, che specifica chi può assumere il ruolo. L'altra è la policy di autorizzazione, che specifica le operazioni e le risorse AWS a cui l'app per dispositivi mobili può accedere o meno.

   Per il web IdPs, ti consigliamo di utilizzare Amazon Cognito per gestire le identità. In tal caso, si utilizza una policy di attendibilità simile all'esempio seguente.

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": {
           "Effect": "Allow",
           "Principal": {"Federated": "cognito-identity.amazonaws.com"},
           "Action": "sts:AssumeRoleWithWebIdentity",
           "Condition": {
               "StringEquals": {"cognito-identity.amazonaws.com:aud": "us-
   east-2:12345678-abcd-abcd-abcd-123456"},
               "ForAnyValue:StringLike": {"cognito-identity.amazonaws.com:amr":
    "unauthenticated"}
           }
       }
   }
   ```

   Sostituisci `us-east-2:12345678-abcd-abcd-abcd-123456` con l'ID del pool di identità che ti ha assegnato Amazon Cognito.

   Nella configurazione manuale di un IdP OIDC, al momento della creazione della policy di attendibilità occorre utilizzare tre valori che garantiscono che solo l'app in questione possa assumere quel ruolo:

- Per l'elemento `Action`, si utilizza l'operazione `sts:AssumeRoleWithWebIdentity`.
- Per l'elemento `Principal`, usa la stringa `{"Federated":`*`providerUrl/providerArn`*`}`.
  - Per alcuni OIDC comuni IdPs, è un URL. *`providerUrl`* Gli esempi seguenti includono metodi per specificare il principio per alcuni comuni: IdPs

    ```
    "Principal":{"Federated":"cognito-identity.amazonaws.com"}
    ```

    ```
    "Principal":{"Federated":"www.amazon.com"}
    ```

    ```
    "Principal":{"Federated":"graph.facebook.com"}
    ```

    ```
    "Principal":{"Federated":"accounts.google.com"}
    ```

  - Per gli altri gestori OIDC, utilizza il nome della risorsa Amazon (ARN) dell'IdP OIDC creato in Step 2, come nell'esempio seguente:

    ```
    "Principal":{"Federated":"arn:aws:iam::123456789012:oidc-provider/
    server.example.com"}
    ```

- Per l'elemento `Condition`, si utilizza una condizione `StringEquals` per limitare le autorizzazioni. È necessario testare l'ID del pool di identità per Amazon Cognito o l'ID app per altri provider. L'ID del pool di identità dovrebbe corrispondere all'ID app che hai ricevuto durante la configurazione dell'app con l'IdP. Questa corrispondenza tra gli ID assicura che la richiesta provenga dalla tua app.

> **Note**
>
> I ruoli IAM per i pool di identità di Amazon Cognito si affidano al principale del servizio `cognito-identity.amazonaws.com` per assumere il ruolo. I ruoli di questo tipo devono contenere almeno una chiave di condizione per limitare i principali che possono assumere il ruolo.
> Considerazioni aggiuntive si applicano ai pool di identità di Amazon Cognito che assumono ruoli IAM su più account. Le policy di attendibilità di questi ruoli devono accettare il principale del servizio `cognito-identity.amazonaws.com` e devono contenere la chiave di condizione `aud` per limitare l'assunzione di ruoli agli utenti dei pool di identità previsti. Una policy che considera attendibili i pool di identità di Amazon Cognito senza questa condizione comporta il rischio che un utente proveniente da un pool di identità non intenzionale possa assumere il ruolo. Per ulteriori informazioni,

consulta Policy di attendibilità per i ruoli IAM nell'autenticazione di base (classica) nella Guida per gli sviluppatori di Amazon Cognito.

Crea un elemento condizione simile agli esempi seguenti, a seconda dell'IdP in uso:

```
"Condition": {"StringEquals": {"cognito-identity.amazonaws.com:aud":
"us-east:12345678-ffff-ffff-ffff-123456"}}
```

```
"Condition": {"StringEquals": {"www.amazon.com:app_id":
"amzn1.application-oa2-123456"}}
```

```
"Condition": {"StringEquals": {"graph.facebook.com:app_id":
"111222333444555"}}
```

```
"Condition": {"StringEquals": {"accounts.google.com:aud":
"66677788899900pro0"}}
```

Per i provider OIDC, si utilizza l'URL completo del provider di identità OIDC con la chiave di contesto aud, come nell'esempio seguente:

```
"Condition": {"StringEquals": {"server.example.com:aud":
"appid_from_oidc_idp"}}
```

> **Note**
>
> I valori per il principale nella policy di attendibilità per il ruolo sono specifici dell'IdP. Un ruolo per OIDC può specificare solo un principale. Pertanto, se l'app per dispositivi mobili consente agli utenti di effettuare l'accesso da più di un IdP, devi creare un ruolo separato per ogni IdP da supportare. Crea policy di attendibilità separate per ogni IdP.

Se un utente utilizza un'app per dispositivi mobili per accedere da Login with Amazon, si applica la policy di attendibilità di esempio riportata di seguito. Nell'esempio, *amzn1.application-oa2-123456* rappresenta l'ID dell'app che Amazon assegna quando hai configurato l'app utilizzando Login with Amazon.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [{
          "Sid": "RoleForLoginWithAmazon",
          "Effect": "Allow",
          "Principal": {"Federated": "www.amazon.com"},
          "Action": "sts:AssumeRoleWithWebIdentity",
          "Condition": {"StringEquals": {"www.amazon.com:app_id":
 "amzn1.application-oa2-123456"}}
      }]
  }
```

Se un utente utilizza un'app per dispositivi mobili per accedere da Facebook, si applica la policy di attendibilità di esempio riportata di seguito. In questo esempio, *111222333444555* rappresenta l'ID dell'app assegnato da Facebook.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [{
          "Sid": "RoleForFacebook",
          "Effect": "Allow",
          "Principal": {"Federated": "graph.facebook.com"},
          "Action": "sts:AssumeRoleWithWebIdentity",
          "Condition": {"StringEquals": {"graph.facebook.com:app_id":
 "111222333444555"}}
      }]
  }
```

Se un utente utilizza un'app per dispositivi mobili per accedere da Google, si applica la policy di attendibilità di esempio riportata di seguito. In questo esempio, *666777888999000* rappresenta l'ID dell'app assegnato da Google.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [{
          "Sid": "RoleForGoogle",
          "Effect": "Allow",
          "Principal": {"Federated": "accounts.google.com"},
          "Action": "sts:AssumeRoleWithWebIdentity",
          "Condition": {"StringEquals": {"accounts.google.com:aud":
 "666777888999000"}}
      }]
  }
```

Se un utente utilizza un'app per dispositivi mobili per accedere da Amazon Cognito, si applica la policy di attendibilità di esempio riportata di seguito. In questo esempio, *us-east:12345678-ffff-ffff-ffff-123456* rappresenta l'ID del pool di identità assegnato da Amazon Cognito.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [{
          "Sid": "RoleForCognito",
          "Effect": "Allow",
          "Principal": {"Federated": "cognito-identity.amazonaws.com"},
          "Action": "sts:AssumeRoleWithWebIdentity",
          "Condition": {"StringEquals": {"cognito-
identity.amazonaws.com:aud": "us-east:12345678-ffff-ffff-ffff-123456"}}
      }]
  }
```

Creazione di un ruolo per OIDC

Una volta completati i prerequisiti, puoi creare il ruolo in IAM. Per i provider di identità condivisi OpenID Connect (OIDC) riconosciuti (IdPs), IAM richiede una valutazione esplicita di dichiarazioni

specifiche in JSON Web Tokens (JWT), note come controlli dei provider di identità. Per ulteriori informazioni su quali OIDC dispongono dei controlli relativi ai provider di identità, consulta. IdPs Identity-provider controlli per provider OIDC condivisi

Nella procedura seguente viene descritto come creare il ruolo per la federazione OIDC nella Console di gestione AWS. Per creare un ruolo dall' AWS API AWS CLI or, consulta le procedure in. Creazione di un ruolo per un provider di identità di terze parti

> **Important**
>
> Se utilizzi Amazon Cognito, utilizza la console di Amazon Cognito per configurare i ruoli. In caso contrario, usa la console IAM per creare un ruolo per la federazione OIDC.

Per creare un ruolo IAM per la federazione OIDC

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Ruoli, quindi Crea ruolo.
3. Scegli Identità Web come tipo di entità attendibile e seleziona Avanti.
4. Per Identity provider (Gestore dell'identità digitale [IdP]), scegli l'IdP per il ruolo:
   - Se vuoi creare un ruolo per un singolo IdP Web, scegli Login with Amazon, Facebook o Google.

     > **Note**
     >
     > Devi creare un ruolo separato per ogni IdP che intendi supportare.

   - Se vuoi creare un ruolo per uno scenario avanzato per Amazon Cognito, scegli Amazon Cognito.

     > **Note**
     >
     > Devi creare manualmente un ruolo da utilizzare con Amazon Cognito solo quando operi in uno scenario avanzato. In caso contrario, i ruoli possono essere creati da Amazon Cognito. Per ulteriori informazioni su Amazon Cognito, consulta la pagina

Provider di identità esterni di pool di identità (identità federate) nella Guida per gli sviluppatori di Amazon Cognito.

- Se desideri creare un ruolo per GitHub Actions, devi iniziare aggiungendo il provider GitHub OIDC a IAM. Dopo aver aggiunto il provider GitHub OIDC a IAM, scegli token.actions.githubusercontent.com.

> **Note**
>
> Per informazioni su come configurare il provider OIDC di AWS to trust come identità federata, consulta GitHub Docs - Configuring OpenID Connect in Amazon Web Services. GitHub Per informazioni sulle best practice per limitare l'accesso ai ruoli associati all'IAM IdP per, consulta questa pagina. GitHub Configurazione di un ruolo per il provider di GitHub identità OIDC

- Se desideri creare un ruolo per HashiCorp Cloud Platform (HCP) Terraform, devi iniziare aggiungendo il provider OIDC Terraform a IAM. Dopo aver aggiunto il provider OIDC Terraform a IAM, scegli app.terraform.io.

> **Important**
>
> I ruoli IAM per il provider OIDC Terraform HashiCorp Cloud Platform (HCP) devono valutare la chiave di condizione IAM, nella policy di fiducia dei ruoli. `app.terraform.io:sub` Questa chiave di condizione limita le organizzazioni, i progetti, gli spazi di lavoro o le fasi di esecuzione di HCP Terraform in grado di assumere il ruolo. Senza questa chiave condizionale, la tua politica di fiducia garantisce l'accesso al tuo ruolo e alle tue AWS risorse da identità esterne alla tua organizzazione, il che non è in linea con il principio del privilegio minimo.
> Se imposti o modifichi una politica di fiducia dei ruoli per un ruolo associato al provider HCP Terraform OIDC nel tuo AWS account, ma non valuti la chiave di condizione IAM, riceverai un errore. `app.terraform.io:sub` Inoltre, AWS STS negherà le richieste di autorizzazione se la policy di attendibilità del ruolo non valuta questa chiave di condizione.

5. Le informazioni richieste variano in base al provider OIDC scelto.

- Inserisci l'identificatore per l'applicazione. L'etichetta relativa all'identificatore cambia in base al gestore scelto:

- Se vuoi creare un ruolo per Login with Amazon, inserisci l'ID app nella casella Application ID (ID applicazione).
- Se vuoi creare un ruolo per Facebook, inserisci l'ID app nella casella Application ID (ID applicazione).
- Se vuoi creare un ruolo per Google, inserisci il nome del destinatario nella casella Audience (Destinatario).
- Se vuoi creare un ruolo per Amazon Cognito, inserisci l'ID del pool di identità che hai creato per le applicazioni Amazon Cognito nella casella Identity Pool ID (ID pool di identità).

- Se desideri creare un ruolo per GitHub Actions, inserisci i seguenti dettagli:
  - Per Pubblico, scegli `sts.amazonaws.com`.
  - Per GitHub l'organizzazione, inserisci il nome GitHub dell'organizzazione. Il nome GitHub dell'organizzazione è obbligatorio e deve essere alfanumerico, compresi i trattini (-). Non puoi usare caratteri jolly (* e?) nel nome dell' GitHub organizzazione.
  - (Facoltativo) Per il GitHub repository, inserisci il nome del GitHub repository. Se non specifichi un valore, viene utilizzato un carattere jolly (*) per impostazione predefinita.
  - (Facoltativo) Per la GitHub filiale, inserisci il nome della GitHub filiale. Se non specifichi un valore, viene utilizzato un carattere jolly (*) per impostazione predefinita.
- Se desideri creare un ruolo per HashiCorp Cloud Platform (HCP) Terraform, inserisci i seguenti dettagli:
  - Per Pubblico, scegli `aws.workload.identity`.
  - Per Organizzazione, inserisci il nome dell'organizzazione. È possibile specificare un carattere jolly (*) per tutte le organizzazioni.
  - Per Progetto, inserisci il nome del progetto. È possibile specificare un carattere jolly (*) per tutti i progetti.
  - In Spazio di lavoro, immetti un nome per lo spazio di lavoro. È possibile specificare un carattere jolly (*) per tutti gli spazi di lavoro.
  - Per Fase di esecuzione, inserisci il nome della fase di esecuzione. È possibile specificare un carattere jolly (*) per tutte le fasi di esecuzione.

6. (Facoltativo) In Condizione (facoltativo) scegli Aggiungi condizioneper creare condizioni aggiuntive che devono essere soddisfatte prima che gli utenti dell'applicazione possano utilizzare le autorizzazioni concesse dal ruolo. Ad esempio, puoi aggiungere una condizione che conceda l'accesso alle AWS risorse solo per un ID utente IAM specifico. Puoi anche aggiungere

condizioni alla policy di attendibilità dopo la creazione del ruolo. Per ulteriori informazioni, consulta Aggiornamento di una policy di attendibilità del ruolo.

7. Verifica le informazioni su OIDC, quindi seleziona Successivo.
8. IAM include un elenco delle policy AWS gestite e gestite dal cliente nel tuo account. Seleziona la policy delle autorizzazioni da utilizzare o scegli Create policy (Crea policy) per aprire una nuova scheda del browser e creare una nuova policy da zero. Per ulteriori informazioni, consulta Creazione di policy IAM. Una volta creata la policy, chiudere la scheda e tornare alla scheda originale. Seleziona la casella di controllo accanto alle policy di autorizzazione che si desidera abbiano gli utenti OIDC. È anche possibile non selezionare le policy ora e collegarle al ruolo in un secondo momento. Per default, un ruolo non dispone di autorizzazioni.
9. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata.

   Apri la sezione Permissions boundary (Limite delle autorizzazioni) e scegli Use a permissions boundary to control the maximum role permissions (Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo). Selezionare la policy da utilizzare per il limite delle autorizzazioni.
10. Scegli Next (Successivo).
11. In Role name, (Nome ruolo), inserisci un nome. I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS. Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare ruoli denominati sia **`PRODROLE`** sia **`prodrole`**. Poiché altre AWS risorse potrebbero fare riferimento al ruolo, non puoi modificare il nome del ruolo dopo averlo creato.
12. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
13. Per modificare i casi d'uso e le autorizzazioni per il ruolo, scegli Edit (Modifica) nelle sezioni Step 1: Select trusted entities (Fase 1: seleziona le entità attendibili) o Step 2: Add permissions (Fase 2: aggiungi autorizzazioni).
14. (Facoltativo) Per aggiungere metadati al ruolo, collegare i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.
15. Rivedere il ruolo e scegliere Crea ruolo.

Configurazione di un ruolo per il provider di GitHub identità OIDC

Se lo utilizzi GitHub come provider di identità (IdP) OpenID Connect (OIDC), la procedura migliore è limitare le entità che possono assumere il ruolo associato all'IdP IAM. Quando includi una dichiarazione di condizione nella policy di fiducia, puoi limitare il ruolo a

un' GitHuborganizzazione, un repository o una filiale specifici. Puoi usare la chiave di condizione `token.actions.githubusercontent.com:sub` con operatori di condizione delle stringhe per limitare l'accesso. Si consiglia di limitare la condizione a un insieme specifico di repository o filiali all'interno dell'organizzazione GitHub . Per informazioni su come configurare l'OIDC AWS GitHub di trust come identità federata, consulta GitHub Docs - Configuring OpenID Connect in Amazon Web Services.

Se utilizzi GitHub gli ambienti nei flussi di lavoro in azione o nelle politiche OIDC, ti consigliamo vivamente di aggiungere regole di protezione all'ambiente per una maggiore sicurezza. Utilizza i rami e i tag di implementazione per limitare i rami e i tag che possono essere implementati nell'ambiente. Per ulteriori informazioni sulla configurazione degli ambienti con regole di protezione, consulta i rami e i tag di distribuzione nell' GitHubarticolo Using environment for deployment.

Quando GitHub l'IdP OIDC è il Principal affidabile per il tuo ruolo, IAM controlla la condizione della politica di attendibilità del ruolo per verificare che la chiave di condizione `token.actions.githubusercontent.com:sub` sia presente e che il suo valore non sia solo un carattere jolly (* e?) o null. IAM esegue questo controllo quando la policy di attendibilità viene creata o aggiornata. Se la chiave di condizione `token.actions.githubusercontent.com:sub` non è presente o il valore della chiave non soddisfa i criteri di valore indicati, la richiesta avrà esito negativo e restituirà un errore.

> **Important**
>
> Se non limiti la chiave condizionale a un'organizzazione o `token.actions.githubusercontent.com:sub` a un repository specifico, GitHub le azioni provenienti da organizzazioni o repository al di fuori del tuo controllo possono assumere i ruoli associati all' GitHub IdP IAM nel tuo account. AWS

Il seguente esempio di politica di fiducia limita l'accesso all' GitHub organizzazione, al repository e alla filiale definiti. Il `token.actions.githubusercontent.com:sub` valore della chiave di condizione nell'esempio seguente è il formato predefinito del valore dell'oggetto documentato da. GitHub

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::012345678910:oidc-provider/
token.actions.githubusercontent.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "token.actions.githubusercontent.com:aud": "sts.amazonaws.com",
          "token.actions.githubusercontent.com:sub":
 "repo:GitHubOrg/GitHubRepo:ref:refs/heads/GitHubBranch"
        }
      }
    }
  ]
}
```

La seguente condizione di esempio limita l'accesso all' GitHub organizzazione e al repository definiti, ma concede l'accesso a qualsiasi filiale all'interno del repository.

```
"Condition": {
  "StringEquals": {
          "token.actions.githubusercontent.com:aud": "sts.amazonaws.com"
  },
  "StringLike": {
    "token.actions.githubusercontent.com:sub": "repo:GitHubOrg/GitHubRepo:*"
  }
}
```

La seguente condizione di esempio limita l'accesso a qualsiasi repository o filiale all'interno dell'organizzazione definita. GitHub Si consiglia di limitare la chiave condizionale `token.actions.githubusercontent.com:sub` a un valore specifico che limiti l'accesso alle GitHub azioni dall'interno GitHub dell'organizzazione.

```
"Condition": {
  "StringEquals": {
          "token.actions.githubusercontent.com:aud": "sts.amazonaws.com"
  },
  "StringLike": {
```

```
    "token.actions.githubusercontent.com:sub": "repo:GitHubOrg/*"
  }
}
```

Per ulteriori informazioni sulle chiavi di federazione OIDC disponibili per i controlli delle condizioni nelle policy, consulta Chiavi disponibili per AWS Federazione OIDC.

Creare un ruolo per una federazione SAML 2.0 (console)

Puoi utilizzare la federazione SAML 2.0 invece di creare utenti IAM nel tuo Account AWS. Con un provider di identità (IdP), puoi gestire le tue identità utente all'esterno AWS e concedere a queste identità utente esterne le autorizzazioni per accedere AWS alle risorse del tuo account. Per ulteriori informazioni sulla federazione e sui provider di identità, consultare Fornitori di identità e federazione in AWS.

> **Note**
>
> Per migliorare la resilienza della federazione, ti consigliamo di configurare l'IdP e la federazione AWS per supportare più endpoint di accesso SAML. Per informazioni dettagliate, consulta l'articolo del AWS Security Blog How to use regional SAML endpoint for failover.

Prerequisiti per la creazione di un ruolo per SAML

Prima di creare un ruolo per la federazione SAML 2.0, devi completare i seguenti passaggi obbligatori.

Preparazione per la creazione di un ruolo per la federazione SAML 2.0

1. Prima di creare un ruolo per la SAML-based federazione, devi creare un provider SAML in IAM. Per ulteriori informazioni, consulta Creare un provider di identità SAML in IAM.
2. Prepara le policy per il ruolo che verrà assunto dagli utenti autenticati con SAML 2.0. Come qualsiasi altro ruolo, anche i ruoli per la federazione SAML includono due policy. Una è la policy di attendibilità del ruolo, che specifica chi può assumere il ruolo. L'altra è la politica delle autorizzazioni IAM che specifica le AWS azioni e le risorse a cui è consentito o negato l'accesso al principale federato SAML.

   Al momento della creazione della policy di attendibilità per il ruolo, devi utilizzare tre valori che garantiscono che solo la tua applicazione possa assumere il ruolo:

- Per l'elemento `Action`, si utilizza l'operazione `sts:AssumeRoleWithSAML`.
- Per l'elemento `Principal`, usa la stringa `{"Federated":`*`ARNofIdentityProvider`*`}`. Sostituire *`ARNofIdentityProvider`* con l'ARN del provider di identità SAML creato in Step 1.
- Per l'elemento `Condition`, utilizza una condizione `StringEquals` per verificare che l'attributo `saml:aud` della risposta SAML corrisponda all'URL visualizzato dal browser quando si accede alla console. L'URL dell'endpoint di accesso corrisponde all'attributo destinatario SAML del provider di identità. Puoi includere gli URL di accesso all'interno di determinate aree geografiche. AWS consiglia di utilizzare gli endpoint regionali anziché gli endpoint globali per migliorare la resilienza della federazione. Per un elenco dei *`region-code`* valori possibili, consulta la colonna Regione negli AWS Sign-In endpoint degli endpoint.

  Se è richiesta la crittografia SAML, l'URL di accesso deve includere l'identificatore univoco che AWS assegna al provider SAML. Puoi visualizzare l'identificatore univoco selezionando il provider di identità nella console IAM per visualizzare la pagina dei dettagli.

  `https://`*`region-code`*`.signin.aws.amazon.com/saml/acs/`*`IdP-ID`*

L'esempio seguente mostra una policy di affidabilità concepita per un utente federato SAML:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRoleWithSAML",
        "Principal": {
            "Federated": "arn:aws:iam::111122223333:saml-provider/PROVIDER-
NAME"
        },
        "Condition": {
            "StringEquals": {
                "SAML:aud": "https://region-code.signin.aws.amazon.com/saml"
            }
        }
    }
}
```

Sostituisci l'ARN principale con l'ARN effettivo del provider SAML, creato in IAM. L'ARN include l'ID account e il nome del provider.

Creazione di un ruolo per SAML

Dopo aver completato i passaggi dei prerequisiti, è possibile creare il ruolo per la federazione. SAML-based

Per creare un ruolo per la SAML-based federazione

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegli Ruoli e quindi Crea ruolo.
3. Selezionare il tipo di ruolo SAML 2.0 federation (Federazione SAML 2.0).
4. In Select a SAML provider (Seleziona un gestore dell'identità digitale SAML), scegli il gestore per il ruolo.
5. Selezionare il metodo di livello di accesso SAML 2.0.

   - Scegli Consenti solo l'accesso programmatico per creare un ruolo che può essere assunto a livello di programmazione dall' AWS API o. AWS CLI
   - Scegli Consenti Console di gestione AWS accesso programmatico e programmatico per creare un ruolo che può essere assunto a livello di programmazione e da. Console di gestione AWS

   I due comandi sono simili, ma il ruolo che può essere assunto anche tramite console include una policy di affidabilità con una condizione particolare. Tale condizione verifica in modo esplicito che il destinatario SAML (attributo `SAML:aud`) sia impostato sull'endpoint di accesso di AWS per il tuo provider SAML.
6. La procedura per definire gli attributi varia a seconda del tipo di accesso.
   - Se si sta creando un ruolo per l'accesso programmatico, scegliere un attributo dall'elenco Attributo. Dopodiché, nella casella Value (Valore), inserisci un valore da includere nel ruolo. In questo modo, l'accesso al ruolo viene limitato agli utenti dal provider di identità la cui risposta di autenticazione SAML (asserzione) includa gli attributi specificati. Per fare in modo che il ruolo sia limitato a un sottoinsieme di utenti all'interno dell'organizzazione, specificare almeno un attributo.

- Se stai creando un ruolo di tipo programmatico e di Console di gestione AWS accesso, la sezione Sign-in Endpoints definisce l'URL visualizzato dal browser quando accedi alla console. Questo endpoint è l'attributo destinatario SAML del tuo provider di identità, che corrisponde alla chiave di contesto `saml:aud`. Per ulteriori informazioni, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

  a. Scegli endpoint o endpoint regionali. Non-regional Ti consigliamo di utilizzare più endpoint di accesso SAML regionali per migliorare la resilienza della federazione.

  b. Per le regioni, scegli le regioni supportate dal tuo provider SAML per l'accesso. AWS

  c. Sign-in Affinché gli URL includano identificatori univoci, seleziona se gli endpoint di accesso includono gli identificatori univoci assegnati al tuo provider di identità AWS SAML. Questa opzione è necessaria per le asserzioni SAML crittografate. Per ulteriori informazioni, consulta Federazione SAML 2.0.

7. Per aggiungere ulteriori condizioni relative agli attributi alla policy di attendibilità, scegli Condition (optional) (Condizione [facoltativo]), seleziona la condizione aggiuntiva e specifica un valore.

> **Note**
>
> L'elenco include gli attributi SAML più comunemente utilizzati. IAM supporta attributi aggiuntivi che puoi usare per creare condizioni. Per un elenco degli attributi supportati, consulta Chiavi disponibili per la federazione SAML. Se si necessita di una condizione per un attributo SAML supportato che non è nell'elenco, è possibile aggiungere tale condizione manualmente. A tale scopo, modificare la policy di attendibilità dopo aver creato il ruolo.

8. Verifica le informazioni di attendibilità di SAML 2.0, quindi scegli Next (Successivo).
9. IAM include un elenco delle politiche AWS gestite e gestite dal cliente nel tuo account. Seleziona la policy delle autorizzazioni da utilizzare o scegli Create policy (Crea policy) per aprire una nuova scheda del browser e creare una nuova policy da zero. Per ulteriori informazioni, consulta Creazione di policy IAM. Una volta creata la policy, chiudere la scheda e tornare alla scheda originale. Seleziona la casella di controllo accanto alle policy di autorizzazione che desideri concedere agli utenti federati SAML. È anche possibile non selezionare le policy ora e collegarle al ruolo in un secondo momento. Per default, un ruolo non dispone di autorizzazioni.
10. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata.

Apri la sezione Permissions boundary (Limite delle autorizzazioni) e scegli Use a permissions boundary to control the maximum role permissions (Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo). Selezionare la policy da utilizzare per il limite delle autorizzazioni.

11. Scegli Next (Successivo).
12. Scegli Prossimo: Rivedi.
13. In Role name, (Nome ruolo), inserisci un nome. I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS. Non si distinguono per maiuscole e minuscole. Ad esempio, non è possibile creare ruoli denominati sia **`PRODROLE`** che **`prodrole`**. Poiché altre AWS risorse potrebbero fare riferimento al ruolo, non è possibile modificare il nome del ruolo dopo che è stato creato.
14. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
15. Scegli Edit (Modifica) nelle sezioni Step 1: Select trusted entities (Fase 1: seleziona le entità attendibili) o Step 2: Add permissions (Fase 2: aggiungi autorizzazioni) per modificare i casi d'uso e le autorizzazioni per il ruolo.
16. (Facoltativo) Aggiungere metadati al ruolo collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.
17. Rivedere il ruolo e scegliere Crea ruolo.

Una volta creato il ruolo, è possibile completare la relazione di trust SAML configurando il software provider di identità con informazioni su AWS. Queste informazioni includono i ruoli che devono utilizzare gli utenti federati SAML. Tale operazione viene definita configurazione della relazione di trust fra IdP e AWS. Per ulteriori informazioni, consulta Configurare il provider di identità SAML 2.0 con una relazione di attendibilità della parte affidabile e aggiunta di attestazioni.

Creare un ruolo utilizzando policy di attendibilità personalizzate

Puoi creare una politica di fiducia personalizzata per delegare l'accesso e consentire ad altri di eseguire azioni nel tuo Account AWS. Per ulteriori informazioni, consulta Creazione di policy IAM.

Per informazioni su come utilizzare i ruoli per delegare le autorizzazioni, consultare Termini e concetti dei ruoli.

Creazione di un ruolo IAM utilizzando una policy di attendibilità personalizzata (console)

Puoi utilizzare il Console di gestione AWS per creare un ruolo che un utente IAM può assumere. Ad esempio, supponiamo che l'organizzazione disponga di più Account AWS elementi per isolare un ambiente di sviluppo da un ambiente di produzione. Per informazioni di alto livello sulla creazione di un ruolo che consenta agli utenti nell'account di sviluppo di accedere alle risorse nell'account di produzione, consulta la sezione Esempio di uno scenario in cui si utilizzano account di sviluppo e produzione separati.

Creare un ruolo utilizzando una policy di attendibilità personalizzata (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, selezionare Ruoli e Crea ruolo.
3. Scegli il tipo di ruolo Custom trust policy (Policy di attendibilità personalizzata).
4. Nella sezione Custom trust policy (Policy di attendibilità personalizzata), inserisci o incolla la policy di attendibilità personalizzata per il ruolo. Per ulteriori informazioni, consulta Creazione di policy IAM.
5. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).
6. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una funzionalità avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.

   Apri la sezione Permissions boundary (Limite delle autorizzazioni) e scegli Use a permissions boundary to control the maximum role permissions (Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo). IAM include un elenco delle politiche AWS gestite e gestite dai clienti nel tuo account. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
7. Scegli Next (Successivo).
8. Il grado di personalizzazione per Nome ruolo viene definito dal servizio. Se il servizio definisce il nome del ruolo, questa opzione non può essere modificata. In altri casi, il servizio può definire un prefisso per il ruolo e consentire all'utente di aggiungere un suffisso opzionale. Alcuni servizi consentono di specificare l'intero nome del ruolo.

   Se possibile, inserisci un nome del ruolo o un suffisso del nome del ruolo. I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS. Non si distinguono per caso. Ad esempio, non è possibile creare ruoli denominati sia **`PRODROLE`** che **`prodrole`**. Poiché altre AWS risorse

potrebbero fare riferimento al ruolo, non è possibile modificare il nome del ruolo dopo che è stato creato.

9. (Facoltativo) In Descrizione, inserisci una descrizione per il nuovo ruolo.
10. (Facoltativo) Scegli Modifica nelle sezioni Fase 1: seleziona le entità attendibili o Fase 2: aggiungi autorizzazioni per modificare la policy personalizzata e le autorizzazioni per il ruolo.
11. (Facoltativo) Aggiungere metadati al ruolo collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.
12. Rivedere il ruolo e scegliere Crea ruolo.

Esempi di policy per la delega dell'accesso

Gli esempi seguenti mostrano come è possibile consentire o concedere Account AWS l'accesso alle risorse di un altro Account AWS. Per ulteriori informazioni su come creare una policy IAM utilizzando questi documenti di policy JSON, consulta the section called "Creazione di policy utilizzando l'editor JSON".

Argomenti

- Utilizzo dei ruoli per delegare l'accesso alle risorse di un altro Account AWS risorse
- Utilizzo di una policy per delegare l'accesso ai servizi
- Utilizzo di una policy basata sulle risorse per delegare l'accesso a un bucket Amazon S3 in un altro account
- Utilizzo di una policy basata sulle risorse per delegare l'accesso a una coda Amazon SQS in un altro account
- Impossibile delegare l'accesso quando l'accesso all'account è rifiutato

Utilizzo dei ruoli per delegare l'accesso alle risorse di un altro Account AWS risorse

Per un tutorial che mostra come utilizzare i ruoli IAM per concedere agli utenti di un account l'accesso alle AWS risorse che si trovano in un altro account, consultaTutorial IAM: Delega l'accesso attraverso AWS account che utilizzano ruoli IAM.

> **Important**
>
> È possibile includere l'ARN per un ruolo o utente specifico nell'elemento `Principal` di una policy di affidabilità del ruolo. Quando salvi la policy, AWS trasforma l'ARN in un ID principale

univoco. Ciò aiuta a mitigare il rischio che qualcuno aumenti i propri privilegi rimuovendo e ricreando il ruolo o l'utente. Questa ID nella console non è normalmente presente, in quanto c'è anche una trasformazione inversa verso il nome ARN quando la policy di affidabilità viene visualizzata. Tuttavia, se si elimina il ruolo o l'utente, la relazione viene interrotta. La policy non è più applicabile, anche se si ricrea l'utente o il ruolo perché non corrisponde all'ID principale archiviato nella policy di attendibilità. Quando ciò accade, l'ID principale viene visualizzato nella console perché non è più AWS possibile ricondurlo a un ARN. Il risultato è che se si elimina e si ricrea un utente o un ruolo referenziato in un elemento `Principal` della policy di attendibilità, è necessario modificare il ruolo per sostituire il nome ARN. L'utente o il ruolo viene trasformato nel nuovo ID principale quando si salva la policy.

Utilizzo di una policy per delegare l'accesso ai servizi

L'esempio seguente mostra una policy che può essere collegata a un ruolo. La policy consente a due servizi, Amazon EMR e AWS Data Pipeline, di assumere il ruolo. I servizi possono eseguire qualsiasi attività concesse da una policy di autorizzazioni assegnata al ruolo (non visualizzato). Per specificare più principali del servizio, non si specificano due elementi `Service`, è possibile averne solo uno. Utilizzare invece una serie di principali del servizio come il valore di un elemento singolo `Service`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "elasticmapreduce.amazonaws.com",
          "datapipeline.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Utilizzo di una policy basata sulle risorse per delegare l'accesso a un bucket Amazon S3 in un altro account

In questo esempio, l'account A utilizza una policy basata sulle risorse (una policy del bucket Amazon S3) per concedere all'account B l'accesso completo al bucket S3 dell'account A. A questo punto, l'account B crea una policy utente IAM per delegare tale accesso al bucket dell'account A a uno degli utenti nell'account B.

La policy del bucket S3 nell'account A potrebbe essere simile alla policy seguente. In questo esempio, il bucket S3 dell'account A è denominato amzn-s3-demo-bucket e il numero dell'account B è 111122223333. Non specifica alcun utente o gruppo nell'account B, ma solo l'account stesso.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Sid": "AccountBAccess1",
    "Effect": "Allow",
    "Principal": {"AWS": "111122223333"},
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket",
      "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
  }
}
```

In alternativa, l'account A può utilizzare le liste di controllo accessi (ACL) di Amazon S3 per concedere l'accesso all'account B a un bucket S3 o a un singolo oggetto all'interno di un bucket. In questo caso, l'unica cosa che cambia è il modo in cui l'account A concede l'accesso all'account B. L'account B utilizza ancora una policy per delegare l'accesso a un gruppo IAM nell'account B, come descritto nella parte successiva di questo esempio. Per ulteriori informazioni sul controllo dell'accesso su bucket e oggetti S3, consulta Access Control nella Amazon Simple Storage Service User Guide.

L'amministratore dell'account B potrebbe creare le seguenti policy di esempio. La policy permette l'accesso in lettura a un gruppo o un utente nell'account B. La policy precedente concede l'accesso all'account B. Tuttavia, i singoli gruppi e gli utenti dell'account B non possono accedere alla risorsa finché una policy utente o di gruppo non concede esplicitamente le autorizzazioni alla risorsa. Le

autorizzazioni in questa policy possono essere solo un subset di quelli nella precedente policy multiaccount. L'account B non può concedere più autorizzazioni ai propri gruppi e utenti rispetto a quanti concessi dall'account A all'account B nella prima policy. In questa policy, l'elemento `Action` è esplicitamente definito per permettere solo operazioni `List` e l'elemento `Resource` di questa policy corrisponde all'elemento `Resource` per la policy del bucket implementata dall'account A.

Per implementare questa policy, l'account B utilizza IAM per collegarla all'utente (o al gruppo) appropriato nell'account B.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:List*",
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket",
      "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
  }
}
```

Utilizzo di una policy basata sulle risorse per delegare l'accesso a una coda Amazon SQS in un altro account

Nell'esempio seguente, l'account A ha una coda Amazon SQS che utilizza una policy basata sulla risorsa collegata alla coda per concedere l'accesso in coda all'account B. Quindi, l'account B utilizza una policy di gruppo IAM per delegare l'accesso a un gruppo nell'account B.

La seguente policy di coda di esempio fornisce all'account B l'autorizzazione per eseguire le operazioni `SendMessage` e `ReceiveMessage` sulla coda dell'account A denominata queue1, ma solo tra mezzogiorno e le 15:00 del 30 novembre 2014. Il numero dell'account B è 1111-2222-3333. L'account A usa Amazon SQS per implementare questa policy.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"AWS": "111122223333"},
    "Action": [
      "sqs:SendMessage",
      "sqs:ReceiveMessage"
    ],
    "Resource": ["arn:aws:sqs:*:123456789012:queue1"],
    "Condition": {
      "DateGreaterThan": {"aws:CurrentTime": "2014-11-30T12:00Z"},
      "DateLessThan": {"aws:CurrentTime": "2014-11-30T15:00Z"}
    }
  }
}
```

La policy dell'account B per la delega dell'accesso a un gruppo nell'account B potrebbe essere simile all'esempio seguente. L'account B usa IAM per collegare questa policy a un gruppo (o utente).

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sqs:*",
    "Resource": "arn:aws:sqs:*:123456789012:queue1"
  }
}
```

Nell'esempio della policy utente IAM precedente, l'account B utilizza un carattere jolly per concedere all'utente l'accesso a tutte le operazioni Amazon SQS per la coda dell'account A. Tuttavia, l'account B può delegare l'accesso solo nella misura in cui all'account B è stato concesso l'accesso. Il gruppo dell'account B con la seconda policy può accedere alla coda solo tra mezzogiorno e le 15:00 del 30 novembre 2014. L'utente può eseguire solo le operazioni `SendMessage` e `ReceiveMessage`, come definito nella policy della coda Amazon SQS dell'account A.

Impossibile delegare l'accesso quando l'accesso all'account è rifiutato

Account AWS Non è possibile delegare l'accesso alle risorse di un altro account se l'altro account ha negato esplicitamente l'accesso all'account principale dell'utente. Il rifiuto si propaga agli utenti di tale account indipendentemente dal fatto che gli utenti dispongano di policy esistenti che garantiscono loro l'accesso.

Ad esempio, l'account A scrive una policy bucket per il bucket S3 dell'account A che rifiuta esplicitamente l'accesso all'account B per il bucket dell'account A. Tuttavia, l'account B scrive una policy utente IAM che concede a un utente dell'account B l'accesso al bucket dell'account A. Il rifiuto esplicito applicato al bucket S3 dell'account A si propaga agli utenti dell'account B e sostituisce la policy dell'utente IAM che concede l'accesso all'utente dell'account B. Per informazioni dettagliate su come vengono valutate le autorizzazioni, consulta Logica di valutazione delle policy.

La policy del bucket dell'account A potrebbe essere simile alla policy seguente. In questo esempio, il bucket S3 dell'account A è denominato amzn-s3-demo-bucket e il numero dell'account B è 1111-2222-3333. L'account A usa Amazon S3 per implementare questa policy.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Sid": "AccountBDeny",
    "Effect": "Deny",
    "Principal": {"AWS": "111122223333"},
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
  }
}
```

Questo rifiuto esplicito sostituisce qualsiasi policy dell'account B che fornisce l'autorizzazione per accedere al bucket S3 nell'account A.

Crea ruoli automaticamente con il gestore dei ruoli

Molti AWS servizi dipendono dai ruoli IAM per eseguire azioni per tuo conto. IAM role manager è un'impostazione opzionale dell'account che fornisce automaticamente tali ruoli, quindi non devi

configurarli tu stesso. Il gestore dei ruoli è progettato per aiutarti a iniziare a creare AWS servizi senza una configurazione manuale aggiuntiva di IAM.

Il gestore dei ruoli è disponibile in alcune console di AWS servizio, elencate in questa the section called "AWS console di servizio che supportano il gestore dei ruoli" pagina.

Come funziona il gestore dei ruoli

Quando il gestore dei ruoli è abilitato, fornisce i ruoli appropriati al servizio e alla risorsa come parte del flusso di lavoro della console pertinente. Le console di servizio supportate indicheranno quando è disponibile il gestore dei ruoli. Non è richiesta alcuna azione aggiuntiva per accettare il ruolo fornito dal gestore dei ruoli. Ad esempio, quando crei una funzione AWS Lambda nella console, Lambda utilizza il gestore dei ruoli per creare il ruolo di esecuzione della funzione per impostazione predefinita. Per utilizzare un ruolo diverso, scegli un ruolo esistente o creane uno nuovo manualmente.

Il gestore dei ruoli funziona applicando un modello di ruolo, una definizione della politica di fiducia di un ruolo e delle autorizzazioni per un servizio e un caso d'uso. Per ulteriori informazioni sul modello utilizzato da ciascun servizio, consultaPanoramica dei modelli di ruolo. Un modello per un'attività definita concede solo le azioni necessarie all'attività. Un modello per lavori a tempo indeterminato, ad esempio l'esecuzione di codice personalizzato, garantisce un ampio accesso alle AWS API in modo da poter creare senza dover concedere personalmente l'accesso a ciascun servizio.

I ruoli IAM creati dal gestore dei ruoli si comportano come qualsiasi altro ruolo che crei. Puoi visualizzarli, modificarli o eliminarli. Ogni ruolo registra anche il modello di ruolo da cui proviene, in modo da poter identificare i ruoli creati dal gestore del ruolo. AWS CloudTrail registra quando il gestore dei ruoli crea i ruoli in modo da poter controllare quando e come ogni ruolo è stato creato. La creazione del ruolo del responsabile del ruolo viene visualizzata CloudTrail come AcquireRole evento.

Prerequisiti

L'abilitazione del gestore dei ruoli richiede le `iam:CreateServiceLinkedRole` autorizzazioni `iam:PutAccountProperties` and. La disabilitazione del gestore dei ruoli richiede l'`iam:PutAccountPropertiesautorizzazione`. La visualizzazione dell'impostazione corrente richiede l'`iam:GetAccountPropertiesautorizzazione`. La politica AWS gestita `IAMFullAccess` include tutte queste autorizzazioni. Per tutte le autorizzazioni richieste dal gestore dei ruoli, vedere. Gestire l'accesso al role manager

Come abilitare e disabilitare il gestore dei ruoli

È possibile abilitare o disabilitare il gestore dei ruoli utilizzando l' AWS API Console di gestione AWS AWS CLI, o. Il gestore dei ruoli è controllato dalla proprietà a livello di account`RoleManager/Enabled`, impostata tramite l'`PutAccountProperties`API. Un valore di `true` abilita il gestore dei ruoli; `false` lo disabilita.

> **Note**
>
> Per la maggior parte AWS degli account, il gestore dei ruoli è disabilitato per impostazione predefinita. Tuttavia, se hai creato il tuo account utilizzando la nostra nuova AWS esperienza, il gestore dei ruoli è abilitato per impostazione predefinita e non può essere disabilitato finché non attivi le funzionalità avanzate. Per ulteriori informazioni, consulta Attivare AWS le funzionalità avanzate.

Per abilitare o disabilitare il gestore dei ruoli (console)

Il gestore dei ruoli può essere abilitato o disabilitato nella pagina delle impostazioni dell'account nella console IAM. Quella pagina mostra anche se il gestore dei ruoli è abilitato per l'account.

1. Accedi Console di gestione AWS e apri la console IAM.
2. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
3. Nella sezione del gestore dei ruoli, scegli Abilita o Disabilita.

I ruoli creati dal gestore dei ruoli non vengono eliminati quando si disabilita il gestore dei ruoli.

Per abilitare o disabilitare il gestore dei ruoli (AWS CLI)

Imposta la `RoleManager/Enabled` proprietà `true` su per abilitare o `false` disabilitare il gestore dei ruoli.

- Per abilitare il gestore dei ruoli, eseguiput-account-properties:

```
aws iam put-account-properties --properties RoleManager/Enabled=true
```

- Per disabilitare il gestore dei ruoli, eseguiput-account-properties:

```
aws iam put-account-properties --properties RoleManager/Enabled=false
```

- Per vedere se il gestore dei ruoli è abilitato per l'account, eseguiget-account-properties:

```
aws iam get-account-properties
```

Per abilitare o disabilitare il gestore dei ruoli (AWS API)

- Per abilitare o disabilitare il gestore dei ruoli, chiama PutAccountPropertiescon la RoleManager/Enabled proprietà impostata su true ofalse.
- Per visualizzare l'impostazione corrente, chiama GetAccountProperties.

Per ulteriori informazioni, consulta PutAccountPropertiese GetAccountPropertiesconsulta lo IAM API Reference.

AWS console di servizio che supportano il gestore dei ruoli

Il gestore dei ruoli supporta la creazione di ruoli per le seguenti console AWS di servizio:

- AWS Backup
- AWS CloudFormation
- Amazon CloudWatch
- Amazon EKS
- AWS Elastic Beanstalk
- Amazon EventBridge
- AWS Glue
- AWS Lambda
- Amazon RDS
- Amazon SageMaker Unified Studio
- AWS Secrets Manager
- AWS Step Functions

AWS continuerà ad aggiungere il supporto del servizio nel tempo. Per ulteriori informazioni sul modello di ruolo utilizzato da ciascun servizio, consultaPanoramica dei modelli di ruolo.

Informazioni correlate

- Gestire l'accesso al role manager
- Applica le autorizzazioni con privilegi minimi a un ruolo creato automaticamente
- Panoramica dei modelli di ruolo

Gestire l'accesso al role manager

È possibile controllare l'accesso al role manager a due livelli. Le autorizzazioni IAM determinano chi in un account può abilitare il role manager e chi può utilizzare i ruoli che fornisce. In un'organizzazione, le politiche di controllo dei servizi (SCP) determinano cosa possono fare gli account dei membri.

Autorizzazioni per il responsabile dei ruoli

Per abilitare o disabilitare il role manager, è necessaria l'`iam:PutAccountProperties`autorizzazione. La policy AWS gestita lo `IAMFullAccess` include.

Per utilizzare il role manager, un utente ha bisogno delle autorizzazioni IAM per le azioni che il role manager esegue per suo conto. Il gestore dei ruoli fornisce i ruoli tramite l'`AcquireRole`API, che valuta ogni azione IAM sottostante rispetto alle autorizzazioni dell'utente:

- Se nell'account esiste già un ruolo che corrisponde al modello, `AcquireRole` restituisce quel ruolo. Richiede l'autorizzazione `iam:GetRole`.
- Se non esiste un ruolo corrispondente, ne `AcquireRole` crea uno dal modello. Ciò richiede l'`iam:CreateRole`autorizzazione, più le autorizzazioni per ciò che il modello definisce: `iam:PutRolePolicy` se il modello aggiunge politiche in linea e `iam:AttachRolePolicy` se il modello allega politiche gestite.

Un utente con accesso IAM completo dispone di tutte le autorizzazioni richieste dal role manager. Per vedere le azioni esatte richieste da un modello specifico, recupera il modello `GetRoleTemplateVersion` e rivedi le sue politiche.

Quando il role manager crea un ruolo, AWS CloudTrail registra la creazione come un singolo `AcquireRole` evento. L'evento mostra chi ha chiamato l'operazione, il modello di ruolo e i valori dei parametri utilizzati e il ruolo che è stato creato.

L'esempio seguente mostra che il role manager ha chiamato l'`AcquireRole`operazione per conto di un utente che ha assunto il ruolo`Admin`, utilizzando il modello di `PowerUserRoleTemplate` ruolo per creare il ruolo`PowerUserRole`.


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA123EXAMPLE:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
        "accountId": "123456789012",
        "accessKeyId": "ASIAWZYMLEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2026-08-11T18:18:50Z",
                "mfaAuthenticated": "false"
            }
        },
        "onBehalfOf": {
            "userId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
        }
    },
    "eventTime": "2026-08-11T18:23:00Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "AcquireRole",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/151.0.0.0 Safari/537.36",
    "requestParameters": {
        "templateArn": "arn:aws:iam::aws:role-template/iam.amazonaws.com/
PowerUserRoleTemplate:1",
        "templateMinorVersion": 0,
```

```
        "replacementValues": {
            "AWSServiceName": {
                "values": [
                    "iam.amazonaws.com"
                ]
            }
        }
    },
    "responseElements": {
        "role": {
            "path": "/",
            "roleName": "PowerUserRole",
            "roleId": "AROA987EXAMPLE",
            "arn": "arn:aws:iam::123456789012:role/PowerUserRole",
            "createDate": "2026-08-11T18:23:00Z",
            "assumeRolePolicyDocument": "%7B%22Version%22:%222012-10-17%22,
%22Statement%22:%5B%7B%22Effect%22:%22Allow%22,%22Principal%22:%7B%22Service%22:
%22iam.amazonaws.com%22%7D,%22Action%22:%22sts:AssumeRole%22%7D%5D%7D"
        }
    },
    "requestID": "fde3dac0-73ac-491c-876b-EXAMPLE89c57",
    "eventID": "c93332cf-1429-4aee-bcaa-EXAMPLE3790d",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "iam.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```


Gestore dei ruoli di controllo in un'organizzazione

Puoi controllare il responsabile dei ruoli in un'organizzazione con una policy di controllo dei servizi (SCP) di AWS Organizations. Gli esempi seguenti mostrano come bloccare l'attivazione del gestore dei ruoli, bloccare la creazione di ruoli o bloccare solo la creazione di ruoli basata su modelli.

Questi esempi utilizzano la strategia dell'elenco negato, il che significa che è necessaria `FullAWSAccess` anche un'altra politica che consenta le azioni associate alle entità

dell'organizzazione. Devi comunque concedere le autorizzazioni ai tuoi dirigenti con politiche basate sull'identità.

Per impedire agli account dei membri di abilitare il role manager tramite la console o l'API, nega l'autorizzazione. `iam:PutAccountProperties` Consigliamo questa politica quando crei nuovi account direttamente tramite Organizations.

```
{
    "Sid": "DenyRoleManagerEnablement",
    "Effect": "Deny",
    "Action": "iam:PutAccountProperties",
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "iam:AccountPropertyNamespaces": "RoleManager"
        }
    }
}
```

Blocca l'accesso per utilizzare il role manager

Per bloccare l'uso del role manager, usa la chiave di `iam:RoleTemplateARN` contesto, presente nelle azioni IAM che creano un ruolo da un modello. La seguente policy nega tutte le azioni IAM che hanno un valore non nullo per la `iam:RoleTemplateARN` chiave di contesto. Questa policy non impedisce la creazione di tutti i ruoli: blocca solo la creazione di ruoli tramite il role manager e la creazione di ruoli basata su modelli tramite l'SDK.

```
{
    "Sid": "DenyTemplatedRoleCreation",
    "Effect": "Deny",
    "Action": "iam:*",
    "Resource": "*",
    "Condition": {
        "Null": {
            "iam:RoleTemplateARN": false
        }
    }
}
```

Blocca la creazione di ruoli tranne tramite il gestore dei ruoli

Per consentire la creazione di ruoli solo tramite il role manager, utilizza la seguente policy per bloccare la creazione di ruoli non gestori di ruoli. Questa policy consente comunque la creazione di ruoli basata su modelli direttamente tramite l'SDK.

```
{
    "Sid": "DenyNonTemplatedRoleCreation",
    "Effect": "Deny",
    "Action": [
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy",
        "iam:PutRolePermissionsBoundary",
        "iam:TagRole"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "iam:RoleTemplateARN": true
        }
    }
}
```

Informazioni correlate

- Crea ruoli automaticamente con il gestore dei ruoli
- Applica le autorizzazioni con privilegi minimi a un ruolo creato automaticamente
- Panoramica dei modelli di ruolo

Applica le autorizzazioni con privilegi minimi a un ruolo creato automaticamente

Quando si crea una risorsa in una console di servizio, il role manager fornisce il ruolo IAM necessario alla risorsa. Per alcune attività, come l'esecuzione del proprio codice, AWS non è possibile determinare in anticipo quali autorizzazioni sono necessarie. Di conseguenza, un ruolo può consentire più autorizzazioni di quelle richieste dalla risorsa. Una volta individuate le autorizzazioni necessarie alla risorsa, è possibile ridurre le autorizzazioni del ruolo in modo che corrispondano.

La disattivazione del role manager non influisce sui ruoli creati in precedenza dal role manager. È possibile disattivare il role manager senza ridurre le autorizzazioni e ridurre le autorizzazioni di un ruolo senza disabilitare il role manager.

Quando applicare le autorizzazioni con privilegi minimi

Concedere a un ruolo solo le autorizzazioni richieste dalla risorsa è una best practice di sicurezza in linea con il principio del privilegio minimo. Per la maggior parte dei ruoli, il role manager crea ruoli con autorizzazioni ben definite. Per alcuni ruoli, come quelli per le risorse di calcolo o la gestione dell'infrastruttura cloud, crea ruoli con ampie autorizzazioni.

Quando sei pronto per iniziare a definire i ruoli creati dal role manager, ti consigliamo di iniziare con i ruoli associati ai carichi di lavoro e alle risorse più sensibili, ad esempio qualsiasi elemento che gestisca informazioni personali o dati di produzione. Per ulteriori informazioni, consulta Preparazione per le autorizzazioni con privilegi minimi.

Come analizzare gli accessi non utilizzati

Prima di disabilitare il role manager, ti AWS consiglia di IAM Access Analyzer utilizzarlo per creare un analizzatore degli accessi inutilizzati nel tuo account. L'analizzatore confronta le azioni che un ruolo è autorizzato a eseguire con le azioni che ha eseguito e riporta le autorizzazioni non utilizzate. L'analizzatore esamina i ruoli IAM nel tuo account, inclusi i ruoli che hai creato tu stesso.

La posizione in cui vengono visualizzati i risultati dipende da come è stato fornito l'analizzatore:

- Analizzatore fornito dal role manager: quando AWS fornisce l'analizzatore degli accessi inutilizzati per il tuo account (vedi la nota in questa sezione), i risultati vengono visualizzati nella pagina Ruoli della console IAM. Per ogni ruolo, puoi vedere il numero di autorizzazioni non utilizzate e aprire un consiglio. Utilizza la procedura seguente per ridurre le autorizzazioni di un ruolo dalla pagina Ruoli. Dopo aver disabilitato il gestore dei ruoli, la pagina delle impostazioni dell'account fornisce un pulsante Visualizza ruoli che apre la pagina Ruoli, in cui vengono visualizzati i risultati di ciascun ruolo.
- Analizzatore creato da te: quando crei il tuo analizzatore di accessi inutilizzato, i risultati e i consigli vengono visualizzati nella console IAM Access Analyzer. Tale esperienza è rimasta invariata dal role manager. Per esaminare e intervenire sui risultati di accesso non utilizzati, consulta i risultati di IAM Access Analyzer.

> **Note**
>
> Se il gestore dei ruoli è AWS abilitato per il tuo account creato utilizzando la nuova AWS esperienza, non è necessario creare un analizzatore la prima volta che disabiliti il gestore dei ruoli. Dopo aver attivato le funzionalità avanzate e disabilitato il role manager, AWS fornisce un analizzatore di accessi inutilizzato senza costi aggiuntivi per 90 giorni. L'analizzatore ti offre visibilità sulle autorizzazioni dei ruoli non utilizzate e sui consigli relativi all'ambito delle policy, in modo da poter aggiornare i ruoli del tuo account, compresi quelli creati dal role manager, con il privilegio minimo.

Prerequisiti

Per esaminare i risultati di Access Analyzer, è necessaria l'autorizzazione per eseguire l'azione. `access-analyzer:ListFindings` Per applicare un consiglio, è necessaria l'autorizzazione all'esecuzione `iam:CreatePolicy` e `iam:DetachRolePolicy` al ruolo. `iam:AttachRolePolicy`

Per ridurre le autorizzazioni di un ruolo nella console (analizzatore fornito dal role manager)

Questi passaggi si applicano quando viene AWS fornito l'analizzatore di accessi inutilizzato per il tuo account. Se hai creato il tuo analizzatore, rivedi e applica invece i consigli nella console IAM Access Analyzer. Consulta i risultati di IAM Access Analyzer.

1. Accedi Console di gestione AWS e apri la console IAM.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Nella pagina Ruoli, nella colonna Risultati, scegli il link Autorizzazioni non utilizzate per il ruolo di cui desideri ridurre le autorizzazioni.
4. Nella pagina Riduci le autorizzazioni del ruolo, esamina le autorizzazioni non utilizzate segnalate per il ruolo.
5. Confronta le autorizzazioni consigliate con le autorizzazioni correnti del ruolo ed esamina le azioni rimosse dal consiglio.
6. Scegli Apporta modifiche.

Come funzionano le autorizzazioni consigliate

Sulla base dei risultati dell'analizzatore di accessi non utilizzati, IAM crea una o più policy gestite dal cliente che contengono le autorizzazioni consigliate. IAM allega le nuove policy prima di scollegare quelle precedenti, in modo che il ruolo mantenga le proprie autorizzazioni per tutta la durata della modifica.

Ad esempio, considera un ruolo di esecuzione AWS Lambda. IAM Access Analyzer segnala quali autorizzazioni del ruolo sono inutilizzate e consiglia una policy limitata alle azioni eseguite dal ruolo, ad esempio la scrittura di log su Amazon Logs. CloudWatch Esamini il consiglio rispetto a ciò che fa la funzione, confermi che copra tutto ciò di cui ha bisogno la funzione e lo applichi. Il ruolo consente quindi solo le azioni necessarie alla funzione.

> **Important**
>
> La raccomandazione si basa sugli ultimi 30 giorni di attività. Un'autorizzazione che la risorsa utilizza raramente su un orizzonte temporale più lungo, ad esempio per un lavoro trimestrale, può apparire inutilizzata. Rivedi il consiglio in base alle esigenze del tuo carico di lavoro prima di applicarlo.
>
> Dopo aver modificato le politiche o le politiche di attendibilità di un ruolo, il role manager non gestisce più quel ruolo. Il ruolo non raccoglierà alcuna modifica alla versione del modello di ruolo originariamente utilizzato per creare il ruolo.

IAM limita il numero di policy gestite che è possibile allegare a un ruolo. Se l'applicazione del consiglio supera tale limite, la console ti avvisa e non applica la modifica. Per informazioni su questa quota, consultaIAM e AWS STS quote.

Informazioni correlate

- Crea ruoli automaticamente con il gestore dei ruoli
- Gestire l'accesso al role manager
- Panoramica dei modelli di ruolo

Panoramica dei modelli di ruolo

Un modello di ruolo è un modello di ruolo utilizzato per creare nuovi ruoli. Il modello imposta la politica di fiducia e le autorizzazioni del ruolo in modo da ottenere un ruolo pronto all'uso per un servizio senza dover scrivere personalmente le politiche.

Il gestore dei ruoli utilizza modelli AWS di ruolo gestiti, ma puoi anche utilizzare modelli di ruolo senza gestore dei ruoli. Questa pagina spiega come funzionano i modelli di ruolo ed elenca il modello per ogni servizio.

Come funzionano i modelli di ruolo

Un modello di ruolo contiene una policy di fiducia e autorizzazioni e dispone di un Amazon Resource Name (ARN). La politica di fiducia definisce quali responsabili possono assumere il ruolo. Le autorizzazioni definiscono cosa può fare il ruolo. Le versioni dei modelli sono immutabili. Quando AWS aggiorna un modello, pubblica una nuova versione anziché modificarne una esistente.

I modelli di ruolo possono accettare parametri aggiuntivi per personalizzare e definire l'ambito delle proprietà e delle autorizzazioni del ruolo. Ad esempio, un modello potrebbe accettare un `RoleName` per il ruolo che crea o un identificatore di risorsa come un. `bucketName` Quando il gestore dei ruoli crea un ruolo nella console, applica i parametri alle autorizzazioni e alla politica di fiducia per il ruolo. Quando crei un ruolo da un modello direttamente con l'API IAM, fornisci questi parametri.

Per creare un ruolo da un modello, usa l'`iam:AcquireRole`API con l'ARN del modello e passa i parametri richiesti dal modello.

Le definizioni di un modello di ruolo possono cambiare tra le versioni. Per visualizzare la politica di attendibilità e le autorizzazioni correnti per un modello, recuperalo `iam:GetRoleTemplateVersion` utilizzando il modello ARN. AWS consiglia di ispezionare il modello prima di fare affidamento su di esso, specialmente per un modello che garantisce un ampio accesso.

In che modo il gestore dei ruoli utilizza i modelli

Quando si crea una risorsa in una console di servizio supportata, il gestore dei ruoli fornisce un ruolo tratto dal modello che corrisponde al servizio e al caso d'uso. Il gestore dei ruoli sceglie il modello e fornisce i parametri per te. Il risultato è lo stesso ruolo che riceveresti per chiamare`iam:AcquireRole`. Per abilitare il gestore dei ruoli, vediGestire l'accesso al role manager.

Directory dei modelli di ruolo

La tabella seguente elenca il modello di ruolo per ogni servizio supportato dal gestore dei ruoli, con l'ARN del modello e i parametri utilizzati. Usa l'ARN per creare un ruolo dal modello o per verificare quali ruoli manager concede per quel servizio. I servizi sono elencati in ordine alfabetico.

Servizio	Modello di ruolo	ARN modello	Parameters
AWS Backup	AWSBackupDefaultServiceRoleTemplate	`arn:aws:iam::aws:role-template/backup.amazonaws.com/AWSBackupDefaultServiceRoleTemplate:1`	AccountID, RoleName
AWS CloudFormation	PowerUserRoleTemplate	`arn:aws:iam::aws:role-template/iam.amazonaws.com/PowerUserRoleTemplate:1`	AWSServiceName, RoleName
Amazon EKS	AmazonEKSAutoClusterRoleTemplate	`arn:aws:iam::aws:role-template/eks.amazonaws.com/AmazonEKSAutoClusterRoleTemplate:1`	RoleName
Amazon EKS	AmazonEKSAutoNodeRoleTemplate	`arn:aws:iam::aws:role-template/ec2.amazonaws.com/AmazonEKSAutoNodeRoleTemplate:1`	RoleName
AWS Elastic Beanstalk	PowerUserRoleTemplate	`arn:aws:iam::aws:role-template/iam.amazonaws.com/PowerUserRoleTemplate:1`	AWSServiceName, RoleName
Amazon EventBridge	PowerUserRoleTemplate	`arn:aws:iam::aws:role-template/iam.amazonaws.com/PowerUserRoleTemplate:1`	AWSServiceName, RoleName
AWS Glue	AWSGlueCrawlerS3PathsRoleTemplate	`arn:aws:iam::aws:role-template/glue.amazonaws.com/AWSGlueCrawlerS3PathsRoleTemplate:1`	RoleName, AccountID, percorso s3
AWS Glue	AWSGluePowerUserRoleTemplate	`arn:aws:iam::aws:role-template/glue.amazonaws.com/AWSGluePowerUserRoleTemplate:1`	AccountID, RoleName

Servizio	Modello di ruolo	ARN modello	Parameters
AWS Lambda	PowerUser RoleTemplate	`arn:aws:iam::aws:role-template/iam.amazonaws.com/PowerUserRoleTemplate:1`	AWSServiceName, RoleName
Amazon RDS	AmazonRDS DirectoryServiceIntegration RoleTemplate	`arn:aws:iam::aws:role-template/rds.amazonaws.com/AmazonRDSDirectoryServiceIntegrationRoleTemplate:1`	RoleName, AccountID
Amazon RDS	AmazonRDS ExportToS3RoleTemplate	`arn:aws:iam::aws:role-template/rds.amazonaws.com/AmazonRDSExportToS3RoleTemplate:1`	RoleName, AccountID, bucketName
Amazon RDS	AmazonRDS ImportFromS3RoleTemplate	`arn:aws:iam::aws:role-template/rds.amazonaws.com/AmazonRDSImportFromS3RoleTemplate:1`	CUSTOM_KMS_KEY, RoleName AccountId, bucketName, chiave, regione chiave, kms, prefisso AccountId KeyId
Amazon RDS	AmazonRDS MonitoringRoleTemplate	`arn:aws:iam::aws:role-template/rds.amazonaws.com/AmazonRDSMonitoringRoleTemplate:1`	RoleName, AccountID

Servizio	Modello di ruolo	ARN modello	Parameters
Amazon RDS	AmazonRDS ProxyRole Template	`arn:aws:iam::aws:role-template/rds.amazonaws.com/AmazonRDSProxyRoleTemplate:2`	PROXY_WITH_CUSTOM_KEY, PROXY_WITH_DB_CONNECTION, PROXY_WITH_SECRET, AccountID, db, dbUser, KeyID, regione, SecretID RoleName ResourceId
Amazon RDS	AmazonRDS SecretAccessForDMS RoleTemplate	`arn:aws:iam::aws:role-template/rds.amazonaws.com/AmazonRDSSecretAccessForDMSRoleTemplate:1`	CUSTOM_KMS_KEY, AccountID RoleName, kms, regione, SecretID KeyId
Amazon SageMaker Unified Studio	AmazonSageMakerAdminIAMPermissiveExecutionRoleTemplate	`arn:aws:iam::aws:role-template/datazone.amazonaws.com/AmazonSageMakerAdminIAMPermissiveExecutionRoleTemplate:1`	CMK_ENABLED, RoleName AccountID, chiave, regione chiave, kms AccountId KeyId
Amazon SageMaker Unified Studio	AmazonSageMakerUserIAMPermissiveExecutionRoleTemplate	`arn:aws:iam::aws:role-template/datazone.amazonaws.com/AmazonSageMakerUserIAMPermissiveExecutionRoleTemplate:1`	RoleName, AccountID

Servizio	Modello di ruolo	ARN modello	Parameters
AWS Secrets Manager	AWSSecretsManagerRotationRoleTemplate	`arn:aws:iam::aws:role-template/secretsmanager.amazonaws.com/AWSSecretsManagerRotationRoleTemplate:1`	ADMIN_RESOURCE_ENABLED, CMK_ENABLED, AccountID, adminType, kms, regione RoleName, ResourceTypeKeyArn
Amazon CloudWatch	PowerUserRoleTemplate	`arn:aws:iam::aws:role-template/iam.amazonaws.com/PowerUserRoleTemplate:1`	AWSServiceName, RoleName

Informazioni correlate

- Crea ruoli automaticamente con il gestore dei ruoli
- Gestire l'accesso al role manager
- Riferimento all'API IAM

Gestione del ruolo IAM

Prima che un utente, applicazione o servizio possa utilizzare un ruolo che è stato creato, è necessario concedere le autorizzazioni per passare al ruolo. È possibile utilizzare qualsiasi policy collegata a gruppi o utenti per concedere le autorizzazioni necessarie. In questa sezione viene descritto come concedere agli utenti l'autorizzazione per l'utilizzo di un ruolo. Spiega inoltre come l'utente può passare a un ruolo dagli Console di gestione AWS strumenti per Windows PowerShell, da AWS Command Line Interface (AWS CLI) e dall'`AssumeRole`API.

> **Important**
>
> Se crei un ruolo a livello programmatico anziché nella console IAM, hai l'opzione per aggiungere un `Path` con un massimo di 512 caratteri in aggiunta a `RoleName`, che può

contenere fino a 64 caratteri. Tuttavia, se intendi utilizzare un ruolo con la funzione Cambia ruolo in Console di gestione AWS, la combinazione `Path RoleName` non può superare i 64 caratteri.

Argomenti

- Visualizzazione dell'accesso per il ruolo
- Generazione di una policy basata sulle informazioni di accesso
- Concedere le autorizzazioni agli utenti per cambiare ruoli
- Concedi a un utente le autorizzazioni per passare un ruolo a un AWS service
- Revocare le credenziali di sicurezza temporanee per i ruoli IAM
- Aggiornamento di un ruolo collegato ai servizi
- Aggiornamento di una policy di attendibilità del ruolo
- Aggiornamento delle autorizzazioni per un ruolo
- Aggiornamento delle impostazioni per un ruolo
- Eliminare i ruoli o i profili delle istanze

Visualizzazione dell'accesso per il ruolo

Prima di modificare le autorizzazioni per un ruolo, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Generazione di una policy basata sulle informazioni di accesso

Talvolta, è possibile concedere autorizzazioni a un'entità IAM (utente o ruolo) oltre a quelle richieste. Per ottimizzare le autorizzazioni concesse, puoi generare una policy IAM basata sull'attività di accesso per un'entità. IAM Access Analyzer esamina AWS CloudTrail i log e genera un modello di policy che contiene le autorizzazioni utilizzate dall'entità nell'intervallo di date specificato. È possibile utilizzare il modello per creare una policy gestita con autorizzazioni granulari e quindi collegarla al ruolo IAM. In questo modo, concedi solo le autorizzazioni necessarie all'utente o al ruolo per interagire con le AWS risorse per il tuo caso d'uso specifico. Per ulteriori informazioni, consulta Generazione di policy per Sistema di analisi degli accessi IAM.

Concedere le autorizzazioni agli utenti per cambiare ruoli

Quando un amministratore crea un ruolo per l'accesso multi-account, stabilisce l'attendibilità tra l'account proprietario del ruolo e le risorse (account che determina l'attendibilità) e l'account che contiene gli utenti (account attendibile). A tale scopo, l'amministratore dell'account attendibile specifica il numero dell'account attendibile come `Principal` nella policy di attendibilità del ruolo. Ciò consente potenzialmente a qualsiasi utente nell'account attendibile di assumere il ruolo. Per completare la configurazione, l'amministratore dell'account attendibile deve fornire a determinati gruppi o utenti dell'account l'autorizzazione per il passaggio al ruolo.

Concessione dell'autorizzazione per passare a un ruolo

1. In qualità di amministratore dell'account attendibile, crea una nuova policy per l'utente oppure modifica una policy esistente per aggiungere gli elementi richiesti. Per informazioni dettagliate, vedi Creazione o modifica della policy.
2. Quindi decidi come desideri eseguire la condivisione delle informazioni sul ruolo:

   - Role link (Link del ruolo): invia agli utenti un collegamento che indirizza alla pagina Switch Role (Cambia ruolo) con tutti i dettagli già compilati.
   - ID account o alias: fornisci a ciascun utente il nome del ruolo insieme al numero dell'ID account o all'alias dell'account. L'utente accede quindi alla pagina Switch Role (Cambia ruolo) e aggiunge i dettagli manualmente.

   Per informazioni dettagliate, vedi Fornire informazioni all'utente.

Tieni presente che puoi cambiare ruolo solo quando accedi come utente IAM, ruolo o SAML-federated ruolo federato con identità web. Non è possibile cambiare i ruoli se si effettua l'accesso come Utente root dell'account AWS.

> **Important**
>
> Non è possibile passare da un ruolo Console di gestione AWS a un ruolo che richiede un valore. ExternalId È possibile passare a tale ruolo solo chiamando l'API `AssumeRole` che supporta il parametro `ExternalId`.

> **Note**
>
> - In questo argomento sono descritte le policy per un utente poiché sostanzialmente concedi autorizzazioni a un utente per l'esecuzione di un'operazione. Tuttavia, consigliamo di non concedere le autorizzazioni direttamente a un singolo utente. Quando un utente assume un ruolo, gli vengono assegnate le autorizzazioni associate a quel ruolo.
> - Quando si cambia ruolo in Console di gestione AWS, la console utilizza sempre le credenziali originali per autorizzare il passaggio. Questo vale sia che tu acceda come utente IAM, come ruolo o come SAML-federated ruolo federato con identità web. Ad esempio, se passi al RuoloA, IAM; utilizza le tue credenziali utente originali o le credenziali del ruolo federato per determinare se è possibile assumere il RuoloA. Se provi quindi a passare al RuoloB mentre stai utilizzando il RuoloA, per autorizzare il tentativo vengono utilizzate le credenziali utente originali o le credenziali del ruolo federato. Le credenziali per RuoloA non vengono utilizzate per questa operazione.

Argomenti

- Creazione o modifica della policy
- Fornire informazioni all'utente

Creazione o modifica della policy

Una policy che concede a un utente l'autorizzazione di assumere un ruolo deve includere una dichiarazione con effetto `Allow` per quanto segue:

- L'operazione `sts:AssumeRole`
- L'Amazon Resource Name (ARN) del ruolo in un elemento `Resource`

Agli utenti che ottengono la policy (mediante l'appartenenza a un gruppo o collegata direttamente) è consentito cambiare ruoli sulla risorsa specificata.

> **Note**
>
> Se `Resource` è impostato su `*`, l'utente può assumere qualsiasi ruolo in qualsiasi account che considera l'account utente attendibile. (In altre parole, la policy di attendibilità del ruolo specifica l'account dell'utente come `Principal`). Come best practice, si consiglia di seguire

il principio di privilegio minimo e specificare l'ARN completo solo per i ruoli necessari per l'utente.

L'esempio seguente mostra una policy che consente all'utente di assumere ruoli in un unico account. Inoltre, la policy utilizza un carattere jolly (*) per specificare che l'utente può passare a un ruolo solo se il nome del ruolo inizia con le lettere `Test`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Resource": "arn:aws:iam::111122223333:role/Test*"
    }
}
```

Note

Le autorizzazioni che il ruolo concede all'utente non vengono aggiunte alle autorizzazioni già concesse per l'utente. Quando un utente passa a un ruolo, l'utente rinuncia temporaneamente alle proprie autorizzazioni originali in cambio di quelle concesse dal ruolo. Quando l'utente esce dal ruolo, le autorizzazioni originali dell'utente vengono ripristinate automaticamente. Ad esempio, assumiamo che le autorizzazioni dell'utente permettano di utilizzare le istanze Amazon EC2, ma le policy di autorizzazione del ruolo non concedano tali autorizzazioni. In tal caso, durante l'utilizzo del ruolo, l'utente non potrà utilizzare le istanze Amazon EC2 nella console. Inoltre, le credenziali temporanee ottenute tramite `AssumeRole` non funzioneranno con le istanze Amazon EC2 in modo programmatico.

Fornire informazioni all'utente

Dopo aver creato un ruolo e concesso all'utente le autorizzazioni per passare a tale ruolo, è necessario fornire all'utente quanto segue:

- Il nome del ruolo

- L'ID o l'alias dell'account che contiene il ruolo

Puoi semplificare le operazioni per gli utenti inviando loro un collegamento preconfigurato con l'ID account e il nome del ruolo. Puoi visualizzare il collegamento al ruolo dopo aver completato la procedura guidata Crea ruolo selezionando il banner Visualizza ruolo o nella pagina Riepilogo del ruolo per qualsiasi ruolo abilitato per più account.

È inoltre possibile utilizzare il formato seguente per creare manualmente il collegamento. Sostituisci l'ID account o alias e il nome del ruolo per i due parametri nel seguente esempio:

```
https://signin.aws.amazon.com/switchrole?
account=your_account_ID_or_alias&roleName=optional_path/role_name
```

Consigliamo di suggerire agli utenti di consultare l'argomento Passare da un utente a un ruolo IAM (console) per guidarli nel processo. Per risolvere i problemi più comuni che si possono verificare quando si assume un ruolo, consulta la pagina Non è possibile assumere un ruolo.

Considerazioni

- Se crei il ruolo in maniera programmatica, puoi crearlo con un percorso e un nome. In tal caso, è necessario fornire il percorso completo e il nome del ruolo agli utenti in modo che possano specificare queste informazioni sulla pagina Cambia ruolo della Console di gestione AWS. Ad esempio: `division_abc/subdivision_efg/role_XYZ`.
- Se crei il ruolo in maniera programmatica, potrai aggiungere un `Path` con un massimo di 512 caratteri e un `RoleName`. Il nome del ruolo può contenere un massimo di 64 caratteri. Tuttavia, per utilizzare un ruolo con la funzione Cambia ruolo in Console di gestione AWS, la combinazione `Path` e `RoleName` non può superare i 64 caratteri.

- Per motivi di sicurezza, puoi esaminare AWS CloudTrail i log per scoprire chi ha eseguito un'azione in AWS. È possibile utilizzare la chiave di condizione `sts:SourceIdentity` nella policy di attendibilità del ruolo per richiedere agli utenti di specificare un'identità quando assumono un ruolo. Ad esempio, è possibile richiedere che gli utenti IAM specifichino il proprio nome utente come identità di origine. In questo modo è possibile determinare quale utente ha eseguito un'operazione specifica in AWS. Per ulteriori informazioni, consulta `sts:SourceIdentity`. Puoi utilizzare `sts:RoleSessionName` anche per richiedere agli utenti di specificare un nome di sessione quando assumono un ruolo. Ciò consente di distinguere tra le sessioni di ruolo quando un ruolo viene utilizzato da principali diversi.

Concedi a un utente le autorizzazioni per passare un ruolo a un AWS service

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Ciò consente al servizio di assumere successivamente il ruolo ed eseguire operazioni per tuo conto. Per la maggior parte dei servizi, è sufficiente passare il ruolo al servizio una sola volta durante la configurazione e non ogni volta che il servizio assume il ruolo. Ad esempio, si supponga di disporre di un'applicazione in esecuzione in un'istanza Amazon EC2. Tale applicazione richiede credenziali temporanee per l'autenticazione e autorizzazioni per autorizzare l'applicazione a eseguire operazioni in AWS. Quando configuri l'applicazione, devi passare un ruolo ad Amazon EC2 per l'utilizzo con l'istanza che fornisce tali credenziali. È possibile definire le autorizzazioni per le applicazioni in esecuzione nell'istanza collegando una policy IAM al ruolo. L'applicazione assume il ruolo ogni volta che è necessario per eseguire le operazioni consentite dal ruolo.

Per passare un ruolo (e le relative autorizzazioni) a un AWS servizio, un utente deve disporre delle autorizzazioni per passare il ruolo al servizio. Ciò consente agli amministratori di garantire che solo gli utenti autorizzati possano configurare un servizio con un ruolo che concede le autorizzazioni. Per consentire a un utente di passare un ruolo a un AWS servizio, devi concedere l'`PassRole`autorizzazione all'utente, al ruolo o al gruppo IAM dell'utente.

> **Warning**
>
> - Puoi utilizzare l'`PassRole`autorizzazione solo per passare un ruolo IAM a un servizio che condivide lo stesso AWS account. Per passare un ruolo nell'Account A a un servizio nell'Account B, devi prima creare un ruolo IAM nell'Account B che possa assumere il ruolo dall'Account A, quindi il ruolo nell'Account B può essere passato al servizio. Per informazioni dettagliate, vedi Accesso alle risorse multi-account in IAM.
> - Non cercare di controllare chi può passare un ruolo assegnando tag al ruolo e utilizzando la chiave di condizione `ResourceTag` in una policy con l'operazione `iam:PassRole`. Questo approccio non produce risultati affidabili.

Quando imposti l'autorizzazione `PassRole`, devi assicurarti che un utente non invii un ruolo dove il ruolo dispone di più autorizzazioni di quelle che desideri che l'utente abbia. Ad esempio, Alice potrebbe non essere autorizzata a eseguire alcune operazioni su Amazon S3. Se Alice potesse trasferire un ruolo a un servizio che consente le azioni di Amazon S3, il servizio potrebbe eseguire azioni Amazon S3 per conto di Alice durante l'esecuzione del processo.

Quando si specifica un ruolo collegato ai servizi, è necessario disporre anche delle autorizzazioni per inoltrare tale ruolo al servizio. Alcuni servizi creano automaticamente un ruolo collegato ai servizi nell'account quando si esegue un'azione in quel servizio. Ad esempio, Amazon EC2 Auto Scaling crea il ruolo collegato ai servizi `AWSServiceRoleForAutoScaling` la prima volta che crei un gruppo Auto Scaling. Se provi a specificare il ruolo collegato ai servizi quando crei un gruppo con scalabilità automatica senza l'autorizzazione `iam:PassRole`, viene visualizzato un messaggio di errore. Se non specifichi esplicitamente il ruolo, l'autorizzazione `iam:PassRole` non è richiesta e l'impostazione predefinita prevede l'utilizzo del ruolo `AWSServiceRoleForAutoScaling` per tutte le operazioni eseguite su quel gruppo. Per scoprire i servizi che supportano i ruoli collegati ai servizi, consulta AWS servizi supportati da IAM. Per scoprire quali servizi creano automaticamente un ruolo collegato ai servizi quando si esegue un'operazione in quel servizio, selezionare il collegamento Yes (Sì) e visualizzare il ruolo collegato ai servizi per il servizio.

Un utente può passare un ruolo ARN come parametro in qualsiasi operazione API che utilizza il ruolo per assegnare le autorizzazioni al servizio. Il servizio quindi verifica se l'utente dispone dell'autorizzazione `iam:PassRole`. Per limitare l'utente a passare solo i ruoli approvati, puoi filtrare l'autorizzazione `iam:PassRole` con l'elemento `Resources` dell'istruzione della policy IAM.

Puoi utilizzare l'`Condition`elemento in una policy JSON per testare il valore delle chiavi incluse nel contesto della richiesta di tutte le AWS richieste. Per ulteriori informazioni sull'utilizzo delle chiavi di condizione in una policy, consulta Elementi della policy JSON IAM: Condizione. La chiave di condizione `iam:PassedToService` può essere utilizzata per specificare il principale del servizio del servizio a cui è possibile passare un ruolo. Per saperne di più sull'utilizzo della chiave di `iam:PassedToService` condizione in una policy, vedi iam: PassedToService.

Esempio 1

Si immagini di voler concedere a un utente la possibilità di trasferire uno qualsiasi dei set di ruoli approvati al servizio Amazon EC2 all'avvio di un'istanza. È necessario disporre di tre elementi:

- Una policy di autorizzazioni IAM collegata al ruolo che determina quali operazioni può compiere il ruolo. Definire l'ambito delle autorizzazioni in modo da includere solo le operazioni che il ruolo deve effettuare e sole le risorse necessarie per tali operazioni. Puoi utilizzare una policy di autorizzazioni IAM AWS gestita o creata dal cliente.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
```

```
        "Effect": "Allow",
        "Action": [ "A list of the permissions the role is allowed to use" ],
        "Resource": [ "A list of the resources the role is allowed to access" ]
    }
}
```

- Una policy di attendibilità per il ruolo che consente al servizio di assumere tale ruolo. Ad esempio, è possibile collegare la seguente policy di affidabilità al ruolo con l'operazione `UpdateAssumeRolePolicy`. Questa policy di attendibilità consente ad Amazon EC2 di utilizzare il ruolo e le autorizzazioni associate al ruolo.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Sid":
 "TrustPolicyStatementThatAllowsEC2ServiceToAssumeTheAttachedRole",
        "Effect": "Allow",
        "Principal": { "Service": "ec2.amazonaws.com" },
       "Action": "sts:AssumeRole"
    }
}
```

- Una policy di autorizzazioni IAM collegata all'utente IAM che consente all'utente di trasferire solo i ruoli approvati. In genere si aggiunge `iam:GetRole` a `iam:PassRole` in modo che l'utente possa ottenere i dettagli del ruolo da passare. In questo esempio, l'utente può passare solo i ruoli esistenti nell'account specificato con nomi che iniziano con `EC2-roles-for-XYZ-`:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/EC2-roles-for-XYZ-*"
        }
    ]
```

```
}
```

Ora l'utente può avviare un'istanza Amazon EC2 con un ruolo assegnato. Le applicazioni in esecuzione nell'istanza possono accedere alle credenziali temporanee per il ruolo tramite i metadati del profilo dell'istanza. Le policy delle autorizzazioni collegate al ruolo determinano cosa può fare l'istanza.

Esempio 2

Amazon Relational Database Service (Amazon RDS) supporta una funzione chiamata Monitoraggio avanzato. Questa funzione consente ad Amazon RDS di monitorare un'istanza di database tramite un agente. Consente inoltre ad Amazon RDS di registrare le metriche su Amazon Logs. CloudWatch Per abilitare questa funzione, è necessario creare un ruolo di servizio per fornire le autorizzazioni Amazon RDS per monitorare e scrivere i parametri per i log.

Come creare un ruolo IAM per il monitoraggio avanzato di Amazon RDS

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Selezionare Roles (Ruoli), quindi selezionare Create role (Crea ruolo).
3. Scegli il tipo di ruolo del AWS servizio, quindi per i casi d'uso per altri Servizi AWS, scegli il servizio RDS. Scegli RDS - Enhanced Monitoring (RDS - Monitoraggio avanzato), quindi seleziona Next (Successivo).
4. Scegli la politica delle AmazonRDSEnhancedMonitoringRole autorizzazioni.
5. Scegli Next (Successivo).
6. In Role name (Nome ruolo), inserisci un nome del ruolo che consenta di identificarne lo scopo. I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS. Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole. Quando un nome di ruolo viene visualizzato ai clienti nella console, ad esempio durante la procedura di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole. Poiché varie entità possono fare riferimento al ruolo, non puoi modificare il nome del ruolo dopo averlo creato.
7. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
8. (Facoltativo) Aggiungi metadati all'utente collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.

9. Rivedere il ruolo e scegliere Crea ruolo.

Il ruolo ottiene automaticamente una policy di affidabilità che concede le autorizzazioni del servizio `monitoring.rds.amazonaws.com` per assumere il ruolo. Dopo l'avvio, Amazon RDS potrà eseguire tutte le operazioni consentite dalla policy `AmazonRDSEnhancedMonitoringRole`.

L'utente a cui si desidera accedere a Enhanced Monitoring deve disporre di una policy che includa un'istruzione che consenta all'utente di elencare i ruoli RDS e un'istruzione che consenta all'utente di passare il ruolo, come la seguente. Utilizza il tuo numero di account e sostituisci il nome del ruolo con il nome fornito nel passaggio 6.

```
    {
      "Sid": "PolicyStatementToAllowUserToListRoles",
      "Effect": "Allow",
      "Action": ["iam:ListRoles"],
      "Resource": "*"
    },
    {
        "Sid": "PolicyStatementToAllowUserToPassOneSpecificRole",
        "Effect": "Allow",
        "Action": [ "iam:PassRole" ],
        "Resource": "arn:aws:iam::account-id:role/RDS-Monitoring-Role"
    }
```

È possibile combinare questa istruzione con dichiarazioni in un'altra policy o collocarla nella policy personalizzata. Invece, per specificare che l'utente può passare qualsiasi ruolo che inizia con `RDS-`, puoi sostituire il nome del ruolo nella risorsa ARN con un carattere jolly, come nell'esempio seguente.

```
        "Resource": "arn:aws:iam::account-id:role/RDS-*"
```

`iam: PassRole` azioni in AWS CloudTrail log

`PassRole`non è una chiamata API. `PassRole`è un'autorizzazione, il che significa che non vengono generati CloudTrail log per IAM`PassRole`. Per verificare quali ruoli vengono passati a chi CloudTrail, è necessario esaminare il CloudTrail registro che ha creato o modificato la AWS risorsa che riceve il ruolo. Servizi AWS Ad esempio, un ruolo viene passato a una AWS Lambda funzione al momento della creazione. Il log per l'operazione `CreateFunction` mostra un record del ruolo passato alla funzione.

Revocare le credenziali di sicurezza temporanee per i ruoli IAM

> **Warning**
>
> Se segui i passaggi in questa pagina, a tutti gli utenti con sessioni correnti create assumendo il ruolo viene negato l'accesso a tutte le AWS azioni e le risorse. Questo può causare la perdita di dati non salvati da parte degli utenti.

Quando consenti agli utenti di accedere a sessioni Console di gestione AWS con una durata di sessione lunga (ad esempio 12 ore), le loro credenziali temporanee non scadono così rapidamente. Se gli utenti espongono inavvertitamente le proprie credenziali a una terza parte non autorizzata, tale parte ha accesso per la durata della sessione. Tuttavia, è possibile revocare immediatamente tutte le autorizzazioni per le credenziali del ruolo rilasciate prima di un certo periodo di tempo, se necessario. Tutte le credenziali temporanee per quel ruolo emesse prima del momento specificata diventano non valide. Questo costringe tutti gli utenti a ripetere l'autenticazione e a richiedere nuove credenziali.

> **Note**
>
> Non è possibile revocare la sessione per un ruolo collegato ai servizi.

Quando si revocano le autorizzazioni per un ruolo utilizzando la procedura AWS riportata in questo argomento, al ruolo viene associata una nuova politica in linea che nega tutte le autorizzazioni a tutte le azioni. Include una condizione che applica le restrizioni solo se l'utente ha assunto il ruolo prima del momento in cui sono state revocate le autorizzazioni. Se l'utente assume il ruolo dopo la revoca delle autorizzazioni, la policy di rifiuto non si applica a quell'utente.

Per ulteriori informazioni sulla negazione dell'accesso, consulta Disabilitazione delle autorizzazioni per le credenziali di sicurezza temporanee.

> **Important**
>
> La policy di rifiuto si applica a tutti gli utenti con il ruolo specificato, non solo a quelli con sessioni della console di durata più lunga.

Autorizzazioni minime per revocare le autorizzazioni di sessione da un ruolo

Per revocare le autorizzazioni di sessione da un ruolo, è necessario disporre dell'autorizzazione `PutRolePolicy` per il ruolo. In questo modo è possibile collegare la policy inline `AWSRevokeOlderSessions` al ruolo.

Revoca delle autorizzazioni di sessione

Puoi revocare le autorizzazioni della sessione da un ruolo per negare tutte le autorizzazioni a tutti gli utenti che hanno assunto il ruolo.

> **Note**
>
> Non è possibile modificare i ruoli in IAM creati dai set di autorizzazioni del Centro identità IAM. È necessario revocare la sessione attiva del set di autorizzazioni per un utente nel Centro identità IAM. Per ulteriori informazioni, consulta Revocare le sessioni attive di un ruolo IAM create dai set di autorizzazioni nella Guida per l'utente del Centro identità IAM.

Per rifiutare immediatamente tutte le autorizzazioni a qualsiasi utente corrente con credenziali del ruolo

1. Accedi e apri la console IAM all'indirizzo. Console di gestione AWS https://console.aws.amazon.com/iam/
2. Nel pannello di navigazione scegli Ruoli, quindi seleziona il nome (non la casella di controllo) del ruolo per cui desideri revocare le autorizzazioni.
3. Nella pagina Summary (Riepilogo) per il ruolo selezionato, selezionare la scheda Revoke sessions (Revoca sessioni).
4. Nelle scheda Revoke sessions (Revoca sessioni) selezionare Revoke active sessions (Revoca sessioni attive).
5. AWS ti chiede di confermare l'azione. Seleziona la casella di controllo Riconosco che sto revocando tutte le sessioni attive per questo ruolo. e scegli Revoca le sessioni attive nella finestra di dialogo.

   IAM quindi collega al ruolo una policy denominata `AWSRevokeOlderSessions`. Dopo aver scelto Revoca sessioni attive, la policy rifiuta l'accesso a tutti gli utenti che hanno assunto il ruolo in passato e per circa 30 secondi nel futuro. Questa scelta temporale futura tiene conto del ritardo di propagazione della policy per gestire una nuova sessione acquisita o rinnovata

prima che la policy aggiornata entrasse in vigore in una determinata regione. Gli utenti che assumono il ruolo più di 30 secondi dopo aver selezionato Revoca sessioni attive non saranno interessati. Per scoprire perché le modifiche non sono sempre immediatamente visibili, consulta Le modifiche che apporto non sono sempre immediatamente visibili.

> **Note**
>
> Se scegli nuovamente Revoca sessioni in un secondo momento, l'indicatore di data e timestamp della policy viene aggiornato e nega nuovamente tutte le autorizzazioni a qualsiasi utente che ha assunto il ruolo prima della nuova ora specificata.

Gli utenti validi le cui sessioni sono revocate in questo modo devono acquisire credenziali provvisorie per una nuova sessione per continuare a lavorare. Le credenziali vengono memorizzate nella AWS CLI cache fino alla loro scadenza. Per forzare la CLI a eliminare e aggiornare le credenziali memorizzate nella cache che non sono più valide, eseguire uno dei seguenti comandi:

Linux, macOS o Unix

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\.aws\cli\cache
```

Revoca delle autorizzazioni di sessione prima di un orario specificato

Puoi anche revocare le autorizzazioni di sessione in qualsiasi momento a tua scelta utilizzando AWS CLI o l'SDK per specificare un valore per la `era: TokenIssueTime` chiave nell'elemento Condition di una politica.

Questa policy nega tutte le autorizzazioni, quando il valore di `aws:TokenIssueTime` è precedente alla data e ora specificate. Il valore di `aws:TokenIssueTime` corrisponde al momento in cui sono state create le credenziali di sicurezza provvisorie. Il `aws:TokenIssueTime` valore è presente solo nel contesto delle AWS richieste firmate con credenziali di sicurezza temporanee, pertanto l'istruzione Deny nella policy non influisce sulle richieste firmate con le credenziali a lungo termine dell'utente IAM.

Questa policy può essere collegata a un ruolo. In questo caso, la policy influisce solo sulle credenziali di sicurezza provvisorie create da tale ruolo prima della data e ora specificate.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "DateLessThan": {"aws:TokenIssueTime": "2014-05-07T23:47:00Z"}
    }
  }
}
```

Gli utenti validi le cui sessioni sono revocate in questo modo devono acquisire credenziali provvisorie per una nuova sessione per continuare a lavorare. Le credenziali vengono memorizzate AWS CLI nella cache fino alla loro scadenza. Per forzare la CLI a eliminare e aggiornare le credenziali memorizzate nella cache che non sono più valide, eseguire uno dei seguenti comandi:

Linux, macOS o Unix

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\.aws\cli\cache
```

Aggiornamento di un ruolo collegato ai servizi

Il metodo utilizzato per modificare un ruolo collegato ai servizi dipende dal servizio. Alcuni servizi consentono di modificare le autorizzazioni per un ruolo collegato ai servizi dalla console di servizio, dalle API o dalla CLI. Tuttavia, dopo aver creato un ruolo collegato ai servizi, non è possibile modificare il nome del ruolo poiché varie entità possono farvi riferimento. Puoi modificare la descrizione di qualsiasi ruolo dalla console IAM, dall'API o dalla CLI.

Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta la pagina AWS servizi supportati da IAM e cerca i servizi per cui è indicato Sì nella colonna Ruolo collegato ai servizi. Per scoprire se il servizio supporta la modifica del ruolo collegato ai servizi, selezionare il link Sì per visualizzare il ruolo collegato ai servizi per quel servizio.

Modifica della descrizione di un ruolo collegato ai servizi (console)

Puoi utilizzare la console IAM per modificare la descrizione di un ruolo collegato ai servizi.

Per modificare la descrizione di un ruolo collegato ai servizi (console)

1. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli).
2. Scegliere il nome del ruolo da modificare.
3. Nella parte destra di Role description (Descrizione ruolo), scegliere Edit (Modifica).
4. Digita una nuova descrizione nella casella e scegli Save (Salva).

Modifica della descrizione di un ruolo collegato ai servizi (AWS CLI)

Puoi utilizzare i comandi IAM di AWS CLI per modificare la descrizione di un ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato ai servizi (AWS CLI)

1. (Facoltativo) Per visualizzare la descrizione attuale di un ruolo, utilizza i seguenti comandi:

   ```
   aws iam get-role --role-name ROLE-NAME
   ```

   Per fare riferimento ai ruoli con i comandi della CLI utilizza il nome del ruolo, non l'ARN. Ad esempio, per fare riferimento a un ruolo il cui ARN è `arn:aws:iam::123456789012:role/myrole`, puoi usare **`myrole`**.
2. Per aggiornare la descrizione di un ruolo collegato ai servizi, utilizza il seguente comando:

   ```
   aws iam update-role --role-name ROLE-NAME --description OPTIONAL-DESCRIPTION
   ```

Modifica di una descrizione di ruolo (API) collegata al servizio AWS

È possibile utilizzare l' AWS API per modificare la descrizione di un ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato al servizio (API)AWS

1. (Facoltativo) Per visualizzare l'attuale descrizione per un ruolo, effettua una chiamata all'operazione seguente e specifica il nome del ruolo:

   AWS API: GetRole

2. Per aggiornare la descrizione di un ruolo, effettua una chiamata all'operazione seguente e specifica il nome (e facoltativamente la descrizione) del ruolo:

   AWS API: UpdateRole

Aggiornamento di una policy di attendibilità del ruolo

Per cambiare l'utente che può assumere un ruolo, modifica la policy di affidabilità del ruolo. Non puoi modificare la policy di attendibilità per un ruolo collegato al servizio.

> **Note**
>
> - Se un utente viene elencato come principale in una policy di attendibilità del ruolo ma non può assumere il ruolo, controlla il limite delle autorizzazioni dell'utente. Se è impostato un limite delle autorizzazioni per l'utente, questo deve consentire l'operazione `sts:AssumeRole`.
> - Per consentire agli utenti di assumere nuovamente il ruolo corrente all'interno di una sessione di ruolo, specifica il ruolo ARN o Account AWS ARN come principale nella politica di fiducia dei ruoli. Servizi AWS che forniscono risorse di calcolo come Amazon EC2, Amazon ECS, Amazon EKS e Lambda forniscono credenziali temporanee e le aggiornano automaticamente. Ciò garantisce di disporre sempre di un set di credenziali valido. Per questi servizi, non è necessario riassumere il ruolo attuale per ottenere credenziali temporanee. Tuttavia, se intendi passare tag di sessione o una Policy di sessione, devi riassumere il ruolo attuale.

Aggiornamento di una policy di attendibilità del ruolo (console)

Per modificare una politica di fiducia dei ruoli nel Console di gestione AWS

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Ruoli.
3. Nell'elenco di ruoli dell'account selezionare il nome del ruolo da modificare.
4. Scegli la scheda Relazioni di attendibilità e quindi Modifica policy di attendibilità.
5. Modificare la policy di affidabilità in base alle esigenze. Per aggiungere ulteriori entità principali che possono assumere il ruolo, specificarle nell'elemento `Principal`. Ad esempio, il seguente frammento di policy mostra come fare riferimento a due elementi Account AWS nell'`Principal`elemento:

   ```
   "Principal": {
     "AWS": [
       "arn:aws:iam::111122223333:root",
       "arn:aws:iam::444455556666:root"
     ]
   },
   ```

   Se specifichi un'entità principale in un altro account, l'aggiunta di un account alla policy di attendibilità di un ruolo è solo una parte della creazione della relazione di trust tra più account. Per impostazione predefinita, gli utenti negli account attendibili non possono assumere il ruolo. L'amministratore del nuovo account attendibile deve concedere agli utenti l'autorizzazione ad assumere il ruolo. A tale scopo, l'amministratore deve creare o modificare una policy collegata all'utente per consentire all'utente di accedere all'operazione `sts:AssumeRole`. Per ulteriori informazioni, consultare la procedura seguente o Concedere le autorizzazioni agli utenti per cambiare ruoli.

   Il seguente frammento di policy mostra come fare riferimento a due AWS servizi nell'elemento: `Principal`

   ```
   "Principal": {
     "Service": [
       "opsworks.amazonaws.com",
       "ec2.amazonaws.com"
     ]
   ```

```
},
```

6. Una volta completata la modifica della policy di attendibilità, scegli Update policy (Aggiorna policy) per salvare le modifiche.

   Per ulteriori informazioni sulla sintassi e sulla struttura della policy, consultare Politiche e autorizzazioni in AWS Identity and Access Management e Documentazione di riferimento degli elementi delle policy JSON IAM.

Per permettere agli utenti in un account esterno attendibile di usare il ruolo (console)

Per ulteriori informazioni e dettagli su questa procedura, consultare Concedere le autorizzazioni agli utenti per cambiare ruoli.

1. Accedi all'esterno attendibile. Account AWS
2. Stabilire se collegare le autorizzazioni a un utente o a un gruppo. Nel riquadro di navigazione della console IAM, scegli Users (Utenti) o User groups (Gruppi di utenti) in base alle esigenze.
3. Scegliere il nome dell'utente o del gruppo a cui si desidera concedere l'accesso e selezionare la scheda Permissions (Autorizzazioni).
4. Esegui una delle seguenti operazioni:

   - Per modificare una policy gestita dal cliente, selezionare il nome della policy, Edit policy (Modifica policy) e la scheda JSON. Non è possibile modificare una policy AWS gestita. AWS le politiche gestite vengono visualizzate con l' AWS icona ().
     Per ulteriori informazioni sulle differenze tra le policy gestite da AWS e quelle gestite dal cliente, consultare Policy gestite e policy inline.
   - Per modificare una policy inline, selezionare la freccia accanto al nome della policy e scegliere Edit policy (Modifica policy).
5. Nell'editor di policy aggiungere un nuovo elemento `Statement` che specifica quanto segue:

```
{
  "Effect": "Allow",
  "Action": "sts:AssumeRole",
  "Resource": "arn:aws:iam::ACCOUNT-ID:role/ROLE-NAME"
}
```

Sostituire l'ARN nell'istruzione con l'ARN del ruolo che l'utente può assumere.

6. Seguire le indicazioni sullo schermo per completare la modifica della policy.

Aggiornamento di una politica di fiducia dei ruoli (AWS CLI)

È possibile utilizzare il AWS CLI per modificare chi può assumere un ruolo.

Per modificare una politica di fiducia del ruolo (AWS CLI)

1. (Facoltativo) Se non si conosce il nome del ruolo da modificare, eseguire il comando seguente per elencare i ruoli nell'account:

   - aws iam list-roles

2. (Facoltativo) Per visualizzare la policy di affidabilità corrente per un ruolo, eseguire il comando seguente:

   - aws iam get-role

3. Per modificare le entità principali attendibili che possono accedere al ruolo, creare un file di testo con la policy di affidabilità aggiornata. È possibile usare qualsiasi editor di testo per creare la policy.

   Ad esempio, la policy di attendibilità seguente illustra come fare riferimento a due Account AWS nell'elemento `Principal`. Ciò consente agli utenti che si trovano all'interno Account AWS di due utenti separati di assumere questo ruolo.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {"AWS": [
            "arn:aws:iam::111122223333:root",
            "arn:aws:iam::444455556666:root"
        ]},
        "Action": "sts:AssumeRole"
    }
}
```

Se specifichi un'entità principale in un altro account, l'aggiunta di un account alla policy di attendibilità di un ruolo è solo una parte della creazione della relazione di trust tra più account. Per impostazione predefinita, gli utenti negli account attendibili non possono assumere il ruolo. L'amministratore del nuovo account attendibile deve concedere agli utenti l'autorizzazione ad assumere il ruolo. A tale scopo, l'amministratore deve creare o modificare una policy collegata all'utente per consentire all'utente di accedere all'operazione `sts:AssumeRole`. Per ulteriori informazioni, consultare la procedura seguente o Concedere le autorizzazioni agli utenti per cambiare ruoli.

4. Per utilizzare il file creato per aggiornare la policy di attendibilità, eseguire il comando seguente:

   - aws iam update-assume-role-policy

Per consentire agli utenti di un account esterno attendibile di utilizzare il ruolo (AWS CLI)

Per ulteriori informazioni e dettagli su questa procedura, consultare Concedere le autorizzazioni agli utenti per cambiare ruoli.

1. Creare un file JSON contenente una policy di autorizzazione che concede le autorizzazioni ad assumere il ruolo. La policy seguente contiene ad esempio le autorizzazioni minime necessarie:

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": {
           "Effect": "Allow",
           "Action": "sts:AssumeRole",
           "Resource": "arn:aws:iam::111122223333:role/ROLE-NAME"
       }
   }
   ```

   Sostituire l'ARN nell'istruzione con l'ARN del ruolo che l'utente può assumere.

2. Esegui il comando seguente per caricare il file JSON contenente la policy di attendibilità in IAM:

   - aws iam create-policy

L'output di questo comando include l'ARN della policy. Prendere nota di questo ARN, perché sarà necessario in una fase successiva.

3. Stabilire a quale utente o gruppo collegare la policy. Se non si conosce il nome dell'utente o del gruppo desiderato, usare uno dei comandi seguenti per elencare gli utenti o i gruppi nell'account:

   - aws iam list-users
   - aws iam list-groups

4. Usare uno dei comandi seguenti per collegare la policy creata nel passaggio precedente all'utente o al gruppo:

   - aws iam attach-user-policy
   - aws iam attach-group-policy

Aggiornamento di una politica di fiducia del ruolo (AWS API)

Puoi utilizzare l' AWS API per modificare chi può assumere un ruolo.

Per modificare una politica di fiducia del ruolo (AWS API)

1. (Facoltativo) Se non si conosce il nome del ruolo che si desidera modificare, chiamare l'operazione seguente per elencare i ruoli nell'account:

   - ListRoles

2. (Facoltativo) Per visualizzare la policy di affidabilità corrente per un ruolo, chiamare l'operazione seguente:

   - GetRole

3. Per modificare le entità principali attendibili che possono accedere al ruolo, creare un file di testo con la policy di affidabilità aggiornata. È possibile usare qualsiasi editor di testo per creare la policy.

   Ad esempio, la policy di attendibilità seguente illustra come fare riferimento a due Account AWS nell'elemento `Principal`. Ciò consente agli utenti che si trovano all'interno Account AWS di due utenti separati di assumere questo ruolo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {"AWS": [
            "arn:aws:iam::111122223333:root",
            "arn:aws:iam::444455556666:root"
        ]},
        "Action": "sts:AssumeRole"
    }
}
```

Se specifichi un'entità principale in un altro account, l'aggiunta di un account alla policy di attendibilità di un ruolo è solo una parte della creazione della relazione di trust tra più account. Per impostazione predefinita, gli utenti negli account attendibili non possono assumere il ruolo. L'amministratore del nuovo account attendibile deve concedere agli utenti l'autorizzazione ad assumere il ruolo. A tale scopo, l'amministratore deve creare o modificare una policy collegata all'utente per consentire all'utente di accedere all'operazione `sts:AssumeRole`. Per ulteriori informazioni, consultare la procedura seguente o Concedere le autorizzazioni agli utenti per cambiare ruoli.

4. Per utilizzare il file creato per aggiornare la policy di attendibilità, chiamare l'operazione seguente:

   - UpdateAssumeRolePolicy

Per consentire agli utenti di un account esterno attendibile di utilizzare il ruolo (AWS API)

Per ulteriori informazioni e dettagli su questa procedura, consultare Concedere le autorizzazioni agli utenti per cambiare ruoli.

1. Creare un file JSON contenente una policy di autorizzazione che concede le autorizzazioni ad assumere il ruolo. La policy seguente contiene ad esempio le autorizzazioni minime necessarie:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Resource": "arn:aws:iam::111122223333:role/ROLE-NAME"
    }
}
```

Sostituire l'ARN nell'istruzione con l'ARN del ruolo che l'utente può assumere.

2. Chiama l'operazione seguente per caricare il file JSON contenente la policy di attendibilità in IAM:

   - CreatePolicy

   L'output di questa operazione include l'ARN della policy. Prendere nota di questo ARN, perché sarà necessario in una fase successiva.

3. Stabilire a quale utente o gruppo collegare la policy. Se non si conosce il nome dell'utente o del gruppo desiderato, chiamare una delle operazioni seguenti per elencare gli utenti o i gruppi nell'account:

   - ListUsers
   - ListGroups

4. Chiamare una delle operazioni seguenti per collegare la policy creata nel passaggio precedente all'utente o al gruppo:

   - API: AttachUserPolicy
   - AttachGroupPolicy

Aggiornamento delle autorizzazioni per un ruolo

Utilizza le seguenti procedure per aggiornare le policy e i limiti delle autorizzazioni di un ruolo.

Prerequisito: Visualizzazione dell'accesso per il ruolo

Prima di modificare le autorizzazioni per un ruolo, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Aggiornamento di una policy delle autorizzazioni per un ruolo

Per modificare le autorizzazioni permesse dal ruolo, modifica la policy (o le policy) di autorizzazioni del ruolo. Non è possibile modificare la policy di autorizzazione per un ruolo collegato ai servizi in IAM. Potresti essere in grado di modificare la policy di autorizzazione all'interno del servizio che dipende dal ruolo. Per controllare se un servizio supporta questa funzionalità, consulta AWS servizi supportati da IAM e individua i servizi che hanno Sì nella colonna Ruoli collegati ai servizi. Scegliere Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Aggiornamento di una policy delle autorizzazioni del ruolo (console)

Per modificare le autorizzazioni permesse da un ruolo (console)

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Ruoli.
3. Selezionare il nome del ruolo da modificare e la scheda Permissions (Autorizzazioni).
4. Esegui una delle seguenti operazioni:

   - Per modificare una policy gestita dal cliente esistente, selezionare il nome della policy e scegliere Edit policy (Modifica policy).

> **Note**
>
> Non è possibile modificare una politica AWS gestita. AWS le politiche gestite vengono visualizzate con l' AWS icona ().
>
> Per ulteriori informazioni sulla differenza tra politiche AWS gestite e politiche gestite dal cliente, vedere Policy gestite e policy inline.

- Per collegare una policy gestita esistente al ruolo, scegli Add permissions (Aggiungi autorizzazioni) e quindi Attach policies (Collega policy).
- Per modificare una policy inline esistente, espandi la policy e scegli Edit (Modifica).
- Per integrare una nuova policy inline, scegli Add permissions (Aggiungi autorizzazioni), quindi Create inline policy (Crea policy inline).
- Per rimuovere una policy esistente dal ruolo, seleziona la casella di controllo accanto al nome della policy, quindi scegli Rimuovi.

Aggiornamento di una policy di autorizzazione del ruolo (AWS CLI)

Per modificare le autorizzazioni permesse dal ruolo, modifica la policy (o le policy) di autorizzazioni del ruolo. Non è possibile modificare la policy di autorizzazione per un ruolo collegato ai servizi in IAM. Potresti essere in grado di modificare la policy di autorizzazione all'interno del servizio che dipende dal ruolo. Per controllare se un servizio supporta questa funzionalità, consulta AWS servizi supportati da IAM e individua i servizi che hanno Sì nella colonna Ruoli collegati ai servizi. Scegliere Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Per modificare le autorizzazioni permesse da un ruolo (AWS CLI)

1. (Facoltativo) Per visualizzare le autorizzazioni correnti associate a un ruolo, eseguire i comandi seguenti:
   1. Lo scopo è quello list-role-policies di elencare le politiche in linea
   2. aws mira list-attached-role-policies a elencare le politiche gestite
2. Il comando per aggiornare le autorizzazioni per il ruolo varia a seconda del fatto che si aggiorni una policy gestita o una policy inline.

   Per aggiornare una policy gestita, eseguire il comando seguente per creare una nuova versione della policy gestita:

   - era io create-policy-version

   Per aggiornare una policy inline, eseguire il comando seguente:

   - era io put-role-policy

Aggiornamento di una politica di autorizzazione dei ruoli (AWS API)

Per modificare le autorizzazioni permesse dal ruolo, modifica la policy (o le policy) di autorizzazioni del ruolo. Non è possibile modificare la policy di autorizzazione per un ruolo collegato ai servizi in IAM. Potresti essere in grado di modificare la policy di autorizzazione all'interno del servizio che dipende dal ruolo. Per controllare se un servizio supporta questa funzionalità, consulta AWS servizi supportati da IAM e individua i servizi che hanno Sì nella colonna Ruoli collegati ai servizi. Scegliere Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Per modificare le autorizzazioni consentite da un ruolo (API)AWS

1. (Facoltativo) Per visualizzare le autorizzazioni correnti associate a un ruolo, chiamare le operazioni seguenti:
   1. ListRolePoliciesper elencare le politiche in linea
   2. ListAttachedRolePoliciesper elencare le politiche gestite
2. L'operazione per aggiornare le autorizzazioni per il ruolo varia a seconda del fatto che si aggiorni una policy gestita o una policy inline.

   Per aggiornare una policy gestita, chiamare l'operazione seguente per creare una nuova versione della policy gestita:

   - CreatePolicyVersion

   Per aggiornare una policy inline, chiamare l'operazione seguente:

   - PutRolePolicy

Aggiornamento del limite delle autorizzazioni per un ruolo

Per modificare il numero massimo di autorizzazioni consentite per un ruolo, modifica il limite delle autorizzazioni del ruolo.

Aggiornamento di un limite delle autorizzazioni del ruolo (console)

Per modificare la policy utilizzata per impostare il limite delle autorizzazioni per un ruolo

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Scegli il nome del ruolo con il limite delle autorizzazioni che desideri modificare.
4. Scegli la scheda Autorizzazioni. Se necessario, aprire la sezione Permissions boundary (Limite delle autorizzazioni) e selezionare Change boundary (Modifica limite).
5. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
6. Selezionare Change boundary (Modifica limite).

   Le modifiche non verranno applicate fino alla volta successiva in cui qualcuno assume questo ruolo.

Aggiornamento di un limite delle autorizzazioni del ruolo (AWS CLI)

Per modificare la policy gestita utilizzata per impostare il limite delle autorizzazioni per un ruolo (AWS CLI)

1. (Facoltativo) Per visualizzare il limite delle autorizzazioni corrente per un ruolo, eseguire il comando seguente:

   - aws iam get-role
2. Per usare un'altra policy gestita per aggiornare il limite delle autorizzazioni per un ruolo, eseguire il comando seguente:

   - era iam put-role-permissions-boundary

   Un ruolo può avere solo una policy gestita impostata come limite delle autorizzazioni. Modificando il limite delle autorizzazioni è possibile modificare il numero massimo di autorizzazioni consentite per un ruolo.

Aggiornamento di un limite di autorizzazioni di ruolo (API)AWS

Per modificare la politica gestita utilizzata per impostare il limite delle autorizzazioni per un ruolo (API)AWS

1. (Facoltativo) Per visualizzare il limite delle autorizzazioni corrente per un ruolo, richiamare l'operazione seguente:
   - GetRole
2. Per usare un'altra policy gestita per aggiornare il limite delle autorizzazioni per un ruolo, chiamare l'operazione seguente:
   - PutRolePermissionsBoundary

   Un ruolo può avere solo una policy gestita impostata come limite delle autorizzazioni. Modificando il limite delle autorizzazioni è possibile modificare il numero massimo di autorizzazioni consentite per un ruolo.

Aggiornamento delle impostazioni per un ruolo

Utilizza le seguenti procedure per aggiornare la descrizione di un ruolo o modificare la durata massima della sessione per un ruolo.

Aggiornamento della descrizione di un ruolo

Per cambiare la descrizione del ruolo, modifica il testo di descrizione.

Aggiornamento della descrizione di un ruolo (console)

Per modificare la descrizione di un ruolo (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli).
3. Scegliere il nome del ruolo da modificare.
4. Nella sezione Summary (Riepilogo), scegli Edit (Modifica).
5. Digita una nuova descrizione nella casella e scegli Save changes (Salva modifiche).

Aggiornamento della descrizione di un ruolo (AWS CLI)

Per modificare la descrizione di un ruolo (AWS CLI)

1. (Facoltativo) Per visualizzare la descrizione corrente di un ruolo, eseguire il comando seguente:
   - aws iam get-role
2. Per aggiornare la descrizione di un ruolo, eseguire il comando seguente con il parametro relativo alla descrizione:
   - aws iam update-role

Aggiornamento di una descrizione del ruolo (AWS API)

Per modificare la descrizione di un ruolo (AWS API)

1. (Facoltativo) Per visualizzare la descrizione corrente per un ruolo, chiamare l'operazione seguente:
   - GetRole
2. Per aggiornare la descrizione di un ruolo, chiamare l'operazione seguente con il parametro relativo alla descrizione:
   - UpdateRole

Aggiornamento della durata massima della sessione per un ruolo

Per specificare l'impostazione della durata massima della sessione per i ruoli che vengono assunti utilizzando la console AWS CLI, l'o l' AWS API, modifica il valore di impostazione della durata massima della sessione. Questa impostazione può avere un valore compreso tra 1 ora e 12 ore. Se non specifichi un valore, viene applicata l'impostazione predefinita massima di 1 ora. Questa impostazione non limita le sessioni assunte dai AWS servizi.

Aggiornamento della durata massima della sessione del ruolo (console)

Per modificare l'impostazione della durata massima della sessione per i ruoli assunti utilizzando la console o l' AWS API (console) AWS CLI

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli).
3. Scegliere il nome del ruolo da modificare.
4. Nella sezione Summary (Riepilogo), scegli Edit (Modifica).
5. In Maximum session duration (Durata massima della sessione), scegli un valore. In alternativa, scegli Custom duration (Durata personalizzata) e inserisci un valore (in secondi).
6. Scegli Save changes (Salva modifiche).

   Le modifiche non verranno applicate fino alla volta successiva in cui qualcuno assume questo ruolo. Per informazioni su come revocare le sessioni esistenti per il ruolo, consultare Revocare le credenziali di sicurezza temporanee per i ruoli IAM.

In Console di gestione AWS, per impostazione predefinita, le sessioni utente IAM durano 12 ore. Agli utenti IAM viene che cambiano ruoli nella console viene concessa la durata massima della sessione del ruolo o il tempo rimanente nella sessione dell'utente IAM, a seconda di quale sia minore.

Chiunque assuma il ruolo dell' AWS API AWS CLI or può richiedere una sessione più lunga, fino a questo massimo. L'impostazione `MaxSessionDuration` determina la durata massima della sessione del ruolo che può essere richiesta.

- Per specificare la durata di una sessione AWS CLI utilizzando il `duration-seconds` parametro. Per ulteriori informazioni, consulta Passa a un ruolo IAM (AWS CLI).
- Per specificare la durata di una sessione utilizzando l' AWS API, utilizza il `DurationSeconds` parametro. Per ulteriori informazioni, consulta Passa a un ruolo IAM (AWS API).

Aggiornamento della durata massima della sessione di ruolo (AWS CLI)

> **Note**
>
> Chiunque assuma il ruolo dell'API AWS CLI o può utilizzare il parametro `duration-seconds` CLI o `DurationSeconds` il parametro API per richiedere una sessione più lunga. L'impostazione `MaxSessionDuration` determina la durata massima della sessione del ruolo che può essere richiesta usando il parametro `DurationSeconds`. Se gli utenti non specificano un valore per il parametro `DurationSeconds` le loro credenziali di sicurezza rimangono valide per un'ora.

Per modificare l'impostazione della durata massima della sessione per i ruoli assunti utilizzando il AWS CLI ()AWS CLI

1. (Facoltativo) Per visualizzare l'impostazione della durata massima della sessione corrente per un ruolo, eseguire il comando seguente:

   - aws iam get-role

2. Per aggiornare l'impostazione della durata massima della sessione di un ruolo, eseguire il comando seguente con il parametro `max-session-duration` della CLI oppure il parametro API `MaxSessionDuration`:

   - aws iam update-role

   Le modifiche non verranno applicate fino alla volta successiva in cui qualcuno assume questo ruolo. Per informazioni su come revocare le sessioni esistenti per il ruolo, consultare Revocare le credenziali di sicurezza temporanee per i ruoli IAM.

Aggiornamento della durata massima della sessione di ruolo (AWS API)

> **Note**
>
> Chiunque assuma il ruolo dell'API AWS CLI o può utilizzare il parametro `duration-seconds` CLI o `DurationSeconds` il parametro API per richiedere una sessione più lunga. L'impostazione `MaxSessionDuration` determina la durata massima della sessione del ruolo che può essere richiesta usando il parametro `DurationSeconds`. Se gli utenti non specificano un valore per il parametro `DurationSeconds` le loro credenziali di sicurezza rimangono valide per un'ora.

Per modificare l'impostazione della durata massima della sessione per i ruoli assunti utilizzando l'API (API)AWS

1. (Facoltativo) Per visualizzare l'impostazione della durata massima della sessione corrente per un ruolo, chiamare l'operazione seguente:

   - GetRole

2. Per aggiornare l'impostazione della durata massima della sessione di un ruolo, chiamare l'operazione seguente con il parametro `max-session-duration` della CLI oppure il parametro API `MaxSessionDuration`:

   - UpdateRole

   Le modifiche non verranno applicate fino alla volta successiva in cui qualcuno assume questo ruolo. Per informazioni su come revocare le sessioni esistenti per il ruolo, consultare Revocare le credenziali di sicurezza temporanee per i ruoli IAM.

Eliminare i ruoli o i profili delle istanze

Se un ruolo non è più necessario, si consiglia di eliminare il ruolo e le autorizzazioni associate. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente.

Se il ruolo è stato associato a un'istanza EC2, è anche possibile rimuovere il ruolo dal profilo dell'istanza e quindi eliminare il profilo dell'istanza.

> **Warning**
>
> Assicurati di non avere istanze Amazon EC2 in esecuzione con il ruolo o il profilo di istanza che stai per eliminare. L'eliminazione di un ruolo o di un profilo di istanza associato a un'istanza in esecuzione interrompe tutte le applicazioni in esecuzione sull'istanza.

Se si preferisce non eliminare definitivamente un ruolo, è possibile disabilitarlo. A tale scopo, modifica le policy del ruolo e quindi revoca tutte le sessioni correnti. Ad esempio, potresti aggiungere una policy al ruolo che nega l'accesso a tutti. AWSÈ inoltre possibile modificare i criteri di attendibilità per negare l'accesso a tutti coloro che tentano di assumere il ruolo. Per ulteriori informazioni sull'avvio delle sessioni, consulta Revocare le credenziali di sicurezza temporanee per i ruoli IAM.

Argomenti

- Visualizzazione dell'accesso del ruolo
- Eliminazione del ruolo collegato ai servizi
- Eliminazione di un ruolo IAM (console)
- Eliminazione di un ruolo IAM (AWS CLI)

- Eliminazione di un ruolo IAM (API)AWS
- Informazioni correlate

Visualizzazione dell'accesso del ruolo

Prima di eliminare un ruolo, è opportuno esaminare quando è stato utilizzato per l'ultima volta. È possibile eseguire questa operazione utilizzando l' Console di gestione AWS AWS CLI, l'o l' AWS API. È consigliabile visualizzare queste informazioni per non privare dell'accesso qualcuno che utilizza il ruolo.

La data dell'ultima attività del ruolo potrebbe non corrispondere all'ultima data riportata nella scheda Ultimo accesso. La scheda Ultimo accesso riporta l'attività solo per i servizi consentiti dalle policy di autorizzazione del ruolo. La data dell'ultima attività del ruolo include l'ultimo tentativo di accesso a qualsiasi servizio in AWS.

> **Note**
>
> Il periodo di monitoraggio dei dati per l'ultima attività di un ruolo e i dati di Ultimo accesso sono per gli ultimi 400 giorni. Questo periodo può essere abbreviato se la regione ha iniziato a supportare queste funzionalità nell'ultimo anno. Il ruolo potrebbe essere stato utilizzato più di 400 giorni fa. Per ulteriori informazioni sul periodo di monitoraggio, consulta Dove AWS tiene traccia delle ultime informazioni a cui si accede.

Per visualizzare la data di ultimo utilizzo di un ruolo (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Individua la riga del ruolo con l'attività che si desidera visualizzare. È possibile utilizzare il campo di ricerca per restringere i risultati. Visualizzare la colonna Last activity (Ultima attività) per visualizzare il numero di giorni trascorsi dalla data di ultimo utilizzo del ruolo. Se il ruolo non è stato utilizzato entro il periodo di monitoraggio, nella tabella viene visualizzato None (Nessuno).
4. Scegliere il nome del ruolo per visualizzare ulteriori informazioni. La pagina Riepilogo del ruolo include anche Ultima attività, che visualizza la data dell'ultimo utilizzo del ruolo. Se il ruolo non è stato utilizzato negli ultimi 400 giorni, Last activity (Ultima attività) visualizza Not accessed in the tracking period (Nessun accesso nel periodo di monitoraggio).

Per visualizzare la data di ultimo utilizzo di un ruolo (AWS CLI)

`aws iam get-role` - Eseguire questo comando per restituire le informazioni su un ruolo, incluso l'oggetto `RoleLastUsed`. Questo oggetto contiene `LastUsedDate` e la `Region` in cui il ruolo è stato utilizzato per l'ultima volta. Se `RoleLastUsed` è presente ma non contiene un valore, il ruolo non è stato utilizzato entro il periodo di monitoraggio.

Per visualizzare quando un ruolo è stato utilizzato l'ultima volta (AWS API)

`GetRole` - Chiamare questa operazione per restituire le informazioni su un ruolo, incluso l'oggetto `RoleLastUsed`. Questo oggetto contiene `LastUsedDate` e la `Region` in cui il ruolo è stato utilizzato per l'ultima volta. Se `RoleLastUsed` è presente ma non contiene un valore, il ruolo non è stato utilizzato entro il periodo di monitoraggio.

Eliminazione del ruolo collegato ai servizi

Il metodo utilizzato per eliminare un ruolo collegato ai servizi dipende dal servizio. In alcuni casi, non devi eliminare manualmente un ruolo collegato ai servizi. Ad esempio, quando completi un'operazione specifica (come eliminare una risorsa) nel servizio, il servizio potrebbe eliminare il ruolo collegato ai servizi per te. In altri casi, il servizio può supportare l'eliminazione di un ruolo collegato ai servizi manualmente dalla console del servizio, dall'API o dalla AWS CLI.

Consulta la documentazione relativa al ruolo collegato al servizio del servizio collegato per ulteriori informazioni su come eliminare il ruolo. Puoi visualizzare i ruoli collegati ai servizi nell'account visitando la pagina Ruoli IAM nella console. I ruoli collegati al servizio vengono visualizzati con l'indicazione (Service-linked role) (Ruolo collegato al servizio) nella colonna Trusted entities (Entità attendibili) della tabella. Un banner nella pagina Riepilogo del ruolo indica anche che il ruolo è un ruolo collegato ai servizi.

Se il servizio non include la documentazione per l'eliminazione del ruolo collegato al servizio, puoi utilizzare la console IAM o l'API per eliminare il ruolo. AWS CLI

Eliminazione di un ruolo IAM (console)

Quando utilizzi il Console di gestione AWS per eliminare un ruolo, IAM scollega automaticamente le policy gestite associate al ruolo. Inoltre elimina automaticamente anche le policy in linea associate al ruolo e qualsiasi profilo dell'istanza Amazon EC2 che contiene il ruolo.

> **Important**
>
> In alcuni casi, un ruolo potrebbe essere associato a un profilo dell'istanza Amazon EC2 e il ruolo e il profilo dell'istanza potrebbero avere lo stesso nome. In tal caso puoi utilizzare il Console di gestione AWS per eliminare il ruolo e il profilo dell'istanza. Questo collegamento avviene automaticamente per i ruoli e i profili delle istanze creati nella console. Se hai creato il ruolo da AWS CLI, Tools for Windows PowerShell o dall' AWS API, il ruolo e il profilo dell'istanza potrebbero avere nomi diversi. In questo caso non è possibile utilizzare la console per eliminarli. È invece necessario utilizzare Tools for Windows PowerShell o AWS API per rimuovere prima il ruolo dal profilo dell'istanza. AWS CLIÈ quindi necessario eseguire un passaggio distinto per eliminare il ruolo.

Per eliminare un ruolo (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegliere Roles (Ruoli), quindi selezionare la casella di controllo accanto al nome del ruolo che si desidera eliminare.
3. Nella parte superiore della pagina, scegli Elimina.
4. Nella finestra di dialogo di conferma, esamina le informazioni relative all'ultimo accesso, che mostrano l'ultima volta che ciascuno dei ruoli selezionati ha effettuato l'ultimo accesso a un AWS servizio. In questo modo potrai verificare se il ruolo è attualmente attivo. Se desideri procedere, inserisci il nome del ruolo nel campo di immissione testo e seleziona Elimina. Se sei sicuro, puoi procedere con l'eliminazione anche se l'ultimo accesso ai dati del servizio è ancora in fase di caricamento.

> **Note**
>
> Non è possibile utilizzare la console per eliminare un profilo dell'istanza, a meno che non abbia lo stesso nome del ruolo. Il profilo dell'istanza viene eliminato come parte del processo di eliminazione di un ruolo descritto nella procedura precedente. Per eliminare un profilo di istanza senza eliminare anche il ruolo, è necessario utilizzare l' AWS API AWS CLI o. Per ulteriori informazioni, consultare le sezioni indicate di seguito.

Eliminazione di un ruolo IAM (AWS CLI)

Quando si utilizza il AWS CLI per eliminare un ruolo, è necessario innanzitutto eliminare le politiche in linea associate al ruolo. È inoltre necessario scollegare le policy gestite associate al ruolo. Se desideri eliminare il profilo dell'istanza associato che contiene il ruolo, devi eliminarlo separatamente.

Per eliminare un ruolo (AWS CLI)

1. Se non conosci il nome del ruolo da eliminare, immetti il comando seguente per elencare i ruoli nell'account:

   ```
   aws iam list-roles
   ```

   L'elenco include l'Amazon Resource Name (ARN) di ogni ruolo. Per fare riferimento ai ruoli con i comandi della CLI utilizza il nome del ruolo, non l'ARN. Ad esempio, per fare riferimento a un ruolo il cui ARN è `arn:aws:iam::123456789012:role/myrole`, puoi usare **`myrole`**.
2. Rimuovi il ruolo da tutti i profili delle istanze a cui è associato.

   a. Per elencare tutti i profili delle istanze cui è associato il ruolo, immetti il seguente comando:

      ```
      aws iam list-instance-profiles-for-role --role-name role-name
      ```

   b. Per rimuovere il ruolo da un profilo dell'istanza, immetti il seguente comando per ogni profilo dell'istanza:

      ```
      aws iam remove-role-from-instance-profile --instance-profile-name instance-profile-name --role-name role-name
      ```
3. Elimina tutte le policy associate al ruolo.

   a. Per elencare tutte le policy inline presenti nel ruolo, immetti il seguente comando:

      ```
      aws iam list-role-policies --role-name role-name
      ```

   b. Per eliminare ogni policy inline dal ruolo, immetti il seguente comando per ogni policy:

      ```
      aws iam delete-role-policy --role-name role-name --policy-name policy-name
      ```

   c. Per elencare tutte le policy gestite collegate al ruolo, immetti il seguente comando:

```
aws iam list-attached-role-policies --role-name role-name
```

d. Per scollegare ogni policy gestita dal ruolo, immetti il seguente comando per ogni policy:

```
aws iam detach-role-policy --role-name role-name --policy-arn policy-arn
```

4. Immetti il seguente comando per eliminare il ruolo:

```
aws iam delete-role --role-name role-name
```

5. Se non prevedi di riutilizzare i profili delle istanze associati al ruolo, puoi immettere il seguente comando per eliminarli:

```
aws iam delete-instance-profile --instance-profile-name instance-profile-name
```

Eliminazione di un ruolo IAM (API)AWS

Se utilizzi l'API IAM per eliminare un ruolo, devi prima eliminare le policy inline associate al ruolo. È inoltre necessario scollegare le policy gestite associate al ruolo. Se desideri eliminare il profilo dell'istanza associato che contiene il ruolo, devi eliminarlo separatamente.

Per eliminare un ruolo (AWS API)

1. Per elencare tutti i profili di istanza a cui è associato un ruolo, chiama ListInstanceProfilesForRole.

   Per rimuovere il ruolo da un profilo di istanza, chiama RemoveRoleFromInstanceProfile. È necessario passare il nome del ruolo e il nome del profilo di istanza.

   Se non intendi riutilizzare un profilo di istanza associato al ruolo, chiama DeleteInstanceProfileper eliminarlo.

2. Per elencare tutte le politiche in linea per un ruolo, chiama. ListRolePolicies

   Per eliminare le politiche in linea associate al ruolo, chiama. DeleteRolePolicy Devi passare il nome del ruolo e il nome della policy inline.

3. Per elencare tutte le politiche gestite associate a un ruolo, chiama ListAttachedRolePolicies.

Per scollegare le politiche gestite allegate al ruolo, chiama DetachRolePolicy. Devi passare il nome del ruolo e l'ARN della policy gestita.

4. Chiama DeleteRoleper eliminare il ruolo.

Informazioni correlate

Per informazioni generali sui profili delle istanze, consulta Usare profili dell'istanza.

Per informazioni generali sui ruoli collegati al servizio, consultare Creare un ruolo collegato ai servizi.

Metodi per assumere un ruolo

Prima che un utente, applicazione o servizio possa utilizzare un ruolo che è stato creato, è necessario concedere le autorizzazioni per passare al ruolo. È possibile utilizzare qualsiasi policy collegata a gruppi o utenti per concedere le autorizzazioni necessarie. Dopo aver concesso le autorizzazioni, l'utente può assumere un ruolo dagli Console di gestione AWS strumenti per Windows PowerShell, da AWS Command Line Interface (AWS CLI) e dall'`AssumeRole`API.

> **Important**
>
> Se crei un ruolo a livello programmatico anziché nella console IAM, hai l'opzione per aggiungere un `Path` con un massimo di 512 caratteri in aggiunta a `RoleName`, che può contenere fino a 64 caratteri. Tuttavia, se intendete utilizzare un ruolo con la funzione Cambia ruolo in Console di gestione AWS, la combinazione `Path` `RoleName` non può superare i 64 caratteri.

Il metodo utilizzato per assumere il ruolo determina chi può assumere il ruolo e per quanto tempo la sessione del ruolo della sessione può durare. Quando utilizzi `AssumeRole*` Operazioni API, il ruolo IAM assunto è la risorsa. L'utente o il ruolo che chiama le operazioni API `AssumeRole*` è il principale.

Nella tabella seguente vengono confrontati i metodi per assumere i ruoli.

Metodo per assumere il ruolo	Chi può assumere il ruolo	Metodo per specificare il ciclo di vita delle credenziali	Ciclo di vita delle credenziali (minimo | massimo | predefinito)
Console di gestione AWS	Utente o ruoli[1] (cambiando ruoli)	Durata massima sessione nella pagina di riepilogo Ruolo	15 min | Impostazione durata massima sessione[2] | 1 ora
assume-role CLI oppure operazione API AssumeRole	Utente o ruolo[1]	CLI duration-seconds oppure parametro API DurationSeconds	15 min | Impostazione durata massima sessione[2] | 1 ora
assume-role-with-saml CLI oppure operazione API AssumeRoleWithSAML	Tutti gli utenti autenticati utilizzando SAML	CLI duration-seconds oppure parametro API DurationSeconds	15 min | Impostazione durata massima sessione[2] | 1 ora
assume-role-with-web-identity CLI oppure operazione API AssumeRoleWithWebIdentity	Tutti gli utenti autenticati che utilizzano un provider OIDC	CLI duration-seconds oppure parametro API DurationSeconds	15 min | Impostazione durata massima sessione[2] | 1 ora
Console URL creata con AssumeRole	Utente o ruolo	Parametro HTML SessionDu	15 min | 12 ore | 1 ora

Metodo per assumere il ruolo	Chi può assumere il ruolo	Metodo per specificare il ciclo di vita delle credenziali	Ciclo di vita delle credenziali (minimo | massimo | predefinito)
		`ration` nell'URL	
Console URL creata con `AssumeRoleWithSAML`	Tutti gli utenti autenticati utilizzando SAML	Parametro HTML `SessionDuration` nell'URL	15 min | 12 ore | 1 ora
Console URL creata con `AssumeRoleWithWebIdentity`	Tutti gli utenti autenticati che utilizzano un provider OIDC	Parametro HTML `SessionDuration` nell'URL	15 min | 12 ore | 1 ora

[1] L'utilizzo delle credenziali di un ruolo per assumere un ruolo diverso si chiama concatenazione dei ruoli. Quando si utilizza la concatenazione dei ruoli, la durata della sessione del ruolo è limitata a un'ora. Questo vale per Console di gestione AWS il cambio di ruolo e le operazioni API. AWS CLI Questa limitazione non si applica all'assunzione iniziale di un ruolo in base alle credenziali utente o alle applicazioni in esecuzione su EC2 istanze Amazon che utilizzano profili di istanza.

[2] Questa impostazione può avere un valore compreso tra 1 ora e 12 ore. Per informazioni sulla modifica dell'impostazione della durata massima della sessione, consulta Gestione del ruolo IAM. Questa impostazione determina la durata massima della sessione che è possibile richiedere quando si ottengono le credenziali del ruolo. Ad esempio, quando utilizzi le operazioni dell'AssumeRoleAPI* per assumere un ruolo, puoi specificare la durata della sessione utilizzando il parametro. `DurationSeconds` Utilizzare questo parametro per specificare la durata della sessione del ruolo da 900 secondi (15 minuti) fino all'impostazione di durata massima della sessione per il ruolo. Agli utenti IAM che cambiano ruoli nella console viene concessa la durata massima della sessione o il tempo rimanente nella sessione dell'utente, a seconda di quale sia minore. Si supponga di impostare una durata massima di 5 ore su un ruolo. Un utente IAM che è stato collegato alla console per 10 ore (rispetto al valore massimo predefinito di 12) può cambiare ruolo. La durata della

sessione di ruolo disponibile è di 2 ore. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Aggiornamento della durata massima della sessione per un ruolo più avanti su questa pagina.

> **Note**
>
> - L'impostazione di durata massima delle sessioni non limita le sessioni assunte dai servizi AWS .
> - Le credenziali del ruolo Amazon EC2 IAM non sono soggette alla durata massima della sessione configurata nel ruolo.
> - Per consentire agli utenti di assumere nuovamente il ruolo corrente all'interno di una sessione di ruolo, specificare il ruolo ARN o Account AWS ARN come principale nella politica di attendibilità dei ruoli. Servizi AWS che forniscono risorse di elaborazione come Amazon EC2, Amazon ECS, Amazon EKS e Lambda forniscono credenziali temporanee e le aggiornano automaticamente. Ciò garantisce di disporre sempre di un set di credenziali valido. Per questi servizi, non è necessario riassumere il ruolo attuale per ottenere credenziali temporanee. Tuttavia, se intendi passare tag di sessione o una Policy di sessione, devi riassumere il ruolo attuale. Per informazioni su come modificare una politica di attendibilità dei ruoli per aggiungere il ruolo principale ARN o Account AWS ARN, vedere. Aggiornamento di una policy di attendibilità del ruolo

Argomenti

- Passare da un utente a un ruolo IAM (console)
- Passa a un ruolo IAM (AWS CLI)
- Passa a un ruolo IAM (Strumenti per Windows PowerShell)
- Passa a un ruolo IAM (AWS API)
- Utilizzare un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2
- Usare profili dell'istanza

Passare da un utente a un ruolo IAM (console)

Puoi cambiare ruolo quando accedi come utente IAM, utente in IAM Identity Center, ruolo o SAML-federated ruolo federato con identità web. Un ruolo specifica un set di autorizzazioni che puoi

utilizzare per accedere AWS alle risorse di cui hai bisogno. Tuttavia, non si effettua l'accesso a un ruolo, ma una volta effettuato l'accesso come utente IAM è possibile passare a un ruolo IAM. Ciò consente di accantonare temporaneamente le autorizzazioni utente originali e usufruire invece delle autorizzazioni assegnate al ruolo. Il ruolo può trovarsi nel tuo account o in qualsiasi altro Account AWS. Per ulteriori informazioni sui ruoli e i relativi vantaggi e su come crearli e configurarli, consulta Ruoli IAM e Creazione di ruoli IAM.

Le autorizzazioni dell'utente e di qualsiasi ruolo a cui si passa non sono cumulative. Un solo set di autorizzazioni è attivo alla volta. Quando passi a un ruolo, lasci temporaneamente le autorizzazioni utente e utilizzi le autorizzazioni assegnate al ruolo. Quando lasci il ruolo, le autorizzazioni utente vengono automaticamente ripristinate.

Quando si cambia ruolo in Console di gestione AWS, la console utilizza sempre le credenziali originali per autorizzare il passaggio. Ad esempio, se passi al RuoloA, IAM utilizza le tue credenziali originali per determinare se è possibile assumere il RuoloA. Se poi si passa al ruolo B mentre si utilizza il ruolo A, vengono AWS comunque utilizzate le credenziali originali per autorizzare il passaggio, non le credenziali per il ruolo A.

> **Note**
>
> Quando accedi come utente in IAM Identity Center, come ruolo o come SAML-federated ruolo federato di identità web, assumi un ruolo IAM all'avvio della sessione. Ad esempio, quando un utente in IAM Identity Center AWS accede al portale di accesso, può scegliere un set di autorizzazioni o accedere a un ruolo assegnato tramite l'account access manager prima di poter accedere alle risorse. AWS

Sessioni come ruolo

Quando cambi ruolo, la Console di gestione AWS sessione dura 1 ora per impostazione predefinita. Le sessioni dell'utente IAM sono 12 ore per impostazione predefinita, mentre per altri utenti è possibile che la durata della sessione sia diversa. Quando cambi ruolo nella console, viene concessa la durata massima della sessione o il tempo rimanente nella sessione dell'utente, a seconda di quale sia minore. Non puoi prolungare la durata della sessione assumendo un ruolo.

Si supponga, ad esempio, che per un ruolo sia impostata una durata massima di sessione di 10 ore. Hai effettuato l'accesso alla console per 8 ore quando decidi di cambiare ruolo. Ci sono 4 ore rimanenti nella sessione utente, quindi la durata della sessione ruolo consentita è di 4 ore e non la

durata massima della sessione di 10 ore. Nella tabella seguente viene illustrato come determinare la durata della sessione per un utente IAM quando si cambia ruolo nella console.

Il tempo rimanente della sessione utente IAM è...	La durata della sessione del ruolo è…		
Meno della durata massima della sessione del ruolo	Tempo rimanente nella sessione utente		
Più della durata massima della sessione del ruolo	Valore della durata massima della sessione		
Uguale alla durata massima della sessione del ruolo	Valore della durata massima della sessione (approssimativo)		

L'utilizzo delle credenziali perché un ruolo assuma un ruolo diverso viene chiamato concatenamento dei ruoli. Quando si utilizza il concatenamento dei ruoli, la durata della sessione è limitata a un'ora, indipendentemente dall'impostazione della durata massima della sessione configurata per i singoli ruoli. Questo vale per Console di gestione AWS il cambio di ruolo e le operazioni API. AWS CLI

> **Note**
>
> Alcune console AWS di servizio possono rinnovare automaticamente la sessione di ruolo alla scadenza senza che tu intraprenda alcuna azione. Alcune potrebbero richiedere di ricaricare la pagina del browser per autenticare nuovamente la sessione.

Considerazioni

- Non puoi cambiare ruolo se accedi come. Utente root dell'account AWS
- Agli utenti deve essere concessa l'autorizzazione a cambiare ruolo in base alla policy. Per istruzioni, consulta Concedere le autorizzazioni agli utenti per cambiare ruoli.
- Non puoi passare da un ruolo Console di gestione AWS a un ruolo che richiede un ExternalId valore. È possibile passare a tale ruolo solo chiamando l'API `AssumeRole` che supporta il parametro `ExternalId`.

Per passare a un ruolo

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Accesso alla Console di gestione AWS nella Guida per l'utente di Accedi ad AWS .
2. In Console di gestione AWS, scegli il tuo nome utente nella barra di navigazione in alto a destra. Di solito ha questo aspetto: ***username*** @ ***account_ID_number_or_alias***.
3. Per cambiare ruolo scegli uno dei seguenti metodi:
   - Scegli Cambia ruolo.
   - Se hai scelto il supporto multisessione, scegli Aggiungi sessione e seleziona Cambia ruolo.

   > **Note**
   >
   > Puoi accedere contemporaneamente fino a un massimo di cinque identità diverse in un unico browser web nella Console di gestione AWS. Per maggiori dettagli, consulta Accesso a più account nella Guida introduttiva alla Console di gestione AWS .

4. Nella pagina Switch Role (Cambia ruolo) inserisci il numero ID dell'account o l'alias dell'account e il nome del ruolo che è stato fornito dall'amministratore.

   > **Note**
   >
   > Se l'amministratore ha creato il ruolo con un percorso, ad esempio `division_abc/subdivision_efg/roleToDoX`, allora è necessario digitare tale percorso completo e il nome nella casella Role (Ruolo). Se digiti solo il nome del ruolo, oppure se il `Path` e il `RoleName` insieme superano 64 caratteri, il passaggio di ruolo fallisce. Si tratta di un

limite dei cookie del browser che memorizzano il nome del ruolo. In questo caso, contatta l'amministratore e chiedi di ridurre le dimensioni del percorso e il nome del ruolo.

5. (Facoltativo) È possibile inserire un nome di visualizzazione e selezionare un colore di visualizzazione che evidenzierà il ruolo nella barra di navigazione della console.

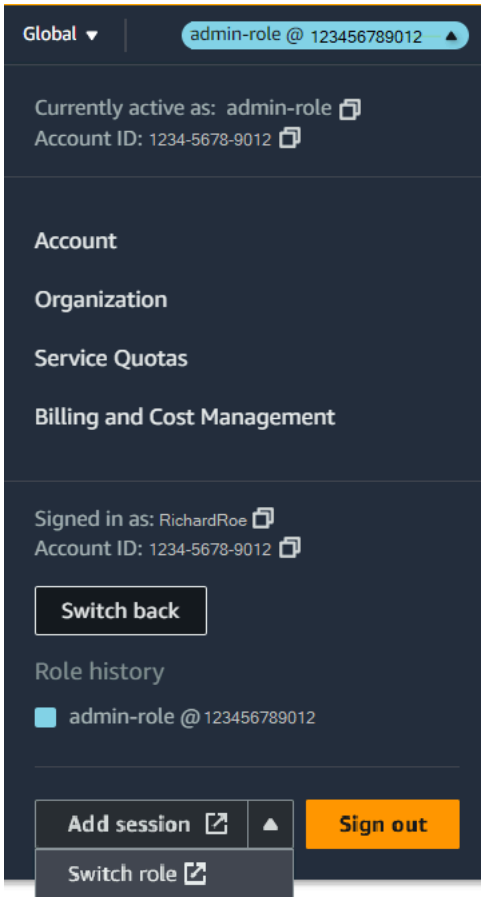
   - Per Nome di visualizzazione inserisci il testo che desideri visualizzare nella barra di navigazione al posto del tuo nome utente quando questo ruolo è attivo. Viene suggerito un nome, in base all'account e alle informazioni del ruolo, ma è possibile modificarlo in base alle proprie esigenze.
   - Per Colore di visualizzazione, seleziona un colore per evidenziare il nome di visualizzazione.

   Il nome e il colore possono aiutarti a ricordare quando questo ruolo è attivo, ciò cambia le tue autorizzazioni. Ad esempio, per un ruolo che ti consente di accedere all'ambiente di test, puoi specificare un Nome di visualizzazione uguale a **`Test`** e selezionare il verde in Colore. Per il ruolo che ti consente di accedere all'ambiente di produzione, puoi specificare un Nome di visualizzazione uguale a **`Production`** e selezionare il rosso in Colore.
6. Seleziona Switch Role (Cambia ruolo). Il nome di visualizzazione e il colore sostituiscono il nome utente nella barra di navigazione ed è possibile iniziare a utilizzare le autorizzazioni concesse dal ruolo.
7. Dopo aver completato le attività che richiedono il ruolo IAM, potrai tornare alla sessione originale. In questo modo verranno rimosse le autorizzazioni aggiuntive fornite dal ruolo e verranno ripristinate le autorizzazioni standard.

   a. Nella console IAM, scegli il Nome di visualizzazione del tuo ruolo sulla barra di navigazione in alto a destra.

   b. Seleziona Torna indietro.

      Ad esempio, supponiamo di aver eseguito l'accesso all'account numero `123456789012` utilizzando il nome utente `Richard`. Dopo aver utilizzato il ruolo `admin-role`, si desidera interrompere l'utilizzo del ruolo e tornare alle autorizzazioni originali. Per smettere di usare il ruolo, scegli admin-role @ 123456789012, quindi scegli Torna indietro.



> **Tip**
>
> Gli ultimi ruoli utilizzati appariranno nel menu. La prossima volta che devi passare a uno di questi ruoli, è sufficiente selezionare il ruolo desiderato. Se il ruolo non è visualizzato nel menu, è sufficiente digitare l'account e le informazioni del ruolo manualmente.

Risorse aggiuntive

- Concedere le autorizzazioni agli utenti per cambiare ruoli
- Concedi a un utente le autorizzazioni per passare un ruolo a un AWS service
- Creazione di un ruolo per fornire le autorizzazioni a un utente IAM
- Crea un ruolo per delegare le autorizzazioni a un AWS service
- Risoluzione dei problemi relativi ai ruoli IAM

Passa a un ruolo IAM (AWS CLI)

Un ruolo specifica un set di autorizzazioni da utilizzare per accedere alle risorse AWS necessarie. In questo senso, è simile a un utente in AWS Identity and Access Management (IAM). Quando effettui l'accesso come utente, ottieni uno specifico set di autorizzazioni. Tuttavia, non effettui l'accesso a un ruolo, ma dopo aver effettuato l'accesso come utente, puoi passare a un ruolo. Ciò consente di accantonare temporaneamente le autorizzazioni utente originali e usufruire invece delle autorizzazioni assegnate al ruolo. Il ruolo può trovarsi nel tuo account o in qualsiasi altro Account AWS. Per ulteriori informazioni sui ruoli e i relativi vantaggi e su come crearli e configurarli, consulta Ruoli IAM e Creazione di ruoli IAM. Per informazioni sui diversi metodi che si possono utilizzare per assumere un ruolo, consulta Metodi per assumere un ruolo.

> **Important**
>
> Le autorizzazioni dell'utente IAM e di qualsiasi ruolo assunto non sono cumulative. Un solo set di autorizzazioni è attivo alla volta. Quando si assume un ruolo, si lascia temporaneamente l'utente precedente o le autorizzazioni del ruolo e si lavora con le autorizzazioni assegnate al ruolo. Quando lasci il ruolo, le autorizzazioni utente vengono automaticamente ripristinate.

Puoi usare un ruolo per eseguire un AWS CLI comando quando hai effettuato l'accesso come utente IAM. Puoi anche utilizzare un ruolo per eseguire un AWS CLI comando quando accedi come utente autenticato esternamente (SAML o OIDC) che sta già utilizzando un ruolo. Inoltre, puoi utilizzare un ruolo per eseguire un comando AWS CLI da un'istanza Amazon EC2 collegata a un ruolo tramite il relativo profilo. Non è possibile assumere un ruolo quando si è effettuato l'accesso come Utente root dell'account AWS.

Concatenamento del ruolo: puoi anche utilizzare la concatenamento dei ruoli che utilizza le autorizzazioni di un ruolo per accedere a un secondo ruolo.

Come impostazione predefinita, la sessione del ruolo dura un'ora. Quando si assume questo ruolo utilizzando le operazioni della CLI `assume-role*`, è possibile specificare un valore per il parametro `duration-seconds`. Questo valore può variare da 900 secondi (15 minuti) fino alla durata massima della sessione per il ruolo. Se cambi ruolo nella console, la durata della sessione è limitata a un massimo di un'ora. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Aggiornamento della durata massima della sessione per un ruolo.

Se si utilizza il concatenamento dei ruoli, la tua durata della sessione è limitata a un massimo di un'ora. Se successivamente utilizzi il parametro `duration-seconds` per fornire un valore superiore a un'ora, l'operazione ha esito negativo.

Scenario di esempio: passaggio a un ruolo di produzione

Immagina di essere un utente IAM per utilizzare l'ambiente di sviluppo. In questo scenario, a volte è necessario utilizzare l'ambiente di produzione nella riga di comando con l'AWS CLI. Disponi già di un set di credenziali con chiave di accesso. Questa può essere la coppia di chiavi di accesso assegnata all'utente IAM standard. Oppure, se hai effettuato l'accesso come principale federato SAML o OIDC, può essere la coppia di chiavi di accesso per il ruolo che ti è stato inizialmente assegnato. Se le tue autorizzazioni attuali ti consentono di assumere un ruolo IAM specifico, puoi identificare quel ruolo in un «profilo» nei file di configurazione. AWS CLI Questo comando viene quindi eseguito con le autorizzazioni del ruolo IAM specificato, non con l'identità originale. Nota che quando specifichi quel profilo in un AWS CLI comando, stai utilizzando il nuovo ruolo. In questa situazione, non puoi utilizzare le autorizzazioni originali nell'account di sviluppo nello stesso momento. Il motivo è che solo un set di autorizzazioni può essere attivo alla volta.

> **Note**
>
> Per motivi di sicurezza, gli amministratori possono esaminare AWS CloudTrail i registri per scoprire chi ha eseguito un'azione in AWS. L'amministratore potrebbe richiedere di specificare una identità di origine o un nome della sessione del ruolo quando si assume il ruolo. Per ulteriori informazioni, consultare `sts:SourceIdentity` e `sts:RoleSessionName`.

Per passare a un ruolo di produzione (AWS CLI)

1. Se non hai mai usato il AWS CLI, devi prima configurare il tuo profilo CLI predefinito. Apri un prompt dei comandi e configura AWS CLI l'installazione per utilizzare la chiave di accesso del tuo utente IAM o del tuo ruolo federato. Per ulteriori informazioni, consulta Configurazione della AWS Command Line Interface nella Guida per l'utente di AWS Command Line Interface .

   Esegui il comando aws configure come riportato di seguito:

   ```
   aws configure
   ```

   Quando viene richiesto, fornire le seguenti informazioni:

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-east-2
Default output format [None]: json
```

2. Creare un nuovo profilo per il ruolo nel file `.aws/config` in Unix o Linux, oppure nel file `C:\Users\USERNAME\.aws\config` in Windows. L'esempio seguente crea un profilo denominato `prodaccess` che passa al ruolo *`ProductionAccessRole`* nell'account `123456789012`. L'ARN del ruolo si ottiene dall'amministratore dell'account che ha creato il ruolo. Quando viene richiamato questo profilo, AWS CLI utilizza le credenziali di `source_profile` per richiedere le credenziali per il ruolo. Per questo motivo, l'identità alla quale viene fatto riferimento come `source_profile` deve disporre delle autorizzazioni `sts:AssumeRole` per il ruolo specificato in `role_arn`.

   ```
   [profile prodaccess]
       role_arn = arn:aws:iam::123456789012:role/ProductionAccessRole
       source_profile = default
   ```

3. Dopo aver creato il nuovo profilo, qualsiasi AWS CLI comando che specifica il parametro `--profile prodaccess` viene eseguito in base alle autorizzazioni associate al ruolo IAM `ProductionAccessRole` anziché all'utente predefinito.

   ```
   aws iam list-users --profile prodaccess
   ```

   Questo comando funziona se le autorizzazioni assegnate a `ProductionAccessRole` permettono di elencare gli utenti nell'account attuale AWS .

4. Per ripristinare le autorizzazioni concesse dalle credenziali originali, eseguire i comandi senza il parametro `--profile`. Quindi AWS CLI torna all'utilizzo delle credenziali nel profilo predefinito, in cui hai configurato. Step 1

Per ulteriori informazioni, consulta Assunzione di un ruolo nella Guida per l'utente di AWS Command Line Interface .

Scenario di esempio: consentire a un ruolo del profilo dell'istanza di passare a un ruolo in un altro account

Immagina di utilizzarne due Account AWS e di voler consentire a un'applicazione in esecuzione su un'istanza Amazon EC2 di eseguire AWS CLI comandi in entrambi gli account. Supponiamo che

l'istanza EC2 esista nell'account 111111111111. Tale istanza include il ruolo del profilo dell'istanza abcd che consente all'applicazione di eseguire attività Amazon S3 di sola lettura nel bucket amzn-s3-demo-bucket1 all'interno dello stesso account 111111111111. Tuttavia, l'applicazione deve anche poter assumere il ruolo tra più account efgh per eseguire attività nell'account 222222222222. A questo scopo, il ruolo del profilo dell'istanza EC2 abcd deve disporre della policy di autorizzazioni seguente:

Policy di autorizzazioni del ruolo ***abcd*** 111111111111 dell'account

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccountLevelS3Actions",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowListAndReadS3ActionOnMyBucket",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1/*",
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
        },
        {
            "Sid": "AllowIPToAssumeCrossAccountRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::222222222222:role/efgh"
```

```
        }
    ]
}
```

Supponiamo che il ruolo tra account `efgh` consenta attività Amazon S3 di sola lettura nel bucket `amzn-s3-demo-bucket2` all'interno dello stesso account `222222222222`. A tale scopo, il ruolo tra account `efgh` deve disporre della seguente policy di autorizzazioni:

Policy di autorizzazioni del ruolo ***efgh*** 222222222222 dell'account

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccountLevelS3Actions",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowListAndReadS3ActionOnMyBucket",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket2/*",
                "arn:aws:s3:::amzn-s3-demo-bucket2"
            ]
        }
    ]
}
```

Il ruolo `efgh` deve consentire al ruolo del profilo dell'istanza `abcd` di assumerlo. A tale scopo, il ruolo `efgh` deve disporre della seguente policy di attendibilità:

Policy di attendibilità del ruolo ***efgh*** dell'account 222222222222

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "efghTrustPolicy",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"AWS": "arn:aws:iam::111111111111:role/abcd"}
        }
    ]
}
```

Per eseguire quindi AWS CLI i comandi nell'account222222222222, è necessario aggiornare il file di configurazione CLI. Identifica il ruolo `efgh` come il "profilo" e il ruolo del profilo dell'istanza EC2 `abcd` come "origine delle credenziali" nel file di configurazione di AWS CLI . I comandi della CLI vengono quindi eseguiti con le autorizzazioni del ruolo `efgh`, non il ruolo `abcd` originale.

> **Note**
>
> Per motivi di sicurezza, è possibile utilizzare AWS CloudTrail per verificare l'uso dei ruoli nell'account. Per distinguere le sessioni di ruolo quando un ruolo viene utilizzato da diversi responsabili nei CloudTrail registri, è possibile utilizzare il nome della sessione del ruolo. Quando AWS CLI assume un ruolo per conto di un utente come descritto in questo argomento, viene creato automaticamente un nome di sessione di ruolo come. `AWS-CLI-session-`*nnnnnnnn nnnnnnnn*Ecco un numero intero che rappresenta l'ora in epoca Unix (il numero di secondi trascorsi dalla mezzanotte UTC del 1° gennaio 1970). Per ulteriori informazioni, vedere CloudTrail Event Reference nella Guida per l'utente. AWS CloudTrail

Per consentire a un ruolo del profilo di un'istanza EC2 di passare a un ruolo tra più account (AWS CLI)

1. Non è necessario configurare un profilo predefinito della CLI. Al contrario, puoi caricare le credenziali dai metadati del profilo dell'istanza EC2. Crea un nuovo profilo per il ruolo nel file `.aws/config`. L'esempio seguente crea un profilo `instancecrossaccount` che passa al ruolo *`efgh`* nell'account `222222222222`. Quando questo profilo viene richiamato, AWS CLI utilizza le credenziali dei metadati del profilo dell'istanza EC2 per richiedere le credenziali per il ruolo. Per questo motivo, il ruolo del profilo dell'istanza EC2 deve disporre delle autorizzazioni `sts:AssumeRole` per il ruolo specificato nel `role_arn`.

   ```
   [profile instancecrossaccount]
   role_arn = arn:aws:iam::222222222222:role/efgh
   credential_source = Ec2InstanceMetadata
   ```

2. Dopo aver creato il nuovo profilo, qualsiasi AWS CLI comando che specifica il parametro `--profile instancecrossaccount` viene eseguito in base alle autorizzazioni associate al ruolo nell'`efgh`account. `222222222222`

   ```
   aws s3 ls amzn-s3-demo-bucket2 --profile instancecrossaccount
   ```

   Questo comando funziona se le autorizzazioni che vengono assegnate al ruolo `efgh` consentono di elencare gli utenti nell' Account AWS corrente.

3. Per tornare alle autorizzazioni del profilo dell'istanza EC2 originale nell'account `111111111111`, esegui i comandi della CLI senza il parametro `--profile`.

Per ulteriori informazioni, consulta Assunzione di un ruolo nella Guida per l'utente di AWS Command Line Interface .

Passa a un ruolo IAM (Strumenti per Windows PowerShell)

Un ruolo specifica un set di autorizzazioni da utilizzare per accedere alle risorse AWS necessarie. In questo senso, è simile a un utente in AWS Identity and Access Management (IAM). Quando effettui l'accesso come utente, ottieni uno specifico set di autorizzazioni. Tuttavia, non accedi a un ruolo, ma una volta effettuato l'accesso puoi passare a un ruolo. Ciò consente di accantonare temporaneamente le autorizzazioni utente originali e usufruire invece delle autorizzazioni assegnate al ruolo. Il ruolo può trovarsi nel tuo account o in qualsiasi altro Account AWS. Per ulteriori

informazioni sui ruoli e i relativi vantaggi e su come crearli e configurarli, consulta Ruoli IAM e Creazione di ruoli IAM.

> **Important**
>
> Le autorizzazioni dell'utente IAM e di qualsiasi ruolo a cui si passa non sono cumulative. Un solo set di autorizzazioni è attivo alla volta. Quando passi a un ruolo, lasci temporaneamente le autorizzazioni utente e utilizzi le autorizzazioni assegnate al ruolo. Quando lasci il ruolo, le autorizzazioni utente vengono automaticamente ripristinate.

Questa sezione descrive come cambiare ruoli quando utilizzi la riga di comando con gli AWS Tools for Windows PowerShell.

Immagina di avere un account nell'ambiente di sviluppo e di dover occasionalmente lavorare con l'ambiente di produzione dalla riga di comando utilizzando gli Strumenti per Windows PowerShell. Disponi già di un set di credenziali con chiave di accesso. Può trattarsi di una coppia di chiavi di accesso assegnata all'utente IAM standard. In alternativa, se hai effettuato l'accesso come principale federato SAML o OIDC, può trattarsi della coppia di chiavi di accesso per il ruolo inizialmente assegnato. Puoi utilizzare queste credenziali per eseguire il cmdlet `Use-STSRole` che passa l'ARN di un nuovo ruolo come parametro. Il comando restituisce le credenziali di sicurezza temporanee per il ruolo richiesto. È quindi possibile utilizzare tali credenziali nei PowerShell comandi successivi con le autorizzazioni del ruolo per accedere alle risorse in produzione. Mentre utilizzi il ruolo, non puoi utilizzare le autorizzazioni utente dell'account di sviluppo perché è attivo un solo set di autorizzazioni alla volta.

> **Note**
>
> Per motivi di sicurezza, gli amministratori possono esaminare AWS CloudTrail i registri per scoprire chi ha eseguito un'azione in. AWS L'amministratore potrebbe richiedere di specificare una identità di origine o un nome della sessione del ruolo quando si assume il ruolo. Per ulteriori informazioni, consultare `sts:SourceIdentity` e `sts:RoleSessionName`.

Tutte le chiavi di accesso e i token sono solo esempi e non possono essere utilizzati come mostrato. Sostituiscili con i valori appropriati del tuo ambiente reale.

Per passare a un ruolo (Strumenti per Windows PowerShell)

1. Apri un PowerShell prompt dei comandi e configura il profilo predefinito per utilizzare la chiave di accesso del tuo attuale utente IAM o del tuo ruolo federato. Se hai già utilizzato gli Strumenti per Windows PowerShell, probabilmente è già stato fatto. Tieni presente che è possibile cambiare ruolo solo se hai effettuato l'accesso come utente IAM e non come Utente root dell'account AWS.

   ```
   PS C:\> Set-AWSCredentials -AccessKey AKIAIOSFODNN7EXAMPLE -
   SecretKey wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY -StoreAs MyMainUserProfile
   PS C:\> Initialize-AWSDefaults -ProfileName MyMainUserProfile -Region us-east-2
   ```

   Per ulteriori informazioni, vedere Utilizzo AWS delle credenziali nella Guida per l'AWS Strumenti per PowerShell utente.

2. Per recuperare le credenziali per il nuovo ruolo, eseguire il comando seguente per passare al ruolo *RoleName* nell'account 123456789012. L'ARN del ruolo si ottiene dall'amministratore dell'account che ha creato il ruolo. Il comando richiede di fornire anche un nome di sessione. È possibile selezionare qualsiasi testo. Il comando seguente richiede le credenziali e quindi acquisisce l'oggetto proprietà `Credentials` dall'oggetto risultati restituiti e lo memorizza nella variabile `$Creds`.

   ```
   PS C:\> $Creds = (Use-STSRole -RoleArn "arn:aws:iam::123456789012:role/RoleName" -
   RoleSessionName "MyRoleSessionName").Credentials
   ```

   `$Creds` è un oggetto che ora contiene gli elementi `AccessKeyId`, `SecretAccessKey` e `SessionToken` necessari nelle fasi successive. I seguenti comandi di esempio illustrano valori tipici:

   ```
   PS C:\> $Creds.AccessKeyId
   AKIAIOSFODNN7EXAMPLE

   PS C:\> $Creds.SecretAccessKey
   wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

   PS C:\> $Creds.SessionToken
   AQoDYXdzEGcaEXAMPLE2gsYULo
   +Im5ZEXAMPLEeYjs1M2FUIgIJx9tQqNMBEXAMPLECvSRyh0FW7jEXAMPLEW+vE/7s1HRp
   XviG7b+qYf4nD00EXAMPLEmj4wxS04L/uZEXAMPLECihzFB5lTYLto9dyBgSDyEXAMPLE9/
   g7QRUhZp4bqbEXAMPLENwGPy
   ```

```
Oj59pFA4lNKCIkVgkREXAMPLEjlzxQ7y52gekeVEXAMPLEDiB9ST3UuysgsKdEXAMPLE1TVastU1A0SKFEXAMPLEiyw
C
s8EXAMPLEpZgOs+6hz4AP4KEXAMPLERbASP+4eZScEXAMPLEsnf87eNhyDHq6ikBQ==

PS C:\> $Creds.Expiration
Thursday, June 18, 2018 2:28:31 PM
```

3. Per utilizzare queste credenziali per ogni successivo comando, includerle con il parametro `-Credential`. Ad esempio, il comando seguente utilizza le credenziali del ruolo e funziona solo se al ruolo viene concessa l'autorizzazione `iam:ListRoles` grazie alla quale può quindi eseguire il cmdlet `Get-IAMRoles`:

```
PS C:\> get-iamroles -Credential $Creds
```

4. Per tornare alle credenziali originali, è sufficiente interrompere l'utilizzo del `-Credentials $Creds` parametro e consentire PowerShell il ripristino delle credenziali archiviate nel profilo predefinito.

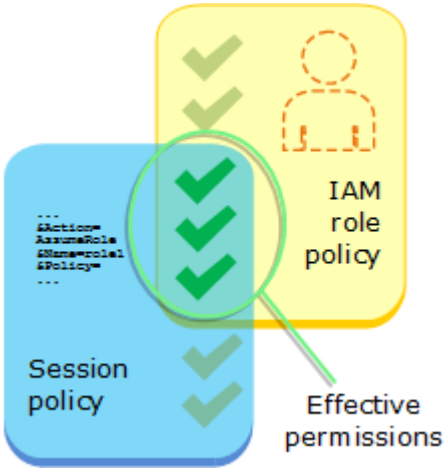
Passa a un ruolo IAM (AWS API)

Un ruolo specifica un set di autorizzazioni da utilizzare per accedere alle risorse di AWS . In questo senso, è simile a un utente IAM. Un principale (persona o applicazione) assume il ruolo di ricevere le autorizzazioni temporanee per svolgere le attività richieste e interagire con AWS le risorse. Il ruolo può trovarsi nel tuo account o in qualsiasi altro Account AWS. Per ulteriori informazioni sui ruoli e i relativi vantaggi e su come crearli e configurarli, consulta Ruoli IAM e Creazione di ruoli IAM. Per informazioni sui diversi metodi che si possono utilizzare per assumere un ruolo, consulta Metodi per assumere un ruolo.

> **Important**
>
> Le autorizzazioni dell'utente IAM e di qualsiasi ruolo assunto non sono cumulative. Un solo set di autorizzazioni è attivo alla volta. Quando si assume un ruolo, si lascia temporaneamente l'utente precedente o le autorizzazioni del ruolo e si lavora con le autorizzazioni assegnate al ruolo. Quando lasci il ruolo, le autorizzazioni originali vengono automaticamente ripristinate.

Per assumere un ruolo, un'applicazione chiama l'operazione AWS STS `AssumeRole`API e passa l'ARN del ruolo da utilizzare. L'operazione crea una nuova sessione con le credenziali temporanee. Questa sessione ha le stesse autorizzazioni delle policy basate su identità per quel ruolo.

Quando chiami `AssumeRole`, puoi passare facoltativamente policy di sessione inline o gestite. Le policy di sessione sono policy avanzate che vengono passate come un parametro quando si crea a livello di programmazione una sessione di credenziali temporanee per un ruolo o per una sessione dell'utente federato. Puoi passare un singolo documento della policy di sessione inline JSON utilizzando il parametro `Policy`. Puoi utilizzare il parametro `PolicyArns` per specificare fino a 10 policy di sessione gestite. Le autorizzazioni della sessione risultanti sono l'intersezione delle policy basate sull'identità dell'entità e delle policy di sessione. Le policy di sessione sono utili quando occorre fornire le credenziali temporanee del ruolo a un'altra persona, che potrà usare le credenziali temporanee del ruolo nelle chiamate API AWS successive, per accedere alle risorse nell'account che possiede il ruolo. Non è possibile utilizzare policy di sessione per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata su identità. Per ulteriori informazioni su come AWS determina le autorizzazioni effettive di un ruolo, consulta. Logica di valutazione delle policy



Puoi effettuare una chiamata `AssumeRole` quando hai effettuato l'accesso come utente IAM oppure come utente autenticato esternamente (SAML o OIDC) che sta già utilizzando un ruolo. Puoi anche utilizzare una concatenazione dei ruoli ovvero usare un ruolo per definirne un secondo. Non è possibile assumere un ruolo quando si è effettuato l'accesso come Utente root dell'account AWS.

Come impostazione predefinita, la sessione del ruolo dura un'ora. Quando assumi questo ruolo utilizzando le operazioni AWS STS `AssumeRole*`API, puoi specificare un valore per il `DurationSeconds` parametro. Questo valore può variare da 900 secondi (15 minuti) fino alla durata massima della sessione per il ruolo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Aggiornamento della durata massima della sessione per un ruolo.

Se scegli di ricorrere alla concatenazione dei ruoli, la durata della sessione è limitata a un'ora. Se successivamente utilizzi il parametro `DurationSeconds` per fornire un valore superiore a un'ora, l'operazione ha esito negativo.

> **Note**
>
> Per motivi di sicurezza, gli amministratori possono esaminare AWS CloudTrail i log per scoprire chi ha eseguito un'azione in. AWS L'amministratore potrebbe richiedere di specificare una identità di origine o un nome della sessione del ruolo quando si assume il ruolo. Per ulteriori informazioni, consultare `sts:SourceIdentity` e `sts:RoleSessionName`.

Gli esempi di codice seguenti mostrano come creare un utente e assumere un ruolo.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come Centro identità AWS IAM.

- Crea un utente che non disponga di autorizzazioni.
- Crea un ruolo che conceda l'autorizzazione per elencare i bucket Amazon S3 per l'account.
- Aggiungi una policy per consentire all'utente di assumere il ruolo.
- Assumi il ruolo ed elenca i bucket S3 utilizzando le credenziali temporanee, quindi ripulisci le risorse.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
global using Amazon.IdentityManagement;
global using Amazon.S3;
global using Amazon.SecurityToken;
global using IAMActions;
global using IamScenariosCommon;
global using Microsoft.Extensions.DependencyInjection;
global using Microsoft.Extensions.Hosting;
global using Microsoft.Extensions.Logging;
global using Microsoft.Extensions.Logging.Console;
global using Microsoft.Extensions.Logging.Debug;


namespace IAMActions;

public class IAMWrapper
{
    private readonly IAmazonIdentityManagementService _IAMService;

    /// <summary>
    /// Constructor for the IAMWrapper class.
    /// </summary>
    /// <param name="IAMService">An IAM client object.</param>
    public IAMWrapper(IAmazonIdentityManagementService IAMService)
    {
        _IAMService = IAMService;
    }

    /// <summary>
    /// Attach an IAM policy to a role.
    /// </summary>
    /// <param name="policyArn">The policy to attach.</param>
    /// <param name="roleName">The role that the policy will be attached to.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AttachRolePolicyAsync(string policyArn, string
 roleName)
    {
        var response = await _IAMService.AttachRolePolicyAsync(new
 AttachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });
```

```
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Create an IAM access key for a user.
    /// </summary>
    /// <param name="userName">The username for which to create the IAM access
    /// key.</param>
    /// <returns>The AccessKey.</returns>
    public async Task<AccessKey> CreateAccessKeyAsync(string userName)
    {
        var response = await _IAMService.CreateAccessKeyAsync(new
 CreateAccessKeyRequest
        {
            UserName = userName,
        });

        return response.AccessKey;

    }


    /// <summary>
    /// Create an IAM policy.
    /// </summary>
    /// <param name="policyName">The name to give the new IAM policy.</param>
    /// <param name="policyDocument">The policy document for the new policy.</
param>
    /// <returns>The new IAM policy object.</returns>
    public async Task<ManagedPolicy> CreatePolicyAsync(string policyName, string
 policyDocument)
    {
        var response = await _IAMService.CreatePolicyAsync(new
 CreatePolicyRequest
        {
            PolicyDocument = policyDocument,
            PolicyName = policyName,
        });

        return response.Policy;
    }
```

```
    /// <summary>
    /// Create a new IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="rolePolicyDocument">The name of the IAM policy document
    /// for the new role.</param>
    /// <returns>The Amazon Resource Name (ARN) of the role.</returns>
    public async Task<string> CreateRoleAsync(string roleName, string
 rolePolicyDocument)
    {
        var request = new CreateRoleRequest
        {
            RoleName = roleName,
            AssumeRolePolicyDocument = rolePolicyDocument,
        };

        var response = await _IAMService.CreateRoleAsync(request);
        return response.Role.Arn;
    }


    /// <summary>
    /// Create an IAM service-linked role.
    /// </summary>
    /// <param name="serviceName">The name of the AWS Service.</param>
    /// <param name="description">A description of the IAM service-linked role.</
param>
    /// <returns>The IAM role that was created.</returns>
    public async Task<Role> CreateServiceLinkedRoleAsync(string serviceName,
 string description)
    {
        var request = new CreateServiceLinkedRoleRequest
        {
            AWSServiceName = serviceName,
            Description = description
        };

        var response = await _IAMService.CreateServiceLinkedRoleAsync(request);
        return response.Role;
    }


    /// <summary>
```

```
    /// Create an IAM user.
    /// </summary>
    /// <param name="userName">The username for the new IAM user.</param>
    /// <returns>The IAM user that was created.</returns>
    public async Task<User> CreateUserAsync(string userName)
    {
        var response = await _IAMService.CreateUserAsync(new CreateUserRequest
{ UserName = userName });
        return response.User;
    }


    /// <summary>
    /// Delete an IAM user's access key.
    /// </summary>
    /// <param name="accessKeyId">The Id for the IAM access key.</param>
    /// <param name="userName">The username of the user that owns the IAM
    /// access key.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteAccessKeyAsync(string accessKeyId, string
userName)
    {
        var response = await _IAMService.DeleteAccessKeyAsync(new
DeleteAccessKeyRequest
        {
            AccessKeyId = accessKeyId,
            UserName = userName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM policy.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the policy to
    /// delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeletePolicyAsync(string policyArn)
    {
        var response = await _IAMService.DeletePolicyAsync(new
DeletePolicyRequest { PolicyArn = policyArn });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
```

```
    }


    /// <summary>
    /// Delete an IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRoleAsync(string roleName)
    {
        var response = await _IAMService.DeleteRoleAsync(new DeleteRoleRequest
 { RoleName = roleName });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM role policy.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="policyName">The name of the IAM role policy to delete.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRolePolicyAsync(string roleName, string
 policyName)
    {
        var response = await _IAMService.DeleteRolePolicyAsync(new
 DeleteRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM user.
    /// </summary>
    /// <param name="userName">The username of the IAM user to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserAsync(string userName)
    {
```

```
        var response = await _IAMService.DeleteUserAsync(new DeleteUserRequest
{ UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM user policy.
    /// </summary>
    /// <param name="policyName">The name of the IAM policy to delete.</param>
    /// <param name="userName">The username of the IAM user.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserPolicyAsync(string policyName, string
userName)
    {
        var response = await _IAMService.DeleteUserPolicyAsync(new
DeleteUserPolicyRequest { PolicyName = policyName, UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Detach an IAM policy from an IAM role.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the IAM
policy.</param>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DetachRolePolicyAsync(string policyArn, string
roleName)
    {
        var response = await _IAMService.DetachRolePolicyAsync(new
DetachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

```
    /// <summary>
    /// Gets the IAM password policy for an AWS account.
    /// </summary>
    /// <returns>The PasswordPolicy for the AWS account.</returns>
    public async Task<PasswordPolicy> GetAccountPasswordPolicyAsync()
    {
        var response = await _IAMService.GetAccountPasswordPolicyAsync(new
 GetAccountPasswordPolicyRequest());
        return response.PasswordPolicy;
    }


    /// <summary>
    /// Get information about an IAM policy.
    /// </summary>
    /// <param name="policyArn">The IAM policy to retrieve information for.</
param>
    /// <returns>The IAM policy.</returns>
    public async Task<ManagedPolicy> GetPolicyAsync(string policyArn)
    {

        var response = await _IAMService.GetPolicyAsync(new GetPolicyRequest
 { PolicyArn = policyArn });
        return response.Policy;
    }


    /// <summary>
    /// Get information about an IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role to retrieve information
    /// for.</param>
    /// <returns>The IAM role that was retrieved.</returns>
    public async Task<Role> GetRoleAsync(string roleName)
    {
        var response = await _IAMService.GetRoleAsync(new GetRoleRequest
        {
            RoleName = roleName,
        });

        return response.Role;
    }
```

```
    /// <summary>
    /// Get information about an IAM user.
    /// </summary>
    /// <param name="userName">The username of the user.</param>
    /// <returns>An IAM user object.</returns>
    public async Task<User> GetUserAsync(string userName)
    {
        var response = await _IAMService.GetUserAsync(new GetUserRequest
{ UserName = userName });
        return response.User;
    }


    /// <summary>
    /// List the IAM role policies that are attached to an IAM role.
    /// </summary>
    /// <param name="roleName">The IAM role to list IAM policies for.</param>
    /// <returns>A list of the IAM policies attached to the IAM role.</returns>
    public async Task<List<AttachedPolicyType>>
ListAttachedRolePoliciesAsync(string roleName)
    {
        var attachedPolicies = new List<AttachedPolicyType>();
        var attachedRolePoliciesPaginator =
_IAMService.Paginators.ListAttachedRolePolicies(new
ListAttachedRolePoliciesRequest { RoleName = roleName });

        await foreach (var response in attachedRolePoliciesPaginator.Responses)
        {
            attachedPolicies.AddRange(response.AttachedPolicies);
        }

        return attachedPolicies;
    }


    /// <summary>
    /// List IAM groups.
    /// </summary>
    /// <returns>A list of IAM groups.</returns>
    public async Task<List<Group>> ListGroupsAsync()
    {
        var groupsPaginator = _IAMService.Paginators.ListGroups(new
ListGroupsRequest());
        var groups = new List<Group>();
```

```
        await foreach (var response in groupsPaginator.Responses)
        {
            groups.AddRange(response.Groups);
        }

        return groups;
    }


    /// <summary>
    /// List IAM policies.
    /// </summary>
    /// <returns>A list of the IAM policies.</returns>
    public async Task<List<ManagedPolicy>> ListPoliciesAsync()
    {
        var listPoliciesPaginator = _IAMService.Paginators.ListPolicies(new
 ListPoliciesRequest());
        var policies = new List<ManagedPolicy>();

        await foreach (var response in listPoliciesPaginator.Responses)
        {
            policies.AddRange(response.Policies);
        }

        return policies;
    }


    /// <summary>
    /// List IAM role policies.
    /// </summary>
    /// <param name="roleName">The IAM role for which to list IAM policies.</
param>
    /// <returns>A list of IAM policy names.</returns>
    public async Task<List<string>> ListRolePoliciesAsync(string roleName)
    {
        var listRolePoliciesPaginator =
 _IAMService.Paginators.ListRolePolicies(new ListRolePoliciesRequest { RoleName =
 roleName });
        var policyNames = new List<string>();

        await foreach (var response in listRolePoliciesPaginator.Responses)
        {
```

```
            policyNames.AddRange(response.PolicyNames);
        }

        return policyNames;
    }


    /// <summary>
    /// List IAM roles.
    /// </summary>
    /// <returns>A list of IAM roles.</returns>
    public async Task<List<Role>> ListRolesAsync()
    {
        var listRolesPaginator = _IAMService.Paginators.ListRoles(new
ListRolesRequest());
        var roles = new List<Role>();

        await foreach (var response in listRolesPaginator.Responses)
        {
            roles.AddRange(response.Roles);
        }

        return roles;
    }


    /// <summary>
    /// List SAML authentication providers.
    /// </summary>
    /// <returns>A list of SAML providers.</returns>
    public async Task<List<SAMLProviderListEntry>> ListSAMLProvidersAsync()
    {
        var response = await _IAMService.ListSAMLProvidersAsync(new
ListSAMLProvidersRequest());
        return response.SAMLProviderList;
    }


    /// <summary>
    /// List IAM users.
    /// </summary>
    /// <returns>A list of IAM users.</returns>
    public async Task<List<User>> ListUsersAsync()
    {
```

```
        var listUsersPaginator = _IAMService.Paginators.ListUsers(new
 ListUsersRequest());
        var users = new List<User>();

        await foreach (var response in listUsersPaginator.Responses)
        {
            users.AddRange(response.Users);
        }

        return users;
    }


    /// <summary>
    /// Update the inline policy document embedded in a role.
    /// </summary>
    /// <param name="policyName">The name of the policy to embed.</param>
    /// <param name="roleName">The name of the role to update.</param>
    /// <param name="policyDocument">The policy document that defines the role.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutRolePolicyAsync(string policyName, string
 roleName, string policyDocument)
    {
        var request = new PutRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutRolePolicyAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Add or update an inline policy document that is embedded in an IAM user.
    /// </summary>
    /// <param name="userName">The name of the IAM user.</param>
    /// <param name="policyName">The name of the IAM policy.</param>
    /// <param name="policyDocument">The policy document defining the IAM
 policy.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
```

```
    public async Task<bool> PutUserPolicyAsync(string userName, string
 policyName, string policyDocument)
    {
        var request = new PutUserPolicyRequest
        {
            UserName = userName,
            PolicyName = policyName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutUserPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }

    /// <summary>
    /// Wait for a new access key to be ready to use.
    /// </summary>
    /// <param name="accessKeyId">The Id of the access key.</param>
    /// <returns>A boolean value indicating the success of the action.</returns>
    public async Task<bool> WaitUntilAccessKeyIsReady(string accessKeyId)
    {
        var keyReady = false;

        do
        {
            try
            {
                var response = await _IAMService.GetAccessKeyLastUsedAsync(
                    new GetAccessKeyLastUsedRequest { AccessKeyId =
 accessKeyId });
                if (response.UserName is not null)
                {
                    keyReady = true;
                }
            }
            catch (NoSuchEntityException)
            {
                keyReady = false;
            }
        } while (!keyReady);

        return keyReady;
    }
}
```

```
using Microsoft.Extensions.Configuration;

namespace IAMBasics;

public class IAMBasics
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for the AWS service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonIdentityManagementService>()
            .AddTransient<IAMWrapper>()
            .AddTransient<UIWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<IAMBasics>();


        IConfiguration configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // Values needed for user, role, and policies.
        string userName = configuration["UserName"]!;
        string s3PolicyName = configuration["S3PolicyName"]!;
        string roleName = configuration["RoleName"]!;
```

```
        var iamWrapper = host.Services.GetRequiredService<IAMWrapper>();
        var uiWrapper = host.Services.GetRequiredService<UIWrapper>();

        uiWrapper.DisplayBasicsOverview();
        uiWrapper.PressEnter();

        // First create a user. By default, the new user has
        // no permissions.
        uiWrapper.DisplayTitle("Create User");
        Console.WriteLine($"Creating a new user with user name: {userName}.");
        var user = await iamWrapper.CreateUserAsync(userName);
        var userArn = user.Arn;

        Console.WriteLine($"Successfully created user: {userName} with ARN:
{userArn}.");
        uiWrapper.WaitABit(15, "Now let's wait for the user to be ready for
use.");

        // Define a role policy document that allows the new user
        // to assume the role.
        string assumeRolePolicyDocument = "{" +
          "\"Version\": \"2012-10-17\"," +
          "\"Statement\": [{" +
              "\"Effect\": \"Allow\"," +
              "\"Principal\": {" +
              $" \"AWS\": \"{userArn}\"" +
              "}," +
              "\"Action\": \"sts:AssumeRole\"" +
          "}]" +
        "}";

        // Permissions to list all buckets.
        string policyDocument = "{" +
            "\"Version\": \"2012-10-17\"," +
            " \"Statement\" : [{" +
                " \"Action\" : [\"s3:ListAllMyBuckets\"]," +
                " \"Effect\" : \"Allow\"," +
                " \"Resource\" : \"*\"" +
            "}]" +
        "}";

        // Create an AccessKey for the user.
        uiWrapper.DisplayTitle("Create access key");
```

```
        Console.WriteLine("Now let's create an access key for the new user.");
        var accessKey = await iamWrapper.CreateAccessKeyAsync(userName);

        var accessKeyId = accessKey.AccessKeyId;
        var secretAccessKey = accessKey.SecretAccessKey;

        Console.WriteLine($"We have created the access key with Access key id:
{accessKeyId}.");

        Console.WriteLine("Now let's wait until the IAM access key is ready to
use.");
        var keyReady = await iamWrapper.WaitUntilAccessKeyIsReady(accessKeyId);

        // Now try listing the Amazon Simple Storage Service (Amazon S3)
        // buckets. This should fail at this point because the user doesn't
        // have permissions to perform this task.
        uiWrapper.DisplayTitle("Try to display Amazon S3 buckets");
        Console.WriteLine("Now let's try to display a list of the user's Amazon
S3 buckets.");
        var s3Client1 = new AmazonS3Client(accessKeyId, secretAccessKey);
        var stsClient1 = new AmazonSecurityTokenServiceClient(accessKeyId,
secretAccessKey);

        var s3Wrapper = new S3Wrapper(s3Client1, stsClient1);
        var buckets = await s3Wrapper.ListMyBucketsAsync();

        Console.WriteLine(buckets is null
            ? "As expected, the call to list the buckets has returned a null
list."
            : "Something went wrong. This shouldn't have worked.");

        uiWrapper.PressEnter();

        uiWrapper.DisplayTitle("Create IAM role");
        Console.WriteLine($"Creating the role: {roleName}");

        // Creating an IAM role to allow listing the S3 buckets. A role name
        // is not case sensitive and must be unique to the account for which it
        // is created.
        var roleArn = await iamWrapper.CreateRoleAsync(roleName,
assumeRolePolicyDocument);

        uiWrapper.PressEnter();
```

```
        // Create a policy with permissions to list S3 buckets.
        uiWrapper.DisplayTitle("Create IAM policy");
        Console.WriteLine($"Creating the policy: {s3PolicyName}");
        Console.WriteLine("with permissions to list the Amazon S3 buckets for the
account.");
        var policy = await iamWrapper.CreatePolicyAsync(s3PolicyName,
policyDocument);

        // Wait 15 seconds for the IAM policy to be available.
        uiWrapper.WaitABit(15, "Waiting for the policy to be available.");

        // Attach the policy to the role you created earlier.
        uiWrapper.DisplayTitle("Attach new IAM policy");
        Console.WriteLine("Now let's attach the policy to the role.");
        await iamWrapper.AttachRolePolicyAsync(policy.Arn, roleName);

        // Wait 15 seconds for the role to be updated.
        Console.WriteLine();
        uiWrapper.WaitABit(15, "Waiting for the policy to be attached.");

        // Use the AWS Security Token Service (AWS STS) to have the user
        // assume the role we created.
        var stsClient2 = new AmazonSecurityTokenServiceClient(accessKeyId,
secretAccessKey);

        // Wait for the new credentials to become valid.
        uiWrapper.WaitABit(10, "Waiting for the credentials to be valid.");

        var assumedRoleCredentials = await
s3Wrapper.AssumeS3RoleAsync("temporary-session", roleArn);

        // Try again to list the buckets using the client created with
        // the new user's credentials. This time, it should work.
        var s3Client2 = new AmazonS3Client(assumedRoleCredentials);

        s3Wrapper.UpdateClients(s3Client2, stsClient2);

        buckets = await s3Wrapper.ListMyBucketsAsync();

        uiWrapper.DisplayTitle("List Amazon S3 buckets");
        Console.WriteLine("This time we should have buckets to list.");
        if (buckets is not null)
        {
            buckets.ForEach(bucket =>
```

```
            {
                Console.WriteLine($"{bucket.BucketName} created:
 {bucket.CreationDate}");
            });
        }

        uiWrapper.PressEnter();

        // Now clean up all the resources used in the example.
        uiWrapper.DisplayTitle("Clean up resources");
        Console.WriteLine("Thank you for watching. The IAM Basics demo is
 complete.");
        Console.WriteLine("Please wait while we clean up the resources we
 created.");

        await iamWrapper.DetachRolePolicyAsync(policy.Arn, roleName);

        await iamWrapper.DeletePolicyAsync(policy.Arn);

        await iamWrapper.DeleteRoleAsync(roleName);

        await iamWrapper.DeleteAccessKeyAsync(accessKeyId, userName);

        await iamWrapper.DeleteUserAsync(userName);

        uiWrapper.PressEnter();

        Console.WriteLine("All done cleaning up our resources. Thank you for your
 patience.");
    }
}


namespace IamScenariosCommon;

using System.Net;

/// <summary>
/// A class to perform Amazon Simple Storage Service (Amazon S3) actions for
/// the IAM Basics scenario.
/// </summary>
public class S3Wrapper
{
    private IAmazonS3 _s3Service;
```

```
    private IAmazonSecurityTokenService _stsService;

    /// <summary>
    /// Constructor for the S3Wrapper class.
    /// </summary>
    /// <param name="s3Service">An Amazon S3 client object.</param>
    /// <param name="stsService">An AWS Security Token Service (AWS STS)
    /// client object.</param>
    public S3Wrapper(IAmazonS3 s3Service, IAmazonSecurityTokenService stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }

    /// <summary>
    /// Assumes an AWS Identity and Access Management (IAM) role that allows
    /// Amazon S3 access for the current session.
    /// </summary>
    /// <param name="roleSession">A string representing the current session.</
param>
    /// <param name="roleToAssume">The name of the IAM role to assume.</param>
    /// <returns>Credentials for the newly assumed IAM role.</returns>
    public async Task<Credentials> AssumeS3RoleAsync(string roleSession, string
 roleToAssume)
    {
        // Create the request to use with the AssumeRoleAsync call.
        var request = new AssumeRoleRequest()
        {
            RoleSessionName = roleSession,
            RoleArn = roleToAssume,
        };

        var response = await _stsService.AssumeRoleAsync(request);

        return response.Credentials;
    }


    /// <summary>
    /// Delete an S3 bucket.
    /// </summary>
    /// <param name="bucketName">Name of the S3 bucket to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteBucketAsync(string bucketName)
```

```
    {
        var result = await _s3Service.DeleteBucketAsync(new DeleteBucketRequest
{ BucketName = bucketName });
        return result.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// List the buckets that are owned by the user's account.
    /// </summary>
    /// <returns>Async Task.</returns>
    public async Task<List<S3Bucket>?> ListMyBucketsAsync()
    {
        try
        {
            // Get the list of buckets accessible by the new user.
            var response = await _s3Service.ListBucketsAsync();

            return response.Buckets;
        }
        catch (AmazonS3Exception ex)
        {
            // Something else went wrong. Display the error message.
            Console.WriteLine($"Error: {ex.Message}");
            return null;
        }
    }

    /// <summary>
    /// Create a new S3 bucket.
    /// </summary>
    /// <param name="bucketName">The name for the new bucket.</param>
    /// <returns>A Boolean value indicating whether the action completed
    /// successfully.</returns>
    public async Task<bool> PutBucketAsync(string bucketName)
    {
        var response = await _s3Service.PutBucketAsync(new PutBucketRequest
{ BucketName = bucketName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Update the client objects with new client objects. This is available
    /// because the scenario uses the methods of this class without and then
    /// with the proper permissions to list S3 buckets.
```

```
    /// </summary>
    /// <param name="s3Service">The Amazon S3 client object.</param>
    /// <param name="stsService">The AWS STS client object.</param>
    public void UpdateClients(IAmazonS3 s3Service, IAmazonSecurityTokenService
 stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }
}


namespace IamScenariosCommon;

public class UIWrapper
{
    public readonly string SepBar = new('-', Console.WindowWidth);

    /// <summary>
    /// Show information about the IAM Groups scenario.
    /// </summary>
    public void DisplayGroupsOverview()
    {
        Console.Clear();

        DisplayTitle("Welcome to the IAM Groups Demo");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates an Amazon Identity and Access Management
 (IAM) group.");
        Console.WriteLine("\t2. Adds an IAM policy to the IAM group giving it
 full access to Amazon S3.");
        Console.WriteLine("\t3. Creates a new IAM user.");
        Console.WriteLine("\t4. Creates an IAM access key for the user.");
        Console.WriteLine("\t5. Adds the user to the IAM group.");
        Console.WriteLine("\t6. Lists the buckets on the account.");
        Console.WriteLine("\t7. Proves that the user has full Amazon S3 access by
 creating a bucket.");
        Console.WriteLine("\t8. List the buckets again to show the new bucket.");
        Console.WriteLine("\t9. Cleans up all the resources created.");
    }

    /// <summary>
    /// Show information about the IAM Basics scenario.
    /// </summary>
```

```
    public void DisplayBasicsOverview()
    {
        Console.Clear();

        DisplayTitle("Welcome to IAM Basics");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates a user with no permissions.");
        Console.WriteLine("\t2. Creates a role and policy that grant
s3:ListAllMyBuckets permission.");
        Console.WriteLine("\t3. Grants the user permission to assume the role.");
        Console.WriteLine("\t4. Creates an S3 client object as the user and tries
to list buckets (this will fail).");
        Console.WriteLine("\t5. Gets temporary credentials by assuming the
role.");
        Console.WriteLine("\t6. Creates a new S3 client object with the temporary
credentials and lists the buckets (this will succeed).");
        Console.WriteLine("\t7. Deletes all the resources.");
    }

    /// <summary>
    /// Display a message and wait until the user presses enter.
    /// </summary>
    public void PressEnter()
    {
        Console.Write("\nPress <Enter> to continue. ");
        _ = Console.ReadLine();
        Console.WriteLine();
    }

    /// <summary>
    /// Pad a string with spaces to center it on the console display.
    /// </summary>
    /// <param name="strToCenter">The string to be centered.</param>
    /// <returns>The padded string.</returns>
    public string CenterString(string strToCenter)
    {
        var padAmount = (Console.WindowWidth - strToCenter.Length) / 2;
        var leftPad = new string(' ', padAmount);
        return $"{leftPad}{strToCenter}";
    }

    /// <summary>
    /// Display a line of hyphens, the centered text of the title, and another
    /// line of hyphens.
```

```
    /// </summary>
    /// <param name="strTitle">The string to be displayed.</param>
    public void DisplayTitle(string strTitle)
    {
        Console.WriteLine(SepBar);
        Console.WriteLine(CenterString(strTitle));
        Console.WriteLine(SepBar);
    }

    /// <summary>
    /// Display a countdown and wait for a number of seconds.
    /// </summary>
    /// <param name="numSeconds">The number of seconds to wait.</param>
    public void WaitABit(int numSeconds, string msg)
    {
        Console.WriteLine(msg);

        // Wait for the requested number of seconds.
        for (int i = numSeconds; i > 0; i--)
        {
            System.Threading.Thread.Sleep(1000);
            Console.Write($"{i}...");
        }

        PressEnter();
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per .NET .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole

- DeleteUser
- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
###############################################################################
# function iam_create_user_assume_role
#
# Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
#
#     "IAM access" permissions are needed to run this code.
#     "STS assume role" permissions are needed to run this code. (Note: It might
 be necessary to
#           create a custom policy).
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function iam_create_user_assume_role() {
  {
    if [ "$IAM_OPERATIONS_SOURCED" != "True" ]; then

      source ./iam_operations.sh
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the IAM create user and assume role demo."
```

```
  echo
  echo "This demo will create an IAM user, create an IAM role, and apply the role
 to the user."
  echo_repeat "*" 88
  echo

  echo -n "Enter a name for a new IAM user: "
  get_input
  user_name=$get_input_result

  local user_arn
  user_arn=$(iam_create_user -u "$user_name")

  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created demo IAM user named $user_name"
  else
    errecho "$user_arn"
    errecho "The user failed to create. This demo will exit."
    return 1
  fi

  local access_key_response
  access_key_response=$(iam_create_user_access_key -u "$user_name")
  # shellcheck disable=SC2181
  if [[ ${?} != 0 ]]; then
    errecho "The access key failed to create. This demo will exit."
    clean_up "$user_name"
    return 1
  fi

  IFS=$'\t ' read -r -a access_key_values <<<"$access_key_response"
  local key_name=${access_key_values[0]}
  local key_secret=${access_key_values[1]}

  echo "Created access key named $key_name"

  echo "Wait 10 seconds for the user to be ready."
  sleep 10
  echo_repeat "*" 88
  echo

  local iam_role_name
  iam_role_name=$(generate_random_name "test-role")
```

```
 echo "Creating a role named $iam_role_name with user $user_name as the
principal."

 local assume_role_policy_document="{
   \"Version\": \"2012-10-17\",
   \"Statement\": [{
       \"Effect\": \"Allow\",
       \"Principal\": {\"AWS\": \"$user_arn\"},
       \"Action\": \"sts:AssumeRole\"
       }]
   }"

 local role_arn
 role_arn=$(iam_create_role -n "$iam_role_name" -p
"$assume_role_policy_document")

 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   echo "Created IAM role named $iam_role_name"
 else
   errecho "The role failed to create. This demo will exit."
   clean_up "$user_name" "$key_name"
   return 1
 fi

 local policy_name
 policy_name=$(generate_random_name "test-policy")
 local policy_document="{
              \"Version\": \"2012-10-17\",
              \"Statement\": [{
                  \"Effect\": \"Allow\",
                  \"Action\": \"s3:ListAllMyBuckets\",
                  \"Resource\": \"arn:aws:s3:::*\"}]}"

 local policy_arn
 policy_arn=$(iam_create_policy -n "$policy_name" -p "$policy_document")
 # shellcheck disable=SC2181
 if [[ ${?} == 0 ]]; then
   echo "Created  IAM policy named $policy_name"
 else
   errecho "The policy failed to create."
   clean_up "$user_name" "$key_name" "$iam_role_name"
   return 1
 fi
```

```
 if (iam_attach_role_policy -n "$iam_role_name" -p "$policy_arn"); then
   echo "Attached policy $policy_arn to role $iam_role_name"
 else
   errecho "The policy failed to attach."
   clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
   return 1
 fi

 local assume_role_policy_document="{
               \"Version\": \"2012-10-17\",
               \"Statement\": [{
                   \"Effect\": \"Allow\",
                   \"Action\": \"sts:AssumeRole\",
                   \"Resource\": \"$role_arn\"}]}"

 local assume_role_policy_name
 assume_role_policy_name=$(generate_random_name "test-assume-role-")

 # shellcheck disable=SC2181
 local assume_role_policy_arn
 assume_role_policy_arn=$(iam_create_policy -n "$assume_role_policy_name" -p
"$assume_role_policy_document")
 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   echo "Created  IAM policy named $assume_role_policy_name for sts assume role"
 else
   errecho "The policy failed to create."
   clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn"
   return 1
 fi

 echo "Wait 10 seconds to give AWS time to propagate these new resources and
connections."
 sleep 10
 echo_repeat "*" 88
 echo

 echo "Try to list buckets without the new user assuming the role."
 echo_repeat "*" 88
 echo

 # Set the environment variables for the created user.
```

```
 # bashsupport disable=BP2001
 export AWS_ACCESS_KEY_ID=$key_name
 # bashsupport disable=BP2001
 export AWS_SECRET_ACCESS_KEY=$key_secret

 local buckets
 buckets=$(s3_list_buckets)

 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   local bucket_count
   bucket_count=$(echo "$buckets" | wc -w | xargs)
   echo "There are $bucket_count buckets in the account. This should not have
happened."
 else
   errecho "Because the role with permissions has not been assumed, listing
buckets failed."
 fi

 echo
 echo_repeat "*" 88
 echo "Now assume the role $iam_role_name and list the buckets."
 echo_repeat "*" 88
 echo

 local credentials

 credentials=$(sts_assume_role -r "$role_arn" -n "AssumeRoleDemoSession")
 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   echo "Assumed role $iam_role_name"
 else
   errecho "Failed to assume role."
   export AWS_ACCESS_KEY_ID=""
   export AWS_SECRET_ACCESS_KEY=""
   clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn" "$assume_role_policy_arn"
   return 1
 fi

 IFS=$'\t ' read -r -a credentials <<<"$credentials"

 export AWS_ACCESS_KEY_ID=${credentials[0]}
 export AWS_SECRET_ACCESS_KEY=${credentials[1]}
```

```
  # bashsupport disable=BP2001
  export AWS_SESSION_TOKEN=${credentials[2]}

  buckets=$(s3_list_buckets)

  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. Listing buckets
 succeeded because of "
    echo "the assumed role."
  else
    errecho "Failed to list buckets. This should not happen."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    export AWS_SESSION_TOKEN=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
 "$policy_arn" "$assume_role_policy_arn"
    return 1
  fi

  local result=0
  export AWS_ACCESS_KEY_ID=""
  export AWS_SECRET_ACCESS_KEY=""

  echo
  echo_repeat "*" 88
  echo "The created resources will now be deleted."
  echo_repeat "*" 88
  echo

  clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
 "$assume_role_policy_arn"

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    result=1
  fi

  return $result
}
```

Le funzioni IAM utilizzate in questo scenario.

```
##############################################################################
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
# Returns:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
##############################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}

  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}

##############################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
```

```
# it already exists.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an AWS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
```

```
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user already exists, we don't want to try to create it.
  if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
  fi

  response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
#     And:
#       0 - If successful.
```

```
#       1 - If it fails.
###############################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]   Optional file name for the access key output."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:f:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      f) file_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

  local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  fi

  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}

###############################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
```

```
  echo "Creates an AWS Identity and Access Management (IAM) role."
  echo "  -n role_name   The name of the IAM role."
  echo "  -p policy_json -- The assume role policy document."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-role \
  --role-name "$role_name" \
  --assume-role-policy-document "$policy_document" \
  --output text \
  --query Role.Arn)

local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_policy() {
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name   The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) policy_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
```

```
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-policy \
    --policy-name "$policy_name" \
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
```

```
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
```

```
    return 1
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam attach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
```

```
 function usage() {
   echo "function iam_detach_role_policy"
   echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
   echo "  -n role_name   The name of the IAM role."
   echo "  -p policy_ARN -- The IAM policy document ARN."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "n:p:h" option; do
   case "${option}" in
     n) role_name="${OPTARG}" ;;
     p) policy_arn="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$role_name" ]]; then
   errecho "ERROR: You must provide a role name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$policy_arn" ]]; then
   errecho "ERROR: You must provide a policy ARN with the -p parameter."
   usage
   return 1
 fi

 response=$(aws iam detach-role-policy \
   --role-name "$role_name" \
   --policy-arn "$policy_arn")

 local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an AWS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
```

```
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Policy arn:  $policy_arn"
  iecho ""

  response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
```

```
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an AWS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""
```

```
  response=$(aws iam delete-role \
    --role-name "$role_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_access_key() {
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_access_key"
    echo "Deletes an AWS Identity and Access Management (IAM) access key for the
 specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to delete."
    echo ""
  }
```

```
  # Retrieve the calling parameters.
  while getopts "u:k:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      k) access_key="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Username:   $user_name"
  iecho "    Access key:   $access_key"
  iecho ""

  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
```

```
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an AWS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
```

```
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user does not exist, we don't want to try to delete it.
  if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
  fi

  response=$(aws iam delete-user \
    --user-name "$user_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .

- AttachRolePolicy
- CreateAccessKey
- CreatePolicy
- CreateRole
- CreateUser
- DeleteAccessKey
- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
namespace AwsDoc {
    namespace IAM {

        //! Cleanup by deleting created entities.
        /*!
          \sa DeleteCreatedEntities
          \param client: IAM client.
          \param role: IAM role.
          \param user: IAM user.
          \param policy: IAM policy.
        */
        static bool DeleteCreatedEntities(const Aws::IAM::IAMClient &client,
                                          const Aws::IAM::Model::Role &role,
                                          const Aws::IAM::Model::User &user,
```

```
                                            const Aws::IAM::Model::Policy &policy);
    }

    static const int LIST_BUCKETS_WAIT_SEC = 20;

    static const char ALLOCATION_TAG[] = "example_code";
}

//! Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
// "IAM access" permissions are needed to run this code.
// "STS assume role" permissions are needed to run this code. (Note: It might be
 necessary to
//    create a custom policy).
/*!
  \sa iamCreateUserAssumeRoleScenario
  \param clientConfig: Aws client configuration.
  \return bool: Successful completion.
*/
bool AwsDoc::IAM::iamCreateUserAssumeRoleScenario(
        const Aws::Client::ClientConfiguration &clientConfig) {

    Aws::IAM::IAMClient client(clientConfig);
    Aws::IAM::Model::User user;
    Aws::IAM::Model::Role role;
    Aws::IAM::Model::Policy policy;

    // 1. Create a user.
    {
        Aws::IAM::Model::CreateUserRequest request;
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String userName = "iam-demo-user-" +
                               Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetUserName(userName);

        Aws::IAM::Model::CreateUserOutcome outcome = client.CreateUser(request);
        if (!outcome.IsSuccess()) {
            std::cout << "Error creating IAM user " << userName << ":" <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }
        else {
            std::cout << "Successfully created IAM user " << userName <<
 std::endl;
```

```
        }

        user = outcome.GetResult().GetUser();
    }

    // 2. Create a role.
    {
        // Get the IAM user for the current client in order to access its ARN.
        Aws::String iamUserArn;
        {
            Aws::IAM::Model::GetUserRequest request;
            Aws::IAM::Model::GetUserOutcome outcome = client.GetUser(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error getting Iam user. " <<
                          outcome.GetError().GetMessage() << std::endl;

                DeleteCreatedEntities(client, role, user, policy);
                return false;
            }
            else {
                std::cout << "Successfully retrieved Iam user "
                          << outcome.GetResult().GetUser().GetUserName()
                          << std::endl;
            }

            iamUserArn = outcome.GetResult().GetUser().GetArn();
        }

        Aws::IAM::Model::CreateRoleRequest request;

        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String roleName = "iam-demo-role-" +
                               Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetRoleName(roleName);

        // Build policy document for role.
        Aws::Utils::Document jsonStatement;
        jsonStatement.WithString("Effect", "Allow");

        Aws::Utils::Document jsonPrincipal;
        jsonPrincipal.WithString("AWS", iamUserArn);
        jsonStatement.WithObject("Principal", jsonPrincipal);
        jsonStatement.WithString("Action", "sts:AssumeRole");
        jsonStatement.WithObject("Condition", Aws::Utils::Document());
```

```
        Aws::Utils::Document policyDocument;
        policyDocument.WithString("Version", "2012-10-17");

        Aws::Utils::Array<Aws::Utils::Document> statements(1);
        statements[0] = jsonStatement;
        policyDocument.WithArray("Statement", statements);

        std::cout << "Setting policy for role\n   "
                  << policyDocument.View().WriteCompact() << std::endl;

        // Set role policy document as JSON string.

request.SetAssumeRolePolicyDocument(policyDocument.View().WriteCompact());

        Aws::IAM::Model::CreateRoleOutcome outcome = client.CreateRole(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error creating role. " <<
                      outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully created a role with name " << roleName
                      << std::endl;
        }

        role = outcome.GetResult().GetRole();
    }

    // 3. Create an IAM policy.
    {
        Aws::IAM::Model::CreatePolicyRequest request;
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String policyName = "iam-demo-policy-" +
                                 Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetPolicyName(policyName);

        // Build IAM policy document.
        Aws::Utils::Document jsonStatement;
        jsonStatement.WithString("Effect", "Allow");
        jsonStatement.WithString("Action", "s3:ListAllMyBuckets");
        jsonStatement.WithString("Resource", "arn:aws:s3:::*");
```

```
        Aws::Utils::Document policyDocument;
        policyDocument.WithString("Version", "2012-10-17");

        Aws::Utils::Array<Aws::Utils::Document> statements(1);
        statements[0] = jsonStatement;
        policyDocument.WithArray("Statement", statements);

        std::cout << "Creating a policy.\n   " <<
policyDocument.View().WriteCompact()
                  << std::endl;

        // Set IAM policy document as JSON string.
        request.SetPolicyDocument(policyDocument.View().WriteCompact());

        Aws::IAM::Model::CreatePolicyOutcome outcome =
client.CreatePolicy(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error creating policy. " <<
                      outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully created a policy with name, " <<
policyName <<
                      "." << std::endl;
        }

        policy = outcome.GetResult().GetPolicy();
    }

    // 4. Assume the new role using the AWS Security Token Service (STS).
    Aws::STS::Model::Credentials credentials;
    {
        Aws::STS::STSClient stsClient(clientConfig);

        Aws::STS::Model::AssumeRoleRequest request;
        request.SetRoleArn(role.GetArn());
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String roleSessionName = "iam-demo-role-session-" +

Aws::Utils::StringUtils::ToLower(uuid.c_str());
```

```
        request.SetRoleSessionName(roleSessionName);

        Aws::STS::Model::AssumeRoleOutcome assumeRoleOutcome;

        // Repeatedly call AssumeRole, because there is often a delay
        // before the role is available to be assumed.
        // Repeat at most 20 times when access is denied.
        int count = 0;
        while (true) {
            assumeRoleOutcome = stsClient.AssumeRole(request);
            if (!assumeRoleOutcome.IsSuccess()) {
                if (count > 20 ||
                    assumeRoleOutcome.GetError().GetErrorType() !=
                    Aws::STS::STSErrors::ACCESS_DENIED) {
                    std::cerr << "Error assuming role after 20 tries. " <<
                              assumeRoleOutcome.GetError().GetMessage() <<
std::endl;

                    DeleteCreatedEntities(client, role, user, policy);
                    return false;
                }
                std::this_thread::sleep_for(std::chrono::seconds(1));
            }
            else {
                std::cout << "Successfully assumed the role after " << count
                          << " seconds." << std::endl;
                break;
            }
            count++;
        }

        credentials = assumeRoleOutcome.GetResult().GetCredentials();
    }


    // 5. List objects in the bucket (This should fail).
    {
        Aws::S3::S3Client s3Client(
                Aws::Auth::AWSCredentials(credentials.GetAccessKeyId(),
                                          credentials.GetSecretAccessKey(),
                                          credentials.GetSessionToken()),
                Aws::MakeShared<Aws::S3::S3EndpointProvider>(ALLOCATION_TAG),
                clientConfig);
```

```
       Aws::S3::Model::ListBucketsOutcome listBucketsOutcome =
s3Client.ListBuckets();
       if (!listBucketsOutcome.IsSuccess()) {
           if (listBucketsOutcome.GetError().GetErrorType() !=
               Aws::S3::S3Errors::ACCESS_DENIED) {
               std::cerr << "Could not lists buckets. " <<
                         listBucketsOutcome.GetError().GetMessage() <<
std::endl;
           }
           else {
               std::cout
                       << "Access to list buckets denied because privileges have
not been applied."
                       << std::endl;
           }
       }
       else {
           std::cerr
                   << "Successfully retrieved bucket lists when this should not
happen."
                   << std::endl;
       }
   }

   // 6. Attach the policy to the role.
   {
       Aws::IAM::Model::AttachRolePolicyRequest request;
       request.SetRoleName(role.GetRoleName());
       request.WithPolicyArn(policy.GetArn());

       Aws::IAM::Model::AttachRolePolicyOutcome outcome =
client.AttachRolePolicy(
               request);
       if (!outcome.IsSuccess()) {
           std::cerr << "Error creating policy. " <<
                     outcome.GetError().GetMessage() << std::endl;

           DeleteCreatedEntities(client, role, user, policy);
           return false;
       }
       else {
           std::cout << "Successfully attached the policy with name, "
                     << policy.GetPolicyName() <<
```

```
                    ", to the role, " << role.GetRoleName() << "." <<
std::endl;
        }
    }

    int count = 0;
    // 7. List objects in the bucket (this should succeed).
    // Repeatedly call ListBuckets, because there is often a delay
    // before the policy with ListBucket permissions has been applied to the
role.
    // Repeat at most LIST_BUCKETS_WAIT_SEC times when access is denied.
    while (true) {
        Aws::S3::S3Client s3Client(
                Aws::Auth::AWSCredentials(credentials.GetAccessKeyId(),
                                          credentials.GetSecretAccessKey(),
                                          credentials.GetSessionToken()),
                Aws::MakeShared<Aws::S3::S3EndpointProvider>(ALLOCATION_TAG),
                clientConfig);
        Aws::S3::Model::ListBucketsOutcome listBucketsOutcome =
s3Client.ListBuckets();
        if (!listBucketsOutcome.IsSuccess()) {
            if ((count > LIST_BUCKETS_WAIT_SEC) ||
                listBucketsOutcome.GetError().GetErrorType() !=
                Aws::S3::S3Errors::ACCESS_DENIED) {
                std::cerr << "Could not lists buckets after " <<
LIST_BUCKETS_WAIT_SEC << " seconds. " <<
                          listBucketsOutcome.GetError().GetMessage() <<
std::endl;
                DeleteCreatedEntities(client, role, user, policy);
                return false;
            }

            std::this_thread::sleep_for(std::chrono::seconds(1));
        }
        else {

            std::cout << "Successfully retrieved bucket lists after " << count
                      << " seconds." << std::endl;
            break;
        }
        count++;
    }

    // 8. Delete all the created resources.
```

```
    return DeleteCreatedEntities(client, role, user, policy);
}

bool AwsDoc::IAM::DeleteCreatedEntities(const Aws::IAM::IAMClient &client,
                                        const Aws::IAM::Model::Role &role,
                                        const Aws::IAM::Model::User &user,
                                        const Aws::IAM::Model::Policy &policy) {
    bool result = true;
    if (policy.ArnHasBeenSet()) {
        // Detach the policy from the role.
        {
            Aws::IAM::Model::DetachRolePolicyRequest request;
            request.SetPolicyArn(policy.GetArn());
            request.SetRoleName(role.GetRoleName());

            Aws::IAM::Model::DetachRolePolicyOutcome outcome =
 client.DetachRolePolicy(
                    request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error Detaching policy from roles. " <<
                          outcome.GetError().GetMessage() << std::endl;
                result = false;
            }
            else {
                std::cout << "Successfully detached the policy with arn "
                          << policy.GetArn()
                          << " from role " << role.GetRoleName() << "." <<
 std::endl;
            }
        }

        // Delete the policy.
        {
            Aws::IAM::Model::DeletePolicyRequest request;
            request.WithPolicyArn(policy.GetArn());

            Aws::IAM::Model::DeletePolicyOutcome outcome =
 client.DeletePolicy(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error deleting policy. " <<
                          outcome.GetError().GetMessage() << std::endl;
                result = false;
            }
            else {
```

```
                std::cout << "Successfully deleted the policy with arn "
                          << policy.GetArn() << std::endl;
            }
        }

    }

    if (role.RoleIdHasBeenSet()) {
        // Delete the role.
        Aws::IAM::Model::DeleteRoleRequest request;
        request.SetRoleName(role.GetRoleName());

        Aws::IAM::Model::DeleteRoleOutcome outcome = client.DeleteRole(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error deleting role. " <<
                      outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Successfully deleted the role with name "
                      << role.GetRoleName() << std::endl;
        }
    }

    if (user.ArnHasBeenSet()) {
        // Delete the user.
        Aws::IAM::Model::DeleteUserRequest request;
        request.WithUserName(user.GetUserName());

        Aws::IAM::Model::DeleteUserOutcome outcome = client.DeleteUser(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error deleting user. " <<
                      outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Successfully deleted the user with name "
                      << user.GetUserName() << std::endl;
        }
    }

    return result;
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per C++ .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

Go

SDK per Go V2

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
import (
 "context"
 "errors"
 "fmt"
 "log"
 "math/rand"
```

```
 "strings"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/credentials"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/aws-sdk-go-v2/service/s3"
 "github.com/aws/aws-sdk-go-v2/service/sts"
 "github.com/aws/smithy-go"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/iam/actions"
)

// AssumeRoleScenario shows you how to use the AWS Identity and Access Management
 (IAM)
// service to perform the following actions:
//
//  1. Create a user who has no permissions.
//  2. Create a role that grants permission to list Amazon Simple Storage Service
//     (Amazon S3) buckets for the account.
//  3. Add a policy to let the user assume the role.
//  4. Try and fail to list buckets without permissions.
//  5. Assume the role and list S3 buckets using temporary credentials.
//  6. Delete the policy, role, and user.
type AssumeRoleScenario struct {
 sdkConfig      aws.Config
 accountWrapper actions.AccountWrapper
 policyWrapper  actions.PolicyWrapper
 roleWrapper    actions.RoleWrapper
 userWrapper    actions.UserWrapper
 questioner     demotools.IQuestioner
 helper         IScenarioHelper
 isTestRun      bool
}

// NewAssumeRoleScenario constructs an AssumeRoleScenario instance from a
 configuration.
// It uses the specified config to get an IAM client and create wrappers for the
 actions
// used in the scenario.
func NewAssumeRoleScenario(sdkConfig aws.Config, questioner
 demotools.IQuestioner,
```

```
 helper IScenarioHelper) AssumeRoleScenario {
 iamClient := iam.NewFromConfig(sdkConfig)
 return AssumeRoleScenario{
  sdkConfig:      sdkConfig,
  accountWrapper: actions.AccountWrapper{IamClient: iamClient},
  policyWrapper:  actions.PolicyWrapper{IamClient: iamClient},
  roleWrapper:    actions.RoleWrapper{IamClient: iamClient},
  userWrapper:    actions.UserWrapper{IamClient: iamClient},
  questioner:     questioner,
  helper:         helper,
 }
}

// addTestOptions appends the API options specified in the original configuration
 to
// another configuration. This is used to attach the middleware stubber to
 clients
// that are constructed during the scenario, which is needed for unit testing.
func (scenario AssumeRoleScenario) addTestOptions(scenarioConfig *aws.Config) {
 if scenario.isTestRun {
  scenarioConfig.APIOptions = append(scenarioConfig.APIOptions,
 scenario.sdkConfig.APIOptions...)
 }
}

// Run runs the interactive scenario.
func (scenario AssumeRoleScenario) Run(ctx context.Context) {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong with the demo.\n")
   log.Println(r)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Println("Welcome to the AWS Identity and Access Management (IAM) assume role
 demo.")
 log.Println(strings.Repeat("-", 88))

 user := scenario.CreateUser(ctx)
 accessKey := scenario.CreateAccessKey(ctx, user)
 role := scenario.CreateRoleAndPolicies(ctx, user)
 noPermsConfig := scenario.ListBucketsWithoutPermissions(ctx, accessKey)
 scenario.ListBucketsWithAssumedRole(ctx, noPermsConfig, role)
```

```
 scenario.Cleanup(ctx, user, role)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}

// CreateUser creates a new IAM user. This user has no permissions.
func (scenario AssumeRoleScenario) CreateUser(ctx context.Context) *types.User {
 log.Println("Let's create an example user with no permissions.")
 userName := scenario.questioner.Ask("Enter a name for the example user:",
 demotools.NotEmpty{})
 user, err := scenario.userWrapper.GetUser(ctx, userName)
 if err != nil {
  panic(err)
 }
 if user == nil {
  user, err = scenario.userWrapper.CreateUser(ctx, userName)
  if err != nil {
   panic(err)
  }
  log.Printf("Created user %v.\n", *user.UserName)
 } else {
  log.Printf("User %v already exists.\n", *user.UserName)
 }
 log.Println(strings.Repeat("-", 88))
 return user
}

// CreateAccessKey creates an access key for the user.
func (scenario AssumeRoleScenario) CreateAccessKey(ctx context.Context, user
 *types.User) *types.AccessKey {
 accessKey, err := scenario.userWrapper.CreateAccessKeyPair(ctx, *user.UserName)
 if err != nil {
  panic(err)
 }
 log.Printf("Created access key %v for your user.", *accessKey.AccessKeyId)
 log.Println("Waiting a few seconds for your user to be ready...")
 scenario.helper.Pause(10)
 log.Println(strings.Repeat("-", 88))
 return accessKey
}
```

```
// CreateRoleAndPolicies creates a policy that grants permission to list S3
 buckets for
// the current account and attaches the policy to a newly created role. It also
 adds an
// inline policy to the specified user that grants the user permission to assume
 the role.
func (scenario AssumeRoleScenario) CreateRoleAndPolicies(ctx context.Context,
 user *types.User) *types.Role {
 log.Println("Let's create a role and policy that grant permission to list S3
 buckets.")
 scenario.questioner.Ask("Press Enter when you're ready.")
 listBucketsRole, err := scenario.roleWrapper.CreateRole(ctx,
 scenario.helper.GetName(), *user.Arn)
 if err != nil {
  panic(err)
 }
 log.Printf("Created role %v.\n", *listBucketsRole.RoleName)
 listBucketsPolicy, err := scenario.policyWrapper.CreatePolicy(
  ctx, scenario.helper.GetName(), []string{"s3:ListAllMyBuckets"},
 "arn:aws:s3:::*")
 if err != nil {
  panic(err)
 }
 log.Printf("Created policy %v.\n", *listBucketsPolicy.PolicyName)
 err = scenario.roleWrapper.AttachRolePolicy(ctx, *listBucketsPolicy.Arn,
 *listBucketsRole.RoleName)
 if err != nil {
  panic(err)
 }
 log.Printf("Attached policy %v to role %v.\n", *listBucketsPolicy.PolicyName,
  *listBucketsRole.RoleName)
 err = scenario.userWrapper.CreateUserPolicy(ctx, *user.UserName,
 scenario.helper.GetName(),
  []string{"sts:AssumeRole"}, *listBucketsRole.Arn)
 if err != nil {
  panic(err)
 }
 log.Printf("Created an inline policy for user %v that lets the user assume the
 role.\n",
  *user.UserName)
 log.Println("Let's give AWS a few seconds to propagate these new resources and
 connections...")
 scenario.helper.Pause(10)
 log.Println(strings.Repeat("-", 88))
```

```
 return listBucketsRole
}

// ListBucketsWithoutPermissions creates an Amazon S3 client from the user's
 access key
// credentials and tries to list buckets for the account. Because the user does
 not have
// permission to perform this action, the action fails.
func (scenario AssumeRoleScenario) ListBucketsWithoutPermissions(ctx
 context.Context, accessKey *types.AccessKey) *aws.Config {
 log.Println("Let's try to list buckets without permissions. This should return
 an AccessDenied error.")
 scenario.questioner.Ask("Press Enter when you're ready.")
 noPermsConfig, err := config.LoadDefaultConfig(ctx,
  config.WithCredentialsProvider(credentials.NewStaticCredentialsProvider(
   *accessKey.AccessKeyId, *accessKey.SecretAccessKey, ""),
  ))
 if err != nil {
  panic(err)
 }

 // Add test options if this is a test run. This is needed only for testing
 purposes.
 scenario.addTestOptions(&noPermsConfig)

 s3Client := s3.NewFromConfig(noPermsConfig)
 _, err = s3Client.ListBuckets(ctx, &s3.ListBucketsInput{})
 if err != nil {
  // The SDK for Go does not model the AccessDenied error, so check ErrorCode
 directly.
  var ae smithy.APIError
  if errors.As(err, &ae) {
   switch ae.ErrorCode() {
   case "AccessDenied":
    log.Println("Got AccessDenied error, which is the expected result because\n"
 +
     "the ListBuckets call was made without permissions.")
   default:
    log.Println("Expected AccessDenied, got something else.")
    panic(err)
   }
  }
 } else {
```

```
  log.Println("Expected AccessDenied error when calling ListBuckets without
 permissions,\n" +
   "but the call succeeded. Continuing the example anyway...")
 }
 log.Println(strings.Repeat("-", 88))
 return &noPermsConfig
}

// ListBucketsWithAssumedRole performs the following actions:
//
//  1. Creates an AWS Security Token Service (AWS STS) client from the config
 created from
//     the user's access key credentials.
//  2. Gets temporary credentials by assuming the role that grants permission to
 list the
//     buckets.
//  3. Creates an Amazon S3 client from the temporary credentials.
//  4. Lists buckets for the account. Because the temporary credentials are
 generated by
//     assuming the role that grants permission, the action succeeds.
func (scenario AssumeRoleScenario) ListBucketsWithAssumedRole(ctx
 context.Context, noPermsConfig *aws.Config, role *types.Role) {
 log.Println("Let's assume the role that grants permission to list buckets and
 try again.")
 scenario.questioner.Ask("Press Enter when you're ready.")
 stsClient := sts.NewFromConfig(*noPermsConfig)
 tempCredentials, err := stsClient.AssumeRole(ctx, &sts.AssumeRoleInput{
  RoleArn:         role.Arn,
  RoleSessionName: aws.String("AssumeRoleExampleSession"),
  DurationSeconds: aws.Int32(900),
 })
 if err != nil {
  log.Printf("Couldn't assume role %v.\n", *role.RoleName)
  panic(err)
 }
 log.Printf("Assumed role %v, got temporary credentials.\n", *role.RoleName)
 assumeRoleConfig, err := config.LoadDefaultConfig(ctx,
  config.WithCredentialsProvider(credentials.NewStaticCredentialsProvider(
   *tempCredentials.Credentials.AccessKeyId,
   *tempCredentials.Credentials.SecretAccessKey,
   *tempCredentials.Credentials.SessionToken),
  ),
 )
 if err != nil {
```

```
  panic(err)
 }

 // Add test options if this is a test run. This is needed only for testing
 purposes.
 scenario.addTestOptions(&assumeRoleConfig)

 s3Client := s3.NewFromConfig(assumeRoleConfig)
 result, err := s3Client.ListBuckets(ctx, &s3.ListBucketsInput{})
 if err != nil {
  log.Println("Couldn't list buckets with assumed role credentials.")
  panic(err)
 }
 log.Println("Successfully called ListBuckets with assumed role credentials, \n"
 +
  "here are some of them:")
 for i := 0; i < len(result.Buckets) && i < 5; i++ {
  log.Printf("\t%v\n", *result.Buckets[i].Name)
 }
 log.Println(strings.Repeat("-", 88))
}

// Cleanup deletes all resources created for the scenario.
func (scenario AssumeRoleScenario) Cleanup(ctx context.Context, user *types.User,
 role *types.Role) {
 if scenario.questioner.AskBool(
  "Do you want to delete the resources created for this example? (y/n)", "y",
 ) {
  policies, err := scenario.roleWrapper.ListAttachedRolePolicies(ctx,
 *role.RoleName)
  if err != nil {
   panic(err)
  }
  for _, policy := range policies {
   err = scenario.roleWrapper.DetachRolePolicy(ctx, *role.RoleName,
 *policy.PolicyArn)
   if err != nil {
    panic(err)
   }
   err = scenario.policyWrapper.DeletePolicy(ctx, *policy.PolicyArn)
   if err != nil {
    panic(err)
   }
   log.Printf("Detached policy %v from role %v and deleted the policy.\n",
```

```
    *policy.PolicyName, *role.RoleName)
  }
  err = scenario.roleWrapper.DeleteRole(ctx, *role.RoleName)
  if err != nil {
   panic(err)
  }
  log.Printf("Deleted role %v.\n", *role.RoleName)

  userPols, err := scenario.userWrapper.ListUserPolicies(ctx, *user.UserName)
  if err != nil {
   panic(err)
  }
  for _, userPol := range userPols {
   err = scenario.userWrapper.DeleteUserPolicy(ctx, *user.UserName, userPol)
   if err != nil {
    panic(err)
   }
   log.Printf("Deleted policy %v from user %v.\n", userPol, *user.UserName)
  }
  keys, err := scenario.userWrapper.ListAccessKeys(ctx, *user.UserName)
  if err != nil {
   panic(err)
  }
  for _, key := range keys {
   err = scenario.userWrapper.DeleteAccessKey(ctx, *user.UserName,
 *key.AccessKeyId)
   if err != nil {
    panic(err)
   }
   log.Printf("Deleted access key %v from user %v.\n", *key.AccessKeyId,
 *user.UserName)
  }
  err = scenario.userWrapper.DeleteUser(ctx, *user.UserName)
  if err != nil {
   panic(err)
  }
  log.Printf("Deleted user %v.\n", *user.UserName)
  log.Println(strings.Repeat("-", 88))
 }

}

// IScenarioHelper abstracts input and wait functions from a scenario so that
 they
```

```
// can be mocked for unit testing.
type IScenarioHelper interface {
 GetName() string
 Pause(secs int)
}

const rMax = 100000

type ScenarioHelper struct {
 Prefix string
 Random *rand.Rand
}

// GetName returns a unique name formed of a prefix and a random number.
func (helper *ScenarioHelper) GetName() string {
 return fmt.Sprintf("%v%v", helper.Prefix, helper.Random.Intn(rMax))
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
 time.Sleep(time.Duration(secs) * time.Second)
}
```

Definisci una struttura che racchiude le azioni dell'account.

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// AccountWrapper encapsulates AWS Identity and Access Management (IAM) account
 actions
// used in the examples.
// It contains an IAM service client that is used to perform account actions.
type AccountWrapper struct {
 IamClient *iam.Client
}
```

```
// GetAccountPasswordPolicy gets the account password policy for the current
 account.
// If no policy has been set, a NoSuchEntityException is error is returned.
func (wrapper AccountWrapper) GetAccountPasswordPolicy(ctx context.Context)
 (*types.PasswordPolicy, error) {
 var pwPolicy *types.PasswordPolicy
 result, err := wrapper.IamClient.GetAccountPasswordPolicy(ctx,
  &iam.GetAccountPasswordPolicyInput{})
 if err != nil {
  log.Printf("Couldn't get account password policy. Here's why: %v\n", err)
 } else {
  pwPolicy = result.PasswordPolicy
 }
 return pwPolicy, err
}



// ListSAMLProviders gets the SAML providers for the account.
func (wrapper AccountWrapper) ListSAMLProviders(ctx context.Context)
 ([]types.SAMLProviderListEntry, error) {
 var providers []types.SAMLProviderListEntry
 result, err := wrapper.IamClient.ListSAMLProviders(ctx,
 &iam.ListSAMLProvidersInput{})
 if err != nil {
  log.Printf("Couldn't list SAML providers. Here's why: %v\n", err)
 } else {
  providers = result.SAMLProviderList
 }
 return providers, err
}
```

Definisci una struttura che racchiude le azioni della policy.

```
import (
 "context"
 "encoding/json"
```

```
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// ListPolicies gets up to maxPolicies policies.
func (wrapper PolicyWrapper) ListPolicies(ctx context.Context, maxPolicies int32)
 ([]types.Policy, error) {
 var policies []types.Policy
 result, err := wrapper.IamClient.ListPolicies(ctx, &iam.ListPoliciesInput{
  MaxItems: aws.Int32(maxPolicies),
 })
 if err != nil {
  log.Printf("Couldn't list policies. Here's why: %v\n", err)
 } else {
  policies = result.Policies
 }
 return policies, err
}



// PolicyDocument defines a policy document as a Go struct that can be serialized
// to JSON.
type PolicyDocument struct {
 Version   string
 Statement []PolicyStatement
}

// PolicyStatement defines a statement in a policy document.
type PolicyStatement struct {
 Effect    string
```

```
 Action    []string
 Principal map[string]string `json:",omitempty"`
 Resource  *string           `json:",omitempty"`
}

// CreatePolicy creates a policy that grants a list of actions to the specified
 resource.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper PolicyWrapper) CreatePolicy(ctx context.Context, policyName string,
 actions []string,
 resourceArn string) (*types.Policy, error) {
 var policy *types.Policy
 policyDoc := PolicyDocument{
  Version: "2012-10-17",
  Statement: []PolicyStatement{{
   Effect:   "Allow",
   Action:   actions,
   Resource: aws.String(resourceArn),
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n",
 resourceArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreatePolicy(ctx, &iam.CreatePolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
 })
 if err != nil {
  log.Printf("Couldn't create policy %v. Here's why: %v\n", policyName, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}



// GetPolicy gets data about a policy.
```

```
func (wrapper PolicyWrapper) GetPolicy(ctx context.Context, policyArn string)
 (*types.Policy, error) {
 var policy *types.Policy
 result, err := wrapper.IamClient.GetPolicy(ctx, &iam.GetPolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't get policy %v. Here's why: %v\n", policyArn, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}



// DeletePolicy deletes a policy.
func (wrapper PolicyWrapper) DeletePolicy(ctx context.Context, policyArn string)
 error {
 _, err := wrapper.IamClient.DeletePolicy(ctx, &iam.DeletePolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't delete policy %v. Here's why: %v\n", policyArn, err)
 }
 return err
}
```

Definisci una struttura che racchiude le azioni del ruolo.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)
```

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// ListRoles gets up to maxRoles roles.
func (wrapper RoleWrapper) ListRoles(ctx context.Context, maxRoles int32)
 ([]types.Role, error) {
 var roles []types.Role
 result, err := wrapper.IamClient.ListRoles(ctx,
  &iam.ListRolesInput{MaxItems: aws.Int32(maxRoles)},
 )
 if err != nil {
  log.Printf("Couldn't list roles. Here's why: %v\n", err)
 } else {
  roles = result.Roles
 }
 return roles, err
}



// CreateRole creates a role that trusts a specified user. The trusted user can
 assume
// the role to acquire its permissions.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper RoleWrapper) CreateRole(ctx context.Context, roleName string,
 trustedUserArn string) (*types.Role, error) {
 var role *types.Role
 trustPolicy := PolicyDocument{
  Version: "2012-10-17",
  Statement: []PolicyStatement{{
   Effect:    "Allow",
   Principal: map[string]string{"AWS": trustedUserArn},
   Action:    []string{"sts:AssumeRole"},
  }},
 }
 policyBytes, err := json.Marshal(trustPolicy)
```

```
 if err != nil {
  log.Printf("Couldn't create trust policy for %v. Here's why: %v\n",
 trustedUserArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreateRole(ctx, &iam.CreateRoleInput{
  AssumeRolePolicyDocument: aws.String(string(policyBytes)),
  RoleName:                 aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't create role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// GetRole gets data about a role.
func (wrapper RoleWrapper) GetRole(ctx context.Context, roleName string)
 (*types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.GetRole(ctx,
  &iam.GetRoleInput{RoleName: aws.String(roleName)})
 if err != nil {
  log.Printf("Couldn't get role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// CreateServiceLinkedRole creates a service-linked role that is owned by the
 specified service.
func (wrapper RoleWrapper) CreateServiceLinkedRole(ctx context.Context,
 serviceName string, description string) (
 *types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.CreateServiceLinkedRole(ctx,
 &iam.CreateServiceLinkedRoleInput{
  AWSServiceName: aws.String(serviceName),
```

```
  Description:    aws.String(description),
 })
 if err != nil {
  log.Printf("Couldn't create service-linked role %v. Here's why: %v\n",
 serviceName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// DeleteServiceLinkedRole deletes a service-linked role.
func (wrapper RoleWrapper) DeleteServiceLinkedRole(ctx context.Context, roleName
 string) error {
 _, err := wrapper.IamClient.DeleteServiceLinkedRole(ctx,
 &iam.DeleteServiceLinkedRoleInput{
  RoleName: aws.String(roleName)},
 )
 if err != nil {
  log.Printf("Couldn't delete service-linked role %v. Here's why: %v\n",
 roleName, err)
 }
 return err
}



// AttachRolePolicy attaches a policy to a role.
func (wrapper RoleWrapper) AttachRolePolicy(ctx context.Context, policyArn
 string, roleName string) error {
 _, err := wrapper.IamClient.AttachRolePolicy(ctx, &iam.AttachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't attach policy %v to role %v. Here's why: %v\n", policyArn,
 roleName, err)
 }
 return err
}
```

```
// ListAttachedRolePolicies lists the policies that are attached to the specified
 role.
func (wrapper RoleWrapper) ListAttachedRolePolicies(ctx context.Context, roleName
 string) ([]types.AttachedPolicy, error) {
 var policies []types.AttachedPolicy
 result, err := wrapper.IamClient.ListAttachedRolePolicies(ctx,
 &iam.ListAttachedRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list attached policies for role %v. Here's why: %v\n",
 roleName, err)
 } else {
  policies = result.AttachedPolicies
 }
 return policies, err
}



// DetachRolePolicy detaches a policy from a role.
func (wrapper RoleWrapper) DetachRolePolicy(ctx context.Context, roleName string,
 policyArn string) error {
 _, err := wrapper.IamClient.DetachRolePolicy(ctx, &iam.DetachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't detach policy from role %v. Here's why: %v\n", roleName,
 err)
 }
 return err
}



// ListRolePolicies lists the inline policies for a role.
func (wrapper RoleWrapper) ListRolePolicies(ctx context.Context, roleName string)
 ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListRolePolicies(ctx,
 &iam.ListRolePoliciesInput{
  RoleName: aws.String(roleName),
```

```
 })
 if err != nil {
  log.Printf("Couldn't list policies for role %v. Here's why: %v\n", roleName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}



// DeleteRole deletes a role. All attached policies must be detached before a
// role can be deleted.
func (wrapper RoleWrapper) DeleteRole(ctx context.Context, roleName string) error
 {
 _, err := wrapper.IamClient.DeleteRole(ctx, &iam.DeleteRoleInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't delete role %v. Here's why: %v\n", roleName, err)
 }
 return err
}
```

Definisci una struttura che racchiude le azioni dell'utente.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)
```

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListUsers gets up to maxUsers number of users.
func (wrapper UserWrapper) ListUsers(ctx context.Context, maxUsers int32)
 ([]types.User, error) {
 var users []types.User
 result, err := wrapper.IamClient.ListUsers(ctx, &iam.ListUsersInput{
  MaxItems: aws.Int32(maxUsers),
 })
 if err != nil {
  log.Printf("Couldn't list users. Here's why: %v\n", err)
 } else {
  users = result.Users
 }
 return users, err
}



// GetUser gets data about a user.
func (wrapper UserWrapper) GetUser(ctx context.Context, userName string)
 (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.GetUser(ctx, &iam.GetUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  var apiError smithy.APIError
  if errors.As(err, &apiError) {
   switch apiError.(type) {
   case *types.NoSuchEntityException:
    log.Printf("User %v does not exist.\n", userName)
    err = nil
   default:
    log.Printf("Couldn't get user %v. Here's why: %v\n", userName, err)
   }
  }
 } else {
```

```
  user = result.User
 }
 return user, err
}



// CreateUser creates a new user with the specified name.
func (wrapper UserWrapper) CreateUser(ctx context.Context, userName string)
 (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.CreateUser(ctx, &iam.CreateUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
 } else {
  user = result.User
 }
 return user, err
}



// CreateUserPolicy adds an inline policy to a user. This example creates a
 policy that
// grants a list of actions on a specified role.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper UserWrapper) CreateUserPolicy(ctx context.Context, userName string,
 policyName string, actions []string,
 roleArn string) error {
 policyDoc := PolicyDocument{
  Version: "2012-10-17",
  Statement: []PolicyStatement{{
   Effect:   "Allow",
   Action:   actions,
   Resource: aws.String(roleArn),
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
```

```
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n", roleArn,
 err)
  return err
 }
 _, err = wrapper.IamClient.PutUserPolicy(ctx, &iam.PutUserPolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
  UserName:       aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create policy for user %v. Here's why: %v\n", userName,
 err)
 }
 return err
}



// ListUserPolicies lists the inline policies for the specified user.
func (wrapper UserWrapper) ListUserPolicies(ctx context.Context, userName string)
 ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListUserPolicies(ctx,
 &iam.ListUserPoliciesInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list policies for user %v. Here's why: %v\n", userName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}



// DeleteUserPolicy deletes an inline policy from a user.
func (wrapper UserWrapper) DeleteUserPolicy(ctx context.Context, userName string,
 policyName string) error {
 _, err := wrapper.IamClient.DeleteUserPolicy(ctx, &iam.DeleteUserPolicyInput{
  PolicyName: aws.String(policyName),
  UserName:   aws.String(userName),
 })
```

```
 if err != nil {
  log.Printf("Couldn't delete policy from user %v. Here's why: %v\n", userName,
 err)
 }
 return err
}



// DeleteUser deletes a user.
func (wrapper UserWrapper) DeleteUser(ctx context.Context, userName string) error
 {
 _, err := wrapper.IamClient.DeleteUser(ctx, &iam.DeleteUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete user %v. Here's why: %v\n", userName, err)
 }
 return err
}



// CreateAccessKeyPair creates an access key for a user. The returned access key
 contains
// the ID and secret credentials needed to use the key.
func (wrapper UserWrapper) CreateAccessKeyPair(ctx context.Context, userName
 string) (*types.AccessKey, error) {
 var key *types.AccessKey
 result, err := wrapper.IamClient.CreateAccessKey(ctx, &iam.CreateAccessKeyInput{
  UserName: aws.String(userName)})
 if err != nil {
  log.Printf("Couldn't create access key pair for user %v. Here's why: %v\n",
 userName, err)
 } else {
  key = result.AccessKey
 }
 return key, err
}



// DeleteAccessKey deletes an access key from a user.
```

```
func (wrapper UserWrapper) DeleteAccessKey(ctx context.Context, userName string,
 keyId string) error {
 _, err := wrapper.IamClient.DeleteAccessKey(ctx, &iam.DeleteAccessKeyInput{
  AccessKeyId: aws.String(keyId),
  UserName:    aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete access key %v. Here's why: %v\n", keyId, err)
 }
 return err
}



// ListAccessKeys lists the access keys for the specified user.
func (wrapper UserWrapper) ListAccessKeys(ctx context.Context, userName string)
 ([]types.AccessKeyMetadata, error) {
 var keys []types.AccessKeyMetadata
 result, err := wrapper.IamClient.ListAccessKeys(ctx, &iam.ListAccessKeysInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list access keys for user %v. Here's why: %v\n", userName,
 err)
 } else {
  keys = result.AccessKeyMetadata
 }
 return keys, err
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Go .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey

- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
/*
  To run this Java V2 code example, set up your development environment,
 including your credentials.

  For information, see this documentation topic:

  https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html

  This example performs these operations:

  1. Creates a user that has no permissions.
  2. Creates a role and policy that grants Amazon S3 permissions.
  3. Creates a role.
  4. Grants the user permissions.
  5. Gets temporary credentials by assuming the role.  Creates an Amazon S3
 Service client object with the temporary credentials.
  6. Deletes the resources.
 */
```

```
public class IAMScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");
    public static final String PolicyDocument = "{" +
            "  \"Version\": \"2012-10-17\"," +
            "  \"Statement\": [" +
            "    {" +
            "        \"Effect\": \"Allow\"," +
            "        \"Action\": [" +
            "            \"s3:*\"" +
            "       ]," +
            "       \"Resource\": \"*\"" +
            "    }" +
            "   ]" +
            "}";

    public static String userArn;

    public static void main(String[] args) throws Exception {

        final String usage = """

                Usage:
                    <username> <policyName> <roleName> <roleSessionName>
 <bucketName>\s

                Where:
                    username - The name of the IAM user to create.\s
                    policyName - The name of the policy to create.\s
                    roleName - The name of the role to create.\s
                    roleSessionName - The name of the session required for the
 assumeRole operation.\s
                    bucketName - The name of the Amazon S3 bucket from which
 objects are read.\s
                """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String userName = args[0];
        String policyName = args[1];
        String roleName = args[2];
```

```
        String roleSessionName = args[3];
        String bucketName = args[4];

        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the AWS IAM example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 1. Create the IAM user.");
        User createUser = createIAMUser(iam, userName);

        System.out.println(DASHES);
        userArn = createUser.arn();

        AccessKey myKey = createIAMAccessKey(iam, userName);
        String accessKey = myKey.accessKeyId();
        String secretKey = myKey.secretAccessKey();
        String assumeRolePolicyDocument = "{" +
                "\"Version\": \"2012-10-17\"," +
                "\"Statement\": [{" +
                "\"Effect\": \"Allow\"," +
                "\"Principal\": {" +
                " \"AWS\": \"" + userArn + "\"" +
                "}," +
                "\"Action\": \"sts:AssumeRole\"" +
                "}]" +
                "}";

        System.out.println(assumeRolePolicyDocument);
        System.out.println(userName + " was successfully created.");
        System.out.println(DASHES);
        System.out.println("2. Creates a policy.");
        String polArn = createIAMPolicy(iam, policyName);
        System.out.println("The policy " + polArn + " was successfully
created.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Creates a role.");
```

```
        TimeUnit.SECONDS.sleep(30);
        String roleArn = createIAMRole(iam, roleName, assumeRolePolicyDocument);
        System.out.println(roleArn + " was successfully created.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Grants the user permissions.");
        attachIAMRolePolicy(iam, roleName, polArn);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Wait for 30 secs so the resource is available");
        TimeUnit.SECONDS.sleep(30);
        System.out.println("5. Gets temporary credentials by assuming the
role.");
        System.out.println("Perform an Amazon S3 Service operation using the
temporary credentials.");
        assumeRole(roleArn, roleSessionName, bucketName, accessKey, secretKey);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6 Getting ready to delete the AWS resources");
        deleteKey(iam, userName, accessKey);
        deleteRole(iam, roleName, polArn);
        deleteIAMUser(iam, userName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("This IAM Scenario has successfully completed");
        System.out.println(DASHES);
    }

    public static AccessKey createIAMAccessKey(IamClient iam, String user) {
        try {
            CreateAccessKeyRequest request = CreateAccessKeyRequest.builder()
                    .userName(user)
                    .build();

            CreateAccessKeyResponse response = iam.createAccessKey(request);
            return response.accessKey();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
        return null;
    }

    public static User createIAMUser(IamClient iam, String username) {
        try {
            // Create an IamWaiter object
            IamWaiter iamWaiter = iam.waiter();
            CreateUserRequest request = CreateUserRequest.builder()
                    .userName(username)
                    .build();

            // Wait until the user is created.
            CreateUserResponse response = iam.createUser(request);
            GetUserRequest userRequest = GetUserRequest.builder()
                    .userName(response.user().userName())
                    .build();

            WaiterResponse<GetUserResponse> waitUntilUserExists =
iamWaiter.waitUntilUserExists(userRequest);

waitUntilUserExists.matched().response().ifPresent(System.out::println);
            return response.user();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static String createIAMRole(IamClient iam, String rolename, String
json) {

        try {
            CreateRoleRequest request = CreateRoleRequest.builder()
                    .roleName(rolename)
                    .assumeRolePolicyDocument(json)
                    .description("Created using the AWS SDK for Java")
                    .build();

            CreateRoleResponse response = iam.createRole(request);
            System.out.println("The ARN of the role is " +
response.role().arn());
```

```
            return response.role().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

    public static String createIAMPolicy(IamClient iam, String policyName) {
        try {
            // Create an IamWaiter object.
            IamWaiter iamWaiter = iam.waiter();
            CreatePolicyRequest request = CreatePolicyRequest.builder()
                    .policyName(policyName)
                    .policyDocument(PolicyDocument).build();

            CreatePolicyResponse response = iam.createPolicy(request);
            GetPolicyRequest polRequest = GetPolicyRequest.builder()
                    .policyArn(response.policy().arn())
                    .build();

            WaiterResponse<GetPolicyResponse> waitUntilPolicyExists =
iamWaiter.waitUntilPolicyExists(polRequest);

waitUntilPolicyExists.matched().response().ifPresent(System.out::println);
            return response.policy().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

    public static void attachIAMRolePolicy(IamClient iam, String roleName, String
policyArn) {
        try {
            ListAttachedRolePoliciesRequest request =
ListAttachedRolePoliciesRequest.builder()
                    .roleName(roleName)
                    .build();
```

```
            ListAttachedRolePoliciesResponse response =
iam.listAttachedRolePolicies(request);
            List<AttachedPolicy> attachedPolicies = response.attachedPolicies();
            String polArn;
            for (AttachedPolicy policy : attachedPolicies) {
                polArn = policy.policyArn();
                if (polArn.compareTo(policyArn) == 0) {
                    System.out.println(roleName + " policy is already attached to
this role.");
                    return;
                }
            }

            AttachRolePolicyRequest attachRequest =
AttachRolePolicyRequest.builder()
                    .roleName(roleName)
                    .policyArn(policyArn)
                    .build();

            iam.attachRolePolicy(attachRequest);
            System.out.println("Successfully attached policy " + policyArn + " to
role " + roleName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Invoke an Amazon S3 operation using the Assumed Role.
    public static void assumeRole(String roleArn, String roleSessionName, String
bucketName, String keyVal,
            String keySecret) {

        // Use the creds of the new IAM user that was created in this code
example.
        AwsBasicCredentials credentials = AwsBasicCredentials.create(keyVal,
keySecret);
        StsClient stsClient = StsClient.builder()
                .region(Region.US_EAST_1)

.credentialsProvider(StaticCredentialsProvider.create(credentials))
                .build();
```

```
        try {
            AssumeRoleRequest roleRequest = AssumeRoleRequest.builder()
                    .roleArn(roleArn)
                    .roleSessionName(roleSessionName)
                    .build();

            AssumeRoleResponse roleResponse = stsClient.assumeRole(roleRequest);
            Credentials myCreds = roleResponse.credentials();
            String key = myCreds.accessKeyId();
            String secKey = myCreds.secretAccessKey();
            String secToken = myCreds.sessionToken();

            // List all objects in an Amazon S3 bucket using the temp creds
retrieved by
            // invoking assumeRole.
            Region region = Region.US_EAST_1;
            S3Client s3 = S3Client.builder()
                    .credentialsProvider(

StaticCredentialsProvider.create(AwsSessionCredentials.create(key, secKey,
secToken)))
                    .region(region)
                    .build();

            System.out.println("Created a S3Client using temp credentials.");
            System.out.println("Listing objects in " + bucketName);
            ListObjectsRequest listObjects = ListObjectsRequest.builder()
                    .bucket(bucketName)
                    .build();

            ListObjectsResponse res = s3.listObjects(listObjects);
            List<S3Object> objects = res.contents();
            for (S3Object myValue : objects) {
                System.out.println("The name of the key is " + myValue.key());
                System.out.println("The owner is " + myValue.owner());
            }

        } catch (StsException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

```
   public static void deleteRole(IamClient iam, String roleName, String polArn)
{

       try {
           // First the policy needs to be detached.
           DetachRolePolicyRequest rolePolicyRequest =
DetachRolePolicyRequest.builder()
                   .policyArn(polArn)
                   .roleName(roleName)
                   .build();

           iam.detachRolePolicy(rolePolicyRequest);

           // Delete the policy.
           DeletePolicyRequest request = DeletePolicyRequest.builder()
                   .policyArn(polArn)
                   .build();

           iam.deletePolicy(request);
           System.out.println("*** Successfully deleted " + polArn);

           // Delete the role.
           DeleteRoleRequest roleRequest = DeleteRoleRequest.builder()
                   .roleName(roleName)
                   .build();

           iam.deleteRole(roleRequest);
           System.out.println("*** Successfully deleted " + roleName);

       } catch (IamException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }

   public static void deleteKey(IamClient iam, String username, String
accessKey) {
       try {
           DeleteAccessKeyRequest request = DeleteAccessKeyRequest.builder()
                   .accessKeyId(accessKey)
                   .userName(username)
                   .build();

           iam.deleteAccessKey(request);
```

```
            System.out.println("Successfully deleted access key " + accessKey +
                    " from user " + username);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteIAMUser(IamClient iam, String userName) {
        try {
            DeleteUserRequest request = DeleteUserRequest.builder()
                    .userName(userName)
                    .build();

            iam.deleteUser(request);
            System.out.println("*** Successfully deleted " + userName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK for Java 2.x .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy

- PutUserPolicy

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
import {
  CreateUserCommand,
  GetUserCommand,
  CreateAccessKeyCommand,
  CreatePolicyCommand,
  CreateRoleCommand,
  AttachRolePolicyCommand,
  DeleteAccessKeyCommand,
  DeleteUserCommand,
  DeleteRoleCommand,
  DeletePolicyCommand,
  DetachRolePolicyCommand,
  IAMClient,
} from "@aws-sdk/client-iam";
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";
import { AssumeRoleCommand, STSClient } from "@aws-sdk/client-sts";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import { ScenarioInput } from "@aws-doc-sdk-examples/lib/scenario/index.js";

// Set the parameters.
const iamClient = new IAMClient({});
const userName = "iam_basic_test_username";
const policyName = "iam_basic_test_policy";
const roleName = "iam_basic_test_role";
```

```
/**
 * Create a new IAM user. If the user already exists, give
 * the option to delete and re-create it.
 * @param {string} name
 */
export const createUser = async (name, confirmAll = false) => {
  try {
    const { User } = await iamClient.send(
      new GetUserCommand({ UserName: name }),
    );
    const input = new ScenarioInput(
      "deleteUser",
      "Do you want to delete and remake this user?",
      { type: "confirm" },
    );
    const deleteUser = await input.handle({}, { confirmAll });
    // If the user exists, and you want to delete it, delete the user
    // and then create it again.
    if (deleteUser) {
      await iamClient.send(new DeleteUserCommand({ UserName: User.UserName }));
      await iamClient.send(new CreateUserCommand({ UserName: name }));
    } else {
      console.warn(
        `${name} already exists. The scenario may not work as expected.`,
      );
      return User;
    }
  } catch (caught) {
    // If there is no user by that name, create one.
    if (caught instanceof Error && caught.name === "NoSuchEntityException") {
      const { User } = await iamClient.send(
        new CreateUserCommand({ UserName: name }),
      );
      return User;
    }
    throw caught;
  }
};

export const main = async (confirmAll = false) => {
  // Create a user. The user has no permissions by default.
  const User = await createUser(userName, confirmAll);

  if (!User) {
```

```
    throw new Error("User not created");
  }

  // Create an access key. This key is used to authenticate the new user to
  // Amazon Simple Storage Service (Amazon S3) and AWS Security Token Service
(AWS STS).
  // It's not best practice to use access keys. For more information, see
https://aws.amazon.com/iam/resources/best-practices/.
  const createAccessKeyResponse = await iamClient.send(
    new CreateAccessKeyCommand({ UserName: userName }),
  );

  if (
    !createAccessKeyResponse.AccessKey?.AccessKeyId ||
    !createAccessKeyResponse.AccessKey?.SecretAccessKey
  ) {
    throw new Error("Access key not created");
  }

  const {
    AccessKey: { AccessKeyId, SecretAccessKey },
  } = createAccessKeyResponse;

  let s3Client = new S3Client({
    credentials: {
      accessKeyId: AccessKeyId,
      secretAccessKey: SecretAccessKey,
    },
  });

  // Retry the list buckets operation until it succeeds. InvalidAccessKeyId is
  // thrown while the user and access keys are still stabilizing.
  await retry({ intervalInMs: 1000, maxRetries: 300 }, async () => {
    try {
      return await listBuckets(s3Client);
    } catch (err) {
      if (err instanceof Error && err.name === "InvalidAccessKeyId") {
        throw err;
      }
    }
  });

  // Retry the create role operation until it succeeds. A MalformedPolicyDocument
error
```

```
// is thrown while the user and access keys are still stabilizing.
const { Role } = await retry(
  {
    intervalInMs: 2000,
    maxRetries: 60,
  },
  () =>
    iamClient.send(
      new CreateRoleCommand({
        AssumeRolePolicyDocument: JSON.stringify({
          Version: "2012-10-17",
          Statement: [
            {
              Effect: "Allow",
              Principal: {
                // Allow the previously created user to assume this role.
                AWS: User.Arn,
              },
              Action: "sts:AssumeRole",
            },
          ],
        }),
        RoleName: roleName,
      }),
    ),
);

if (!Role) {
  throw new Error("Role not created");
}

// Create a policy that allows the user to list S3 buckets.
const { Policy: listBucketPolicy } = await iamClient.send(
  new CreatePolicyCommand({
    PolicyDocument: JSON.stringify({
      Version: "2012-10-17",
      Statement: [
        {
          Effect: "Allow",
          Action: ["s3:ListAllMyBuckets"],
          Resource: "*",
        },
      ],
    }),
```

```
    PolicyName: policyName,
  }),
);

if (!listBucketPolicy) {
  throw new Error("Policy not created");
}

// Attach the policy granting the 's3:ListAllMyBuckets' action to the role.
await iamClient.send(
  new AttachRolePolicyCommand({
    PolicyArn: listBucketPolicy.Arn,
    RoleName: Role.RoleName,
  }),
);

// Assume the role.
const stsClient = new STSClient({
  credentials: {
    accessKeyId: AccessKeyId,
    secretAccessKey: SecretAccessKey,
  },
});

// Retry the assume role operation until it succeeds.
const { Credentials } = await retry(
  { intervalInMs: 2000, maxRetries: 60 },
  () =>
    stsClient.send(
      new AssumeRoleCommand({
        RoleArn: Role.Arn,
        RoleSessionName: `iamBasicScenarioSession-${Math.floor(
          Math.random() * 1000000,
        )}`,
        DurationSeconds: 900,
      }),
    ),
);

if (!Credentials?.AccessKeyId || !Credentials?.SecretAccessKey) {
  throw new Error("Credentials not created");
}

s3Client = new S3Client({
```

```
    credentials: {
      accessKeyId: Credentials.AccessKeyId,
      secretAccessKey: Credentials.SecretAccessKey,
      sessionToken: Credentials.SessionToken,
    },
  });

  // List the S3 buckets again.
  // Retry the list buckets operation until it succeeds. AccessDenied might
  // be thrown while the role policy is still stabilizing.
  await retry({ intervalInMs: 2000, maxRetries: 120 }, () =>
    listBuckets(s3Client),
  );

  // Clean up.
  await iamClient.send(
    new DetachRolePolicyCommand({
      PolicyArn: listBucketPolicy.Arn,
      RoleName: Role.RoleName,
    }),
  );

  await iamClient.send(
    new DeletePolicyCommand({
      PolicyArn: listBucketPolicy.Arn,
    }),
  );

  await iamClient.send(
    new DeleteRoleCommand({
      RoleName: Role.RoleName,
    }),
  );

  await iamClient.send(
    new DeleteAccessKeyCommand({
      UserName: userName,
      AccessKeyId,
    }),
  );

  await iamClient.send(
    new DeleteUserCommand({
      UserName: userName,
```

```
    }),
  );
};

/**
 *
 * @param {S3Client} s3Client
 */
const listBuckets = async (s3Client) => {
  const { Buckets } = await s3Client.send(new ListBucketsCommand({}));

  if (!Buckets) {
    throw new Error("Buckets not listed");
  }

  console.log(Buckets.map((bucket) => bucket.Name).join("\n"));
};
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per JavaScript .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

Kotlin

SDK per Kotlin

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <username> <policyName> <roleName> <roleSessionName> <fileLocation>
 <bucketName>

    Where:
        username - The name of the IAM user to create.
        policyName - The name of the policy to create.
        roleName - The name of the role to create.
        roleSessionName - The name of the session required for the assumeRole
 operation.
        fileLocation - The file location to the JSON required to create the role
 (see Readme).
        bucketName - The name of the Amazon S3 bucket from which objects are
 read.
    """

    if (args.size != 6) {
        println(usage)
        exitProcess(1)
    }

    val userName = args[0]
    val policyName = args[1]
    val roleName = args[2]
    val roleSessionName = args[3]
    val fileLocation = args[4]
    val bucketName = args[5]

    createUser(userName)
```

```
    println("$userName was successfully created.")

    val polArn = createPolicy(policyName)
    println("The policy $polArn was successfully created.")

    val roleArn = createRole(roleName, fileLocation)
    println("$roleArn was successfully created.")
    attachRolePolicy(roleName, polArn)

    println("*** Wait for 1 MIN so the resource is available.")
    delay(60000)
    assumeGivenRole(roleArn, roleSessionName, bucketName)

    println("*** Getting ready to delete the AWS resources.")
    deleteRole(roleName, polArn)
    deleteUser(userName)
    println("This IAM Scenario has successfully completed.")
}

suspend fun createUser(usernameVal: String?): String? {
    val request =
        CreateUserRequest {
            userName = usernameVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createUser(request)
        return response.user?.userName
    }
}

suspend fun createPolicy(policyNameVal: String?): String {
    val policyDocumentValue = """
    {
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "s3:*"
                ],
                "Resource": "*"
            }
        ]
```

```
    }
    """.trimIndent()

    val request =
        CreatePolicyRequest {
            policyName = policyNameVal
            policyDocument = policyDocumentValue
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createPolicy(request)
        return response.policy?.arn.toString()
    }
}

suspend fun createRole(
    rolenameVal: String?,
    fileLocation: String?,
): String? {
    val jsonObject = fileLocation?.let { readJsonSimpleDemo(it) } as JSONObject

    val request =
        CreateRoleRequest {
            roleName = rolenameVal
            assumeRolePolicyDocument = jsonObject.toJSONString()
            description = "Created using the AWS SDK for Kotlin"
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createRole(request)
        return response.role?.arn
    }
}

suspend fun attachRolePolicy(
    roleNameVal: String,
    policyArnVal: String,
) {
    val request =
        ListAttachedRolePoliciesRequest {
            roleName = roleNameVal
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
```

```
        val response = iamClient.listAttachedRolePolicies(request)
        val attachedPolicies = response.attachedPolicies

        // Ensure that the policy is not attached to this role.
        val checkStatus: Int
        if (attachedPolicies != null) {
            checkStatus = checkMyList(attachedPolicies, policyArnVal)
            if (checkStatus == -1) {
                return
            }
        }

        val policyRequest =
            AttachRolePolicyRequest {
                roleName = roleNameVal
                policyArn = policyArnVal
            }
        iamClient.attachRolePolicy(policyRequest)
        println("Successfully attached policy $policyArnVal to role
 $roleNameVal")
    }
}

fun checkMyList(
    attachedPolicies: List<AttachedPolicy>,
    policyArnVal: String,
): Int {
    for (policy in attachedPolicies) {
        val polArn = policy.policyArn.toString()

        if (polArn.compareTo(policyArnVal) == 0) {
            println("The policy is already attached to this role.")
            return -1
        }
    }
    return 0
}

suspend fun assumeGivenRole(
    roleArnVal: String?,
    roleSessionNameVal: String?,
    bucketName: String,
) {
    val stsClient = StsClient.fromEnvironment { region = "us-east-1" }
```

```
    val roleRequest =
        AssumeRoleRequest {
            roleArn = roleArnVal
            roleSessionName = roleSessionNameVal
        }

    val roleResponse = stsClient.assumeRole(roleRequest)
    val myCreds = roleResponse.credentials
    val key = myCreds?.accessKeyId
    val secKey = myCreds?.secretAccessKey
    val secToken = myCreds?.sessionToken

    val staticCredentials = StaticCredentialsProvider {
        accessKeyId = key
        secretAccessKey = secKey
        sessionToken = secToken
    }

    // List all objects in an Amazon S3 bucket using the temp creds.
    val s3 = S3Client.fromEnvironment {
        region = "us-east-1"
        credentialsProvider = staticCredentials
    }

    println("Created a S3Client using temp credentials.")
    println("Listing objects in $bucketName")

    val listObjects =
        ListObjectsRequest {
            bucket = bucketName
        }

    val response = s3.listObjects(listObjects)
    response.contents?.forEach { myObject ->
        println("The name of the key is ${myObject.key}")
        println("The owner is ${myObject.owner}")
    }
}

suspend fun deleteRole(
    roleNameVal: String,
    polArn: String,
) {
    val iam = IamClient.fromEnvironment { region = "AWS_GLOBAL" }
```

```
    // First the policy needs to be detached.
    val rolePolicyRequest =
        DetachRolePolicyRequest {
            policyArn = polArn
            roleName = roleNameVal
        }

    iam.detachRolePolicy(rolePolicyRequest)

    // Delete the policy.
    val request =
        DeletePolicyRequest {
            policyArn = polArn
        }

    iam.deletePolicy(request)
    println("*** Successfully deleted $polArn")

    // Delete the role.
    val roleRequest =
        DeleteRoleRequest {
            roleName = roleNameVal
        }

    iam.deleteRole(roleRequest)
    println("*** Successfully deleted $roleNameVal")
}

suspend fun deleteUser(userNameVal: String) {
    val iam = IamClient.fromEnvironment { region = "AWS_GLOBAL" }
    val request =
        DeleteUserRequest {
            userName = userNameVal
        }

    iam.deleteUser(request)
    println("*** Successfully deleted $userNameVal")
}

@Throws(java.lang.Exception::class)
fun readJsonSimpleDemo(filename: String): Any? {
    val reader = FileReader(filename)
    val jsonParser = JSONParser()
```

```
    return jsonParser.parse(reader)
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Kotlin.
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

PHP

SDK per PHP

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
namespace Iam\Basics;

require 'vendor/autoload.php';

use Aws\Credentials\Credentials;
use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;
```

```
use Aws\Sts\StsClient;
use Iam\IAMService;

echo("\n");
echo("--------------------------------------\n");
print("Welcome to the IAM getting started demo using PHP!\n");
echo("--------------------------------------\n");

$uuid = uniqid();
$service = new IAMService();

$user = $service->createUser("iam_demo_user_$uuid");
echo "Created user with the arn: {$user['Arn']}\n";

$key = $service->createAccessKey($user['UserName']);
$assumeRolePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{$user['Arn']}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }";
$assumeRoleRole = $service->createRole("iam_demo_role_$uuid",
 $assumeRolePolicyDocument);
echo "Created role: {$assumeRoleRole['RoleName']}\n";

$listAllBucketsPolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
}";
$listAllBucketsPolicy = $service->createPolicy("iam_demo_policy_$uuid",
 $listAllBucketsPolicyDocument);
echo "Created policy: {$listAllBucketsPolicy['PolicyName']}\n";

$service->attachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);

$inlinePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
```

```
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"{$assumeRoleRole['Arn']}\"}]
}";
$inlinePolicy = $service->createUserPolicy("iam_demo_inline_policy_$uuid",
 $inlinePolicyDocument, $user['UserName']);
//First, fail to list the buckets with the user
$credentials = new Credentials($key['AccessKeyId'], $key['SecretAccessKey']);
$s3Client = new S3Client(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $credentials]);
try {
    $s3Client->listBuckets([
    ]);
    echo "this should not run";
} catch (S3Exception $exception) {
    echo "successfully failed!\n";
}

$stsClient = new StsClient(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $credentials]);
sleep(10);
$assumedRole = $stsClient->assumeRole([
    'RoleArn' => $assumeRoleRole['Arn'],
    'RoleSessionName' => "DemoAssumeRoleSession_$uuid",
]);
$assumedCredentials = [
    'key' => $assumedRole['Credentials']['AccessKeyId'],
    'secret' => $assumedRole['Credentials']['SecretAccessKey'],
    'token' => $assumedRole['Credentials']['SessionToken'],
];
$s3Client = new S3Client(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $assumedCredentials]);
try {
    $s3Client->listBuckets([]);
    echo "this should now run!\n";
} catch (S3Exception $exception) {
    echo "this should now not fail\n";
}

$service->detachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);
$deletePolicy = $service->deletePolicy($listAllBucketsPolicy['Arn']);
echo "Delete policy: {$listAllBucketsPolicy['PolicyName']}\n";
$deletedRole = $service->deleteRole($assumeRoleRole['Arn']);
```

```
echo "Deleted role: {$assumeRoleRole['RoleName']}\n";
$deletedKey = $service->deleteAccessKey($key['AccessKeyId'], $user['UserName']);
$deletedUser = $service->deleteUser($user['UserName']);
echo "Delete user: {$user['UserName']}\n";
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per PHP .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
import json
import sys
import time
from uuid import uuid4

import boto3
from botocore.exceptions import ClientError


def progress_bar(seconds):
    """Shows a simple progress bar in the command window."""
    for _ in range(seconds):
        time.sleep(1)
        print(".", end="")
        sys.stdout.flush()
    print()


def setup(iam_resource):
    """
    Creates a new user with no permissions.
    Creates an access key pair for the user.
    Creates a role with a policy that lets the user assume the role.
    Creates a policy that allows listing Amazon S3 buckets.
    Attaches the policy to the role.
    Creates an inline policy for the user that lets the user assume the role.

    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
 resource
                         that has permissions to create users, roles, and
 policies
                         in the account.
    :return: The newly created user, user key, and role.
    """
    try:
        user = iam_resource.create_user(UserName=f"demo-user-{uuid4()}")
        print(f"Created user {user.name}.")
    except ClientError as error:
        print(
            f"Couldn't create a user for the demo. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise
```

```
    try:
        user_key = user.create_access_key_pair()
        print(f"Created access key pair for user.")
    except ClientError as error:
        print(
            f"Couldn't create access keys for user {user.name}. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    print(f"Wait for user to be ready.", end="")
    progress_bar(10)

    try:
        role = iam_resource.create_role(
            RoleName=f"demo-role-{uuid4()}",
            AssumeRolePolicyDocument=json.dumps(
                {
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {"AWS": user.arn},
                            "Action": "sts:AssumeRole",
                        }
                    ],
                }
            ),
        )
        print(f"Created role {role.name}.")
    except ClientError as error:
        print(
            f"Couldn't create a role for the demo. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        policy = iam_resource.create_policy(
            PolicyName=f"demo-policy-{uuid4()}",
            PolicyDocument=json.dumps(
                {
                    "Version":"2012-10-17",
```

```
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "s3:ListAllMyBuckets",
                            "Resource": "arn:aws:s3:::*",
                        }
                    ],
                }
            ),
        )
        role.attach_policy(PolicyArn=policy.arn)
        print(f"Created policy {policy.policy_name} and attached it to the
role.")
    except ClientError as error:
        print(
            f"Couldn't create a policy and attach it to role {role.name}. Here's
why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        user.create_policy(
            PolicyName=f"demo-user-policy-{uuid4()}",
            PolicyDocument=json.dumps(
                {
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "sts:AssumeRole",
                            "Resource": role.arn,
                        }
                    ],
                }
            ),
        )
        print(
            f"Created an inline policy for {user.name} that lets the user assume
"
            f"the role."
        )
    except ClientError as error:
        print(
```

```
            f"Couldn't create an inline policy for user {user.name}. Here's why:
 "
            f"{error.response['Error']['Message']}"
        )
        raise

    print("Give AWS time to propagate these new resources and connections.",
 end="")
    progress_bar(10)

    return user, user_key, role


def show_access_denied_without_role(user_key):
    """
    Shows that listing buckets without first assuming the role is not allowed.

    :param user_key: The key of the user created during setup. This user does not
                     have permission to list buckets in the account.
    """
    print(f"Try to list buckets without first assuming the role.")
    s3_denied_resource = boto3.resource(
        "s3", aws_access_key_id=user_key.id,
 aws_secret_access_key=user_key.secret
    )
    try:
        for bucket in s3_denied_resource.buckets.all():
            print(bucket.name)
        raise RuntimeError("Expected to get AccessDenied error when listing
 buckets!")
    except ClientError as error:
        if error.response["Error"]["Code"] == "AccessDenied":
            print("Attempt to list buckets with no permissions: AccessDenied.")
        else:
            raise


def list_buckets_from_assumed_role(user_key, assume_role_arn, session_name):
    """
    Assumes a role that grants permission to list the Amazon S3 buckets in the
 account.
    Uses the temporary credentials from the role to list the buckets that are
 owned
    by the assumed role's account.
```

```
    :param user_key: The access key of a user that has permission to assume the
role.
    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    """
    sts_client = boto3.client(
        "sts", aws_access_key_id=user_key.id,
aws_secret_access_key=user_key.secret
    )
    try:
        response = sts_client.assume_role(
            RoleArn=assume_role_arn, RoleSessionName=session_name
        )
        temp_credentials = response["Credentials"]
        print(f"Assumed role {assume_role_arn} and got temporary credentials.")
    except ClientError as error:
        print(
            f"Couldn't assume role {assume_role_arn}. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    # Create an S3 resource that can access the account with the temporary
credentials.
    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )
    print(f"Listing buckets for the assumed role's account:")
    try:
        for bucket in s3_resource.buckets.all():
            print(bucket.name)
    except ClientError as error:
        print(
            f"Couldn't list buckets for the account. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise
```

```
def teardown(user, role):
    """
    Removes all resources created during setup.

    :param user: The demo user.
    :param role: The demo role.
    """
    try:
        for attached in role.attached_policies.all():
            policy_name = attached.policy_name
            role.detach_policy(PolicyArn=attached.arn)
            attached.delete()
            print(f"Detached and deleted {policy_name}.")
        role.delete()
        print(f"Deleted {role.name}.")
    except ClientError as error:
        print(
            "Couldn't detach policy, delete policy, or delete role. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        for user_pol in user.policies.all():
            user_pol.delete()
            print("Deleted inline user policy.")
        for key in user.access_keys.all():
            key.delete()
            print("Deleted user's access key.")
        user.delete()
        print(f"Deleted {user.name}.")
    except ClientError as error:
        print(
            "Couldn't delete user policy or delete user. Here's why: "
            f"{error.response['Error']['Message']}"
        )


def usage_demo():
    """Drives the demonstration."""
    print("-" * 88)
    print(f"Welcome to the IAM create user and assume role demo.")
```

```
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    user = None
    role = None
    try:
        user, user_key, role = setup(iam_resource)
        print(f"Created {user.name} and {role.name}.")
        show_access_denied_without_role(user_key)
        list_buckets_from_assumed_role(user_key, role.arn,
 "AssumeRoleDemoSession")
    except Exception:
        print("Something went wrong!")
    finally:
        if user is not None and role is not None:
            teardown(user, role)
        print("Thanks for watching!")


if __name__ == "__main__":
    usage_demo()
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

Ruby

SDK per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
# Wraps the scenario actions.
class ScenarioCreateUserAssumeRole
  attr_reader :iam_client

  # @param [Aws::IAM::Client] iam_client: The AWS IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Waits for the specified number of seconds.
  #
  # @param duration [Integer] The number of seconds to wait.
  def wait(duration)
    puts('Give AWS time to propagate resources...')
    sleep(duration)
  end

  # Creates a user.
  #
  # @param user_name [String] The name to give the user.
  # @return [Aws::IAM::User] The newly created user.
  def create_user(user_name)
    user = @iam_client.create_user(user_name: user_name).user
    @logger.info("Created demo user named #{user.user_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info('Tried and failed to create demo user.')
    @logger.info("\t#{e.code}: #{e.message}")
```

```
    @logger.info("\nCan't continue the demo without a user!")
    raise
  else
    user
  end

  # Creates an access key for a user.
  #
  # @param user [Aws::IAM::User] The user that owns the key.
  # @return [Aws::IAM::AccessKeyPair] The newly created access key.
  def create_access_key_pair(user)
    user_key = @iam_client.create_access_key(user_name:
user.user_name).access_key
    @logger.info("Created accesskey pair for user #{user.user_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create access keys for user #{user.user_name}.")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  else
    user_key
  end

  # Creates a role that can be assumed by a user.
  #
  # @param role_name [String] The name to give the role.
  # @param user [Aws::IAM::User] The user who is granted permission to assume the
role.
  # @return [Aws::IAM::Role] The newly created role.
  def create_role(role_name, user)
    trust_policy = {
      Version: '2012-10-17',
      Statement: [{
        Effect: 'Allow',
        Principal: { 'AWS': user.arn },
        Action: 'sts:AssumeRole'
      }]
    }.to_json
    role = @iam_client.create_role(
      role_name: role_name,
      assume_role_policy_document: trust_policy
    ).role
    @logger.info("Created role #{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create a role for the demo. Here's why: ")
```

```
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  else
    role
  end

  # Creates a policy that grants permission to list S3 buckets in the account,
and
  # then attaches the policy to a role.
  #
  # @param policy_name [String] The name to give the policy.
  # @param role [Aws::IAM::Role] The role that the policy is attached to.
  # @return [Aws::IAM::Policy] The newly created policy.
  def create_and_attach_role_policy(policy_name, role)
    policy_document = {
      Version: '2012-10-17',
      Statement: [{
        Effect: 'Allow',
        Action: 's3:ListAllMyBuckets',
        Resource: 'arn:aws:s3:::*'
      }]
    }.to_json
    policy = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document
    ).policy
    @iam_client.attach_role_policy(
      role_name: role.role_name,
      policy_arn: policy.arn
    )
    @logger.info("Created policy #{policy.policy_name} and attached it to role
#{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create a policy and attach it to role
#{role.role_name}. Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  end

  # Creates an inline policy for a user that lets the user assume a role.
  #
  # @param policy_name [String] The name to give the policy.
  # @param user [Aws::IAM::User] The user that owns the policy.
  # @param role [Aws::IAM::Role] The role that can be assumed.
```

```
 # @return [Aws::IAM::UserPolicy] The newly created policy.
 def create_user_policy(policy_name, user, role)
   policy_document = {
     Version: '2012-10-17',
     Statement: [{
       Effect: 'Allow',
       Action: 'sts:AssumeRole',
       Resource: role.arn
     }]
   }.to_json
   @iam_client.put_user_policy(
     user_name: user.user_name,
     policy_name: policy_name,
     policy_document: policy_document
   )
   puts("Created an inline policy for #{user.user_name} that lets the user
assume role #{role.role_name}.")
 rescue Aws::Errors::ServiceError => e
   @logger.info("Couldn't create an inline policy for user #{user.user_name}.
Here's why: ")
   @logger.info("\t#{e.code}: #{e.message}")
   raise
 end

 # Creates an Amazon S3 resource with specified credentials. This is separated
into a
 # factory function so that it can be mocked for unit testing.
 #
 # @param credentials [Aws::Credentials] The credentials used by the Amazon S3
resource.
 def create_s3_resource(credentials)
   Aws::S3::Resource.new(client: Aws::S3::Client.new(credentials: credentials))
 end

 # Lists the S3 buckets for the account, using the specified Amazon S3 resource.
 # Because the resource uses credentials with limited access, it may not be able
to
 # list the S3 buckets.
 #
 # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
 def list_buckets(s3_resource)
   count = 10
   s3_resource.buckets.each do |bucket|
     @logger.info "\t#{bucket.name}"
```

```
      count -= 1
      break if count.zero?
    end
  rescue Aws::Errors::ServiceError => e
    if e.code == 'AccessDenied'
      puts('Attempt to list buckets with no permissions: AccessDenied.')
    else
      @logger.info("Couldn't list buckets for the account. Here's why: ")
      @logger.info("\t#{e.code}: #{e.message}")
      raise
    end
  end

  # Creates an AWS Security Token Service (AWS STS) client with specified
credentials.
  # This is separated into a factory function so that it can be mocked for unit
testing.
  #
  # @param key_id [String] The ID of the access key used by the STS client.
  # @param key_secret [String] The secret part of the access key used by the STS
client.
  def create_sts_client(key_id, key_secret)
    Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
  end

  # Gets temporary credentials that can be used to assume a role.
  #
  # @param role_arn [String] The ARN of the role that is assumed when these
credentials
  #                          are used.
  # @param sts_client [AWS::STS::Client] An AWS STS client.
  # @return [Aws::AssumeRoleCredentials] The credentials that can be used to
assume the role.
  def assume_role(role_arn, sts_client)
    credentials = Aws::AssumeRoleCredentials.new(
      client: sts_client,
      role_arn: role_arn,
      role_session_name: 'create-use-assume-role-scenario'
    )
    @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
    credentials
  end

  # Deletes a role. If the role has policies attached, they are detached and
```

```
  # deleted before the role is deleted.
  #
  # @param role_name [String] The name of the role to delete.
  def delete_role(role_name)
    @iam_client.list_attached_role_policies(role_name:
 role_name).attached_policies.each do |policy|
      @iam_client.detach_role_policy(role_name: role_name, policy_arn:
 policy.policy_arn)
      @iam_client.delete_policy(policy_arn: policy.policy_arn)
      @logger.info("Detached and deleted policy #{policy.policy_name}.")
    end
    @iam_client.delete_role({ role_name: role_name })
    @logger.info("Role deleted: #{role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't detach policies and delete role #{role.name}. Here's
 why:")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  end

  # Deletes a user. If the user has inline policies or access keys, they are
 deleted
  # before the user is deleted.
  #
  # @param user [Aws::IAM::User] The user to delete.
  def delete_user(user_name)
    user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
    user.each do |key|
      @iam_client.delete_access_key({ access_key_id: key.access_key_id,
 user_name: user_name })
      @logger.info("Deleted access key #{key.access_key_id} for user
 '#{user_name}'.")
    end

    @iam_client.delete_user(user_name: user_name)
    @logger.info("Deleted user '#{user_name}'.")
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting user '#{user_name}': #{e.message}")
  end
end

# Runs the IAM create a user and assume a role scenario.
def run_scenario(scenario)
  puts('-' * 88)
```

```
  puts('Welcome to the IAM create a user and assume a role demo!')
  puts('-' * 88)
  user = scenario.create_user("doc-example-user-#{Random.uuid}")
  user_key = scenario.create_access_key_pair(user)
  scenario.wait(10)
  role = scenario.create_role("doc-example-role-#{Random.uuid}", user)
  scenario.create_and_attach_role_policy("doc-example-role-policy-
#{Random.uuid}", role)
  scenario.create_user_policy("doc-example-user-policy-#{Random.uuid}", user,
 role)
  scenario.wait(10)
  puts('Try to list buckets with credentials for a user who has no permissions.')
  puts('Expect AccessDenied from this call.')
  scenario.list_buckets(
    scenario.create_s3_resource(Aws::Credentials.new(user_key.access_key_id,
 user_key.secret_access_key))
  )
  puts('Now, assume the role that grants permission.')
  temp_credentials = scenario.assume_role(
    role.arn, scenario.create_sts_client(user_key.access_key_id,
 user_key.secret_access_key)
  )
  puts('Here are your buckets:')
  scenario.list_buckets(scenario.create_s3_resource(temp_credentials))
  puts("Deleting role '#{role.role_name}' and attached policies.")
  scenario.delete_role(role.role_name)
  puts("Deleting user '#{user.user_name}', policies, and keys.")
  scenario.delete_user(user.user_name)
  puts('Thanks for watching!')
  puts('-' * 88)
rescue Aws::Errors::ServiceError => e
  puts('Something went wrong with the demo.')
  puts("\t#{e.code}: #{e.message}")
end

run_scenario(ScenarioCreateUserAssumeRole.new(Aws::IAM::Client.new)) if
 $PROGRAM_NAME == __FILE__
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Ruby .
  - AttachRolePolicy

- CreateAccessKey
- CreatePolicy
- CreateRole
- CreateUser
- DeleteAccessKey
- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
use aws_config::meta::region::RegionProviderChain;
use aws_sdk_iam::Error as iamError;
use aws_sdk_iam::{config::Credentials as iamCredentials, config::Region, Client
 as iamClient};
use aws_sdk_s3::Client as s3Client;
use aws_sdk_sts::Client as stsClient;
use tokio::time::{sleep, Duration};
use uuid::Uuid;

#[tokio::main]
async fn main() -> Result<(), iamError> {
    let (client, uuid, list_all_buckets_policy_document, inline_policy_document)
 =
        initialize_variables().await;
```

```
    if let Err(e) = run_iam_operations(
        client,
        uuid,
        list_all_buckets_policy_document,
        inline_policy_document,
    )
    .await
    {
        println!("{:?}", e);
    };

    Ok(())
}

async fn initialize_variables() -> (iamClient, String, String, String) {
    let region_provider = RegionProviderChain::first_try(Region::new("us-
west-2"));

    let shared_config =
 aws_config::from_env().region(region_provider).load().await;
    let client = iamClient::new(&shared_config);
    let uuid = Uuid::new_v4().to_string();

    let list_all_buckets_policy_document = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
    }"
    .to_string();
    let inline_policy_document = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"{}\"}]
    }"
    .to_string();

    (
        client,
        uuid,
```

```
        list_all_buckets_policy_document,
        inline_policy_document,
    )
}

async fn run_iam_operations(
    client: iamClient,
    uuid: String,
    list_all_buckets_policy_document: String,
    inline_policy_document: String,
) -> Result<(), iamError> {
    let user = iam_service::create_user(&client, &format!("{}{}",
 "iam_demo_user_", uuid)).await?;
    println!("Created the user with the name: {}", user.user_name());
    let key = iam_service::create_access_key(&client, user.user_name()).await?;

    let assume_role_policy_document = "{
        \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }"
    .to_string()
    .replace("{}", user.arn());

    let assume_role_role = iam_service::create_role(
        &client,
        &format!("{}{}", "iam_demo_role_", uuid),
        &assume_role_policy_document,
    )
    .await?;
    println!("Created the role with the ARN: {}", assume_role_role.arn());

    let list_all_buckets_policy = iam_service::create_policy(
        &client,
        &format!("{}{}", "iam_demo_policy_", uuid),
        &list_all_buckets_policy_document,
    )
    .await?;
    println!(
        "Created policy: {}",
        list_all_buckets_policy.policy_name.as_ref().unwrap()
```

```
    );

    let attach_role_policy_result =
        iam_service::attach_role_policy(&client, &assume_role_role,
&list_all_buckets_policy)
            .await?;
    println!(
        "Attached the policy to the role: {:?}",
        attach_role_policy_result
    );

    let inline_policy_name = format!("{}{}", "iam_demo_inline_policy_", uuid);
    let inline_policy_document = inline_policy_document.replace("{}",
assume_role_role.arn());
    iam_service::create_user_policy(&client, &user, &inline_policy_name,
&inline_policy_document)
        .await?;
    println!("Created inline policy.");

    //First, fail to list the buckets with the user.
    let creds = iamCredentials::from_keys(key.access_key_id(),
key.secret_access_key(), None);
    let fail_config = aws_config::from_env()
        .credentials_provider(creds.clone())
        .load()
        .await;
    println!("Fail config: {:?}", fail_config);
    let fail_client: s3Client = s3Client::new(&fail_config);
    match fail_client.list_buckets().send().await {
        Ok(e) => {
            println!("This should not run. {:?}", e);
        }
        Err(e) => {
            println!("Successfully failed with error: {:?}", e)
        }
    }

    let sts_config = aws_config::from_env()
        .credentials_provider(creds.clone())
        .load()
        .await;
    let sts_client: stsClient = stsClient::new(&sts_config);
    sleep(Duration::from_secs(10)).await;
    let assumed_role = sts_client
```

```
        .assume_role()
        .role_arn(assume_role_role.arn())
        .role_session_name(format!("iam_demo_assumerole_session_{uuid}"))
        .send()
        .await;
    println!("Assumed role: {:?}", assumed_role);
    sleep(Duration::from_secs(10)).await;

    let assumed_credentials = iamCredentials::from_keys(
        assumed_role
            .as_ref()
            .unwrap()
            .credentials
            .as_ref()
            .unwrap()
            .access_key_id(),
        assumed_role
            .as_ref()
            .unwrap()
            .credentials
            .as_ref()
            .unwrap()
            .secret_access_key(),
        Some(
            assumed_role
                .as_ref()
                .unwrap()
                .credentials
                .as_ref()
                .unwrap()
                .session_token
                .clone(),
        ),
    );

    let succeed_config = aws_config::from_env()
        .credentials_provider(assumed_credentials)
        .load()
        .await;
    println!("succeed config: {:?}", succeed_config);
    let succeed_client: s3Client = s3Client::new(&succeed_config);
    sleep(Duration::from_secs(10)).await;
    match succeed_client.list_buckets().send().await {
        Ok(_) => {
```

```
            println!("This should now run successfully.")
        }
        Err(e) => {
            println!("This should not run. {:?}", e);
            panic!()
        }
    }

    //Clean up.
    iam_service::detach_role_policy(
        &client,
        assume_role_role.role_name(),
        list_all_buckets_policy.arn().unwrap_or_default(),
    )
    .await?;
    iam_service::delete_policy(&client, list_all_buckets_policy).await?;
    iam_service::delete_role(&client, &assume_role_role).await?;
    println!("Deleted role {}", assume_role_role.role_name());
    iam_service::delete_access_key(&client, &user, &key).await?;
    println!("Deleted key for {}", key.user_name());
    iam_service::delete_user_policy(&client, &user, &inline_policy_name).await?;
    println!("Deleted inline user policy: {}", inline_policy_name);
    iam_service::delete_user(&client, &user).await?;
    println!("Deleted user {}", user.user_name());

    Ok(())
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API SDK AWS per Rust.
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole

- DeleteUser
- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

Utilizzare un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2

Le applicazioni eseguite su un'istanza Amazon EC2 devono includere AWS le credenziali nelle richieste API. AWS Puoi chiedere ai tuoi sviluppatori di archiviare AWS le credenziali direttamente all'interno dell'istanza Amazon EC2 e consentire alle applicazioni in quell'istanza di utilizzare tali credenziali. Tuttavia, in questo caso, gli sviluppatori dovrebbero gestire le credenziali, accertarsi che vengano passate in modo sicuro a ciascuna istanza e aggiornare ogni istanza Amazon EC2 al momento di aggiornare le credenziali. Si tratta di una notevole quantità di lavoro aggiuntivo.

In alternativa, puoi (e devi) utilizzare un ruolo IAM per gestire credenziali temporanee per le applicazioni eseguite in un'istanza Amazon EC2. Quando utilizzi un ruolo, non occorre distribuire credenziali a lungo termine (come, ad esempio, credenziali di accesso oppure chiavi di accesso) per un'istanza Amazon EC2. Il ruolo fornisce invece autorizzazioni temporanee che le applicazioni possono utilizzare quando effettuano chiamate ad altre risorse. AWS Quando avvii un'istanza Amazon EC2, devi specificare un ruolo IAM da associare ad essa. Le applicazioni eseguite nell'istanza possono quindi utilizzare le credenziali provvisorie fornite dal ruolo per firmare le richieste API.

L'utilizzo dei ruoli per concedere autorizzazioni alle applicazioni eseguite nelle istanze Amazon EC2 richiede una configurazione leggermente più elaborata Un'applicazione in esecuzione su un'istanza Amazon EC2 viene astratta AWS dal sistema operativo virtualizzato. A causa di questa ulteriore separazione, è necessario un passaggio aggiuntivo per assegnare un AWS ruolo e le autorizzazioni associate a un'istanza Amazon EC2 e renderle disponibili per le sue applicazioni. Tale passaggio aggiuntivo prevede la creazione di un profilo dell'istanza collegato all'istanza. Il profilo dell'istanza contiene il ruolo e può fornire le credenziali provvisorie del ruolo a un'applicazione eseguita nell'istanza. Le credenziali provvisorie possono essere utilizzate nelle chiamate dell'API dell'applicazione per accedere alle risorse e limitare l'accesso alle sole risorse specificate dal ruolo.

> **Note**
>
> A un'istanza Amazon EC2 può essere assegnato soltanto un ruolo alla volta e tutte le applicazioni dell'istanza condividono lo stesso ruolo e le stesse autorizzazioni. Quando si utilizza Amazon ECS per gestire le istanze Amazon EC2, alle attività di Amazon ECS è possibile assegnare dei ruoli che possono essere distinti dal ruolo dell'istanza Amazon EC2 su cui è in esecuzione. L'assegnazione di un ruolo a ciascuna attività è conforme al principio dell'accesso con privilegi minimi e consente un controllo più granulare su operazioni e risorse. Per ulteriori informazioni, consulta la pagina Utilizzo dei ruoli IAM con le attività Amazon ECS nella Guida alle best practice per Amazon Elastic Container Service.

Questo tipo di utilizzo dei ruoli offre diversi vantaggi. Dato che le credenziali dei ruoli sono temporanee e vengono aggiornate automaticamente, non dovrai preoccuparti della gestione né dei rischi di sicurezza a lungo termine. Inoltre, se utilizzi un singolo ruolo per più istanze, quando apporti una modifica a un ruolo, queste si propaga automaticamente a tutte le istanze.

> **Note**
>
> Anche se in genere un ruolo viene assegnato a un'istanza Amazon EC2 all'avvio, puoi comunque effettuare il collegamento anche a un'istanza Amazon EC2 già in esecuzione. Per informazioni sul collegamento di un ruolo a un'istanza in esecuzione, consulta Ruoli IAM per Amazon EC2.

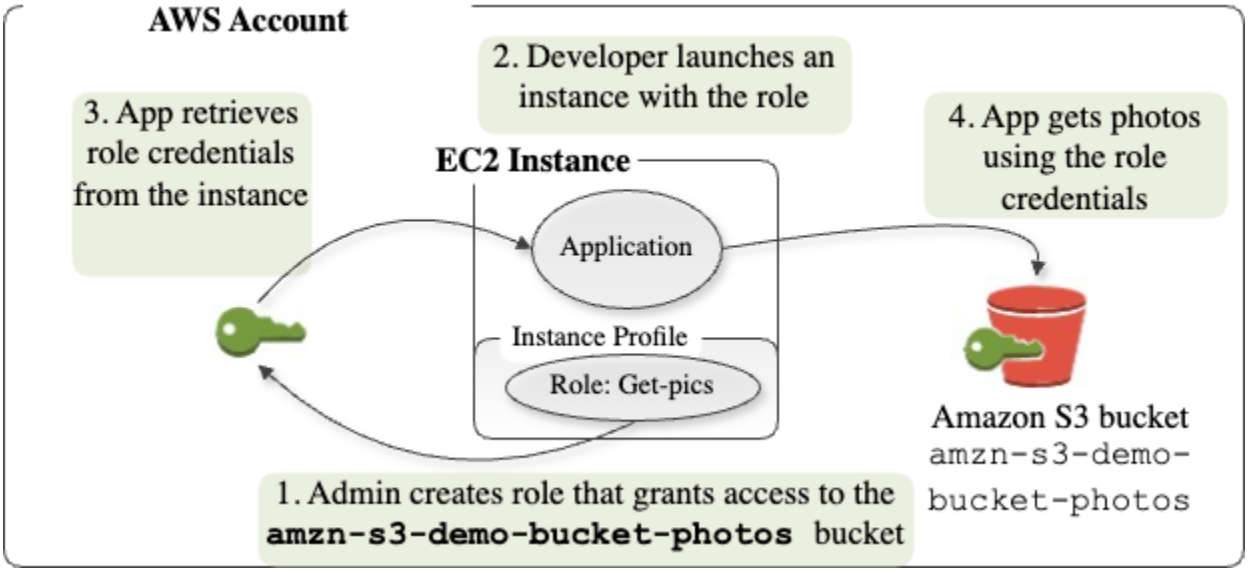
Argomenti

- Funzionamento dei ruoli per le istanze Amazon EC2
- Autorizzazioni richieste per l'utilizzo dei ruoli con Amazon EC2
- Come si inizia?
- Informazioni correlate

Funzionamento dei ruoli per le istanze Amazon EC2

Nella figura di seguito, uno sviluppatore esegue un'applicazione su un'istanza Amazon EC2 che richiede l'accesso al bucket S3 denominato `amzn-s3-demo-bucket-photos`. Un amministratore crea il ruolo di servizio `Get-pics` e lo collega all'istanza Amazon EC2. Il ruolo include una policy di autorizzazione che consente l'accesso in sola lettura al bucket S3 specificato. Include anche

una policy di affidabilità che consente all'istanza Amazon EC2 di assumere il ruolo e recuperare le credenziali provvisorie. Quando l'applicazione viene eseguita sull'istanza, può utilizzare le credenziali provvisorie del ruolo per accedere al bucket delle foto. L'amministratore non ha bisogno di concedere allo sviluppatore l'autorizzazione di accedere al bucket delle foto e lo sviluppatore non si trova mai nella necessità di condividere o gestire credenziali.



1. L'amministratore utilizza IAM per creare il ruolo **`Get-pics`**. Nella policy di affidabilità del ruolo l'amministratore specifica che solo le istanze Amazon EC2 possono assumere quel ruolo. Nella policy di autorizzazione del ruolo l'amministratore specifica autorizzazioni di sola lettura per il bucket `amzn-s3-demo-bucket-photos`.
2. Uno sviluppatore avvia un'istanza Amazon EC2 e assegna il ruolo `Get-pics` all'istanza.

> **Note**
>
> Se utilizzi la console IAM, il profilo dell'istanza viene gestito in automatico, con un processo quasi completamente trasparente. Tuttavia, se utilizzi l'API AWS CLI or per creare e gestire il ruolo e l'istanza Amazon EC2, devi creare il profilo dell'istanza e assegnargli il ruolo in passaggi separati. Quindi, quando avvii l'istanza dovrai specificare il nome del profilo dell'istanza anziché il nome del ruolo.

3. Quando l'applicazione è in esecuzione, raccoglie le credenziali di sicurezza provvisorie dai metadati dell'istanza Amazon EC2, come descritto in Recupero delle credenziali di sicurezza dai metadati delle istanze. Si tratta di credenziali di sicurezza provvisorie che rappresentano il ruolo e hanno un periodo di validità limitato.

Con alcuni AWS SDK, lo sviluppatore può utilizzare un provider per la gestione trasparente delle credenziali di sicurezza provvisorie. (La documentazione per i singoli AWS SDK descrive le funzionalità supportate da tale SDK per la gestione delle credenziali.)

In alternativa, l'applicazione può ottenere le credenziali provvisorie direttamente dai metadati dell'istanza Amazon EC2. Le credenziali e i valori correlati sono disponibili nella categoria `iam/security-credentials/`*`role-name`* (in questo caso `iam/security-credentials/Get-pics`) dei metadati. Se l'applicazione ottiene le credenziali dai metadati dell'istanza, può memorizzarle nella cache.

4. Grazie all'utilizzo delle credenziali provvisorie recuperate, l'applicazione può accedere al bucket delle foto. In virtù della policy collegata al ruolo **`Get-pics`** (Ottieni foto), l'applicazione dispone di autorizzazioni di sola lettura.

   Le credenziali di sicurezza temporanee disponibili nell'istanza vengono aggiornate automaticamente prima della scadenza, in modo da avere un set valido sempre disponibile. L'applicazione deve solo assicurarsi di ottenere un nuovo set di credenziali dai metadati dell'istanza prima della scadenza di quelle esistenti. È possibile utilizzare l' AWS SDK per gestire le credenziali in modo che l'applicazione non debba includere logica aggiuntiva per aggiornare le credenziali. Ad esempio, creando istanze di client con provider di credenziali del profilo dell'istanza. Tuttavia, se l'applicazione ottiene le credenziali di sicurezza provvisorie dai metadati dell'istanza e le memorizza nella cache, è necessario fornire un set di credenziali aggiornato ogni ora o almeno 15 minuti prima della scadenza del set corrente. L'ora di scadenza è indicata nelle informazioni restituite nella categoria `iam/security-credentials/`*`role-name`*.

Autorizzazioni richieste per l'utilizzo dei ruoli con Amazon EC2

Per avviare un'istanza con un ruolo, lo sviluppatore deve avere l'autorizzazione per avviare le istanze Amazon EC2 e per passare i ruoli IAM.

La seguente policy di esempio consente agli utenti di utilizzare il per Console di gestione AWS avviare un'istanza con un ruolo. La policy include caratteri jolly (`*`) per consentire a un utente di passare qualsiasi ruolo ed eseguire tutte le operazioni di Amazon EC2 elencate. L'operazione `ListInstanceProfiles` consente agli utenti di visualizzare tutti i ruoli disponibili nell' Account AWS.

Example Esempio di policy che concede a un utente l'autorizzazione di utilizzare la console Amazon EC2 per avviare un'istanza con qualsiasi ruolo

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "IamPassRole",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
            }
        },
        {
            "Sid": "ListEc2AndListInstanceProfiles",
            "Effect": "Allow",
            "Action": [
                "iam:ListInstanceProfiles",
                "ec2:Describe*",
                "ec2:Search*",
                "ec2:Get*"
            ],
            "Resource": "*"
        }
    ]
}
```

Limitare i ruoli che possono essere passati alle istanze Amazon EC2 (utilizzando) PassRole

È possibile utilizzare l'autorizzazione `PassRole` per limitare i ruoli che un utente può passare a un'istanza Amazon EC2 quando avvia l'istanza. In questo modo si impedisce all'utente di eseguire le applicazioni che dispongono di più autorizzazioni rispetto a quelle concesse all'utente (ovvero di ottenere privilegi elevati). Ad esempio, immaginiamo che l'utente Alice disponga solo delle autorizzazioni per avviare le istanze Amazon EC2 e per operare con i bucket di Amazon S3, ma

che passi a un'istanza Amazon EC2 un ruolo con autorizzazioni per operare con IAM e Amazon DynamoDB. In questo caso, Alice potrebbe essere in grado di avviare l'istanza, accedere a essa, ottenere credenziali di sicurezza temporanee e quindi eseguire operazioni IAM o DynamoDB per cui non dispone dell'autorizzazione.

Per limitare i ruoli che un utente può passare a un'istanza Amazon EC2, devi creare una policy che consenta l'operazione `PassRole`. A quel punto, puoi collegare la policy all'utente (o a un gruppo IAM a cui l'utente appartiene) che avvierà le istanze Amazon EC2. Nell'elemento `Resource` della policy devi elencare il ruolo o i ruoli che l'utente può passare alle istanze Amazon EC2. Quando l'utente avvia un'istanza e la associa a un ruolo, Amazon EC2 verifica se l'utente è autorizzato a inviare quel ruolo. Ovviamente, devi anche accertarti che il ruolo passato dall'utente non includa un numero di autorizzazioni maggiore di quello consentito all'utente.

> **Note**
>
> `PassRole` non è un'operazione API pari a `RunInstances` o `ListInstanceProfiles`. Si tratta invece di un'autorizzazione che AWS verifica ogni volta che un ruolo ARN viene passato come parametro a un'API (o la console lo fa per conto dell'utente). In questo modo, un amministratore ha la possibilità di controllare quali ruoli possono essere passati dai vari utenti. In questo caso, garantisce che l'utente abbia l'autorizzazione per collegare un ruolo specifico a un'istanza Amazon EC2.

Example policy che concede a un utente l'autorizzazione di avviare un'istanza Amazon EC2 con un ruolo specifico

Il seguente esempio di policy consente agli utenti di utilizzare l'API Amazon EC2 per avviare un'istanza con un ruolo. L'elemento `Resource` specifica l'Amazon Resource Name (ARN) di un ruolo. Specificando l'ARN, la policy concede all'utente l'autorizzazione di passare solo il ruolo `Get-pics`. Se, all'avvio di un'istanza, l'utente cerca di specificare un ruolo diverso, l'operazione ha esito negativo. L'utente non è autorizzato a eseguire alcuna istanza, indipendentemente dal passaggio di un ruolo.

JSON

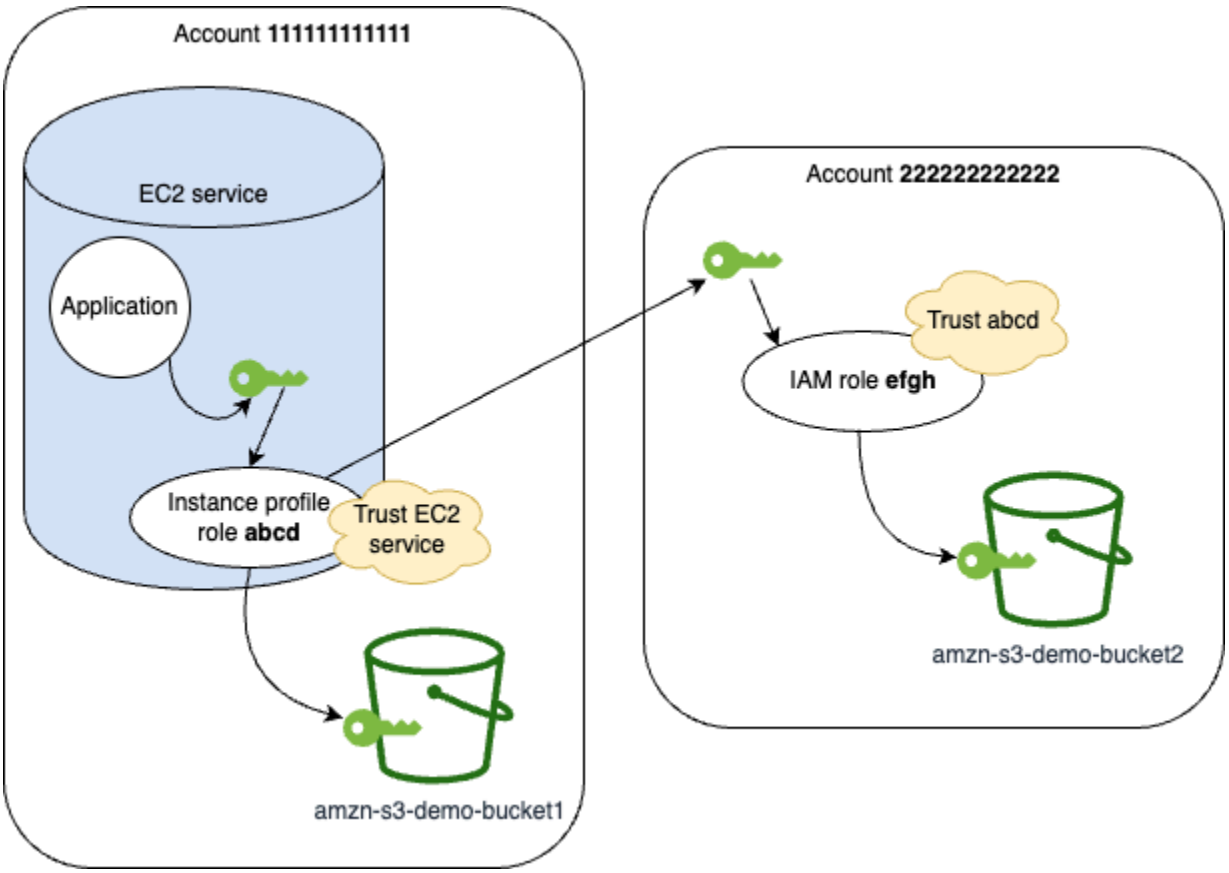
```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/Get-pics"
        }
    ]
}
```

Consentire a un ruolo del profilo dell'istanza di passare a un ruolo in un altro account

Puoi consentire a un'applicazione in esecuzione su un'istanza Amazon EC2 di eseguire comandi in un altro account. A tale scopo, devi consentire al ruolo dell'istanza Amazon EC2 nel primo account di passare a un ruolo nel secondo account.

Immagina di utilizzarne due Account AWS e di voler consentire a un'applicazione in esecuzione su un'istanza Amazon EC2 di eseguire AWS CLI comandi in entrambi gli account. Supponiamo che l'istanza Amazon EC2 esista nell'account `111111111111`. Tale istanza include il ruolo del profilo dell'istanza `abcd` che consente all'applicazione di eseguire attività Amazon S3 di sola lettura nel bucket `amzn-s3-demo-bucket1` all'interno dello stesso account `111111111111`. Tuttavia, l'applicazione deve anche poter assumere il ruolo tra account `efgh` per accedere al bucket `amzn-s3-demo-bucket2` di Amazon S3 nell'account `222222222222`.



Il ruolo del profilo dell'istanza Amazon EC2 `abcd` deve disporre della policy di autorizzazioni seguente per consentire all'applicazione di accedere al bucket `amzn-s3-demo-bucket1` di Amazon S3:

Policy di autorizzazioni del ruolo ***abcd*** 111111111111 dell'account

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccountLevelS3Actions",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
```

```
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowListAndReadS3ActionOnMyBucket",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1/*",
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
        },
        {
            "Sid": "AllowIPToAssumeCrossAccountRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::222222222222:role/efgh"
        }
    ]
}
```

Il ruolo `abcd` deve considerare il servizio Amazon EC2 come attendibile ad assumere il ruolo. A tale scopo, il ruolo `abcd` deve disporre della seguente policy di attendibilità:

Policy di attendibilità del ruolo ***abcd*** dell'account 111111111111

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "abcdTrustPolicy",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"Service": "ec2.amazonaws.com"}
        }
    ]
}
```

Supponiamo che il ruolo tra account `efgh` consenta attività Amazon S3 di sola lettura nel bucket `amzn-s3-demo-bucket2` all'interno dello stesso account `222222222222`. A tale scopo, il ruolo tra account `efgh` deve disporre della seguente policy di autorizzazioni:

Policy di autorizzazioni del ruolo ***efgh*** 222222222222 dell'account

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccountLevelS3Actions",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowListAndReadS3ActionOnMyBucket",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket2/*",
                "arn:aws:s3:::amzn-s3-demo-bucket2"
            ]
        }
    ]
}
```

Il ruolo `efgh` deve consentire al ruolo del profilo dell'istanza `abcd` di assumerlo. A tale scopo, il ruolo `efgh` deve disporre della seguente policy di attendibilità:

Policy di attendibilità del ruolo ***efgh*** dell'account 222222222222

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "efghTrustPolicy",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"AWS": "arn:aws:iam::111111111111:role/abcd"}
        }
    ]
}
```

Come si inizia?

Per comprendere come funzionano i ruoli nelle istanze Amazon EC2, crea un ruolo con la console IAM, avvia un'istanza Amazon EC2 che usi tale ruolo e quindi osserva l'istanza in esecuzione. Puoi prendere in esame i metadati dell'istanza per consultare in che modo le credenziali provvisorie del ruolo vengano rese disponibili a un'istanza. Potrai anche consultare il modo in cui un'applicazione eseguita in un'istanza può utilizzare tale ruolo. Per ottenere ulteriori informazioni, usare le risorse indicate di seguito.

-
- Procedure guidate sugli SDK. La documentazione AWS SDK include procedure dettagliate che mostrano un'applicazione in esecuzione su un'istanza Amazon EC2 che utilizza credenziali temporanee per i ruoli per leggere un bucket Amazon S3. Ogni procedura guidata presenta passaggi simili, ma utilizza un linguaggio di programmazione diverso:
  - Configurazione dei ruoli IAM per Amazon EC2 con SDK per Java nella Guida per gli sviluppatori di AWS SDK per Java
  - Avvio di un'istanza Amazon EC2 utilizzando SDK per .NET nella Guida per gli sviluppatori di AWS SDK per .NET
  - Creazione di una istanza Amazon EC2 con SDK for Ruby nella Guida per gli sviluppatori di AWS SDK per Ruby

Informazioni correlate

Per ulteriori informazioni sulla creazione di ruoli o di ruoli per le istanze Amazon EC2, consulta la seguente documentazione:

- Per ulteriori informazioni sull'utilizzo dei ruoli IAM con istanze Amazon EC2, consulta la Guida per l'utente di Amazon EC2.
- Per creare un ruolo, consulta Creazione di ruoli IAM
- Per ulteriori informazioni sull'utilizzo delle credenziali di sicurezza provvisorie, vedi Credenziali di sicurezza temporanee in IAM.
- Se lavori con l'API IAM o la CLI, devi creare e gestire i profili delle istanze IAM. Per ulteriori informazioni sui profili delle istanze, consulta Usare profili dell'istanza.
- Per ulteriori informazioni sulle credenziali di sicurezza provvisorie per i ruoli dei metadati dell'istanza, consulta Recupero delle credenziali di sicurezza dai metadati delle istanze nella Guida per l'utente di Amazon EC2.

Usare profili dell'istanza

Utilizza un profilo dell'istanza per passare un ruolo IAM a un'istanza EC2. Per ulteriori informazioni, consulta Ruoli IAM per Amazon EC2 nella Guida per l'utente di Amazon EC2.

Gestione dei profili delle istanze (console)

Se si utilizza per Console di gestione AWS creare un ruolo per Amazon EC2, la console crea automaticamente un profilo di istanza e gli assegna lo stesso nome del ruolo. Successivamente, quando utilizzi la console Amazon EC2 per avviare un'istanza con un ruolo IAM, potrai selezionare un ruolo da associare all'istanza. L'elenco visualizzato nella console è in effetti elenco di nomi di profili delle istanze. La console non crea un profilo dell'istanza per un ruolo non associato ad Amazon EC2.

Puoi utilizzare Console di gestione AWS per eliminare i ruoli IAM e i profili di istanza per Amazon EC2 se il ruolo e il profilo dell'istanza hanno lo stesso nome. Per ulteriori informazioni sull'eliminazione dei profili di istanza, consulta Eliminare i ruoli o i profili delle istanze.

> **Note**
>
> Per aggiornare le autorizzazioni di un'istanza, sostituisci il suo profilo dell'istanza. Sconsigliamo di rimuovere un ruolo da un profilo dell'istanza, poiché è necessario attendere fino a un'ora prima che la modifica abbia effetto.

Gestione dei profili di istanza (AWS CLI or AWS API)

Se gestisci i tuoi ruoli da AWS CLI o dall' AWS API, crei ruoli e profili di istanza come azioni separate. Poiché ruoli e profili delle istanze possono avere nomi diversi, è importante che tu conosca i nomi dei profili e dei ruoli che contengono. In questo modo,sarà più semplice selezionare il corretto profilo all'avvio di un'istanza EC2.

È possibile collegare tag alle risorse IAM, inclusi i profili dell'istanza, per identificare, organizzare e controllare l'accesso a tali risorse. Puoi etichettare i profili di istanza solo quando usi l' AWS API AWS CLI or.

> **Note**
>
> Un profilo dell'istanza può contenere un solo ruolo IAM, mentre lo stesso ruolo può essere in più profili delle istanze. Non è possibile aumentare il numero di ruoli per profilo dell'istanza. Tuttavia, puoi rimuovere il ruolo esistente nel profilo dell'istanza e aggiungerne uno diverso. È quindi necessario attendere che la modifica venga visualizzata su tutti i dispositivi per AWS motivi di https://en.wikipedia.org/wiki/Eventual_consistency coerenza. Per forzare la modifica, devi dissociare il profilo dell'istanza e quindi associare il profilo dell'istanza oppure arrestare l'istanza e riavviarla.

Gestione dei profili delle istanze (AWS CLI)

È possibile utilizzare i seguenti AWS CLI comandi per lavorare con i profili di istanza in un AWS account.

- Per creare un profilo dell'istanza: `aws iam create-instance-profile`
- Applicare tag a un profilo dell'istanza: `aws iam tag-instance-profile`
- Elencare i tag per un profilo dell'istanza: `aws iam list-instance-profile-tags`
- Rimuovere i tag da un profilo dell'istanza: `aws iam untag-instance-profile`
- Per aggiungere un ruolo a un profilo dell'istanza: `aws iam add-role-to-instance-profile`
- Per elencare i profili delle istanze: `aws iam list-instance-profiles`, `aws iam list-instance-profiles-for-role`
- Per ottenere informazioni su un profilo dell'istanza: `aws iam get-instance-profile`
- Per rimuovere un ruolo da un profilo dell'istanza: `aws iam remove-role-from-instance-profile`

- Per eliminare un profilo dell'istanza: `aws iam delete-instance-profile`

È anche possibile collegare un ruolo a un'istanza EC2 già in esecuzione, utilizzando i comandi riportati di seguito. Per ulteriori informazioni, consulta Ruoli IAM per Amazon EC2.

- Collegare un profilo dell'istanza con ruolo a un'istanza EC2 in funzione o arrestata: `aws ec2 associate-iam-instance-profile`
- Ottenere informazioni su un profilo dell'istanza collegato a un'istanza EC2: `aws ec2 describe-iam-instance-profile-associations`
- Distaccare un profilo dell'istanza con ruolo da un'istanza EC2 in funzione o arrestata: `aws ec2 disassociate-iam-instance-profile`

Gestione dei profili di istanza (AWS API)

Puoi chiamare le seguenti operazioni AWS API per lavorare con i profili di istanza in un Account AWS.

- Per creare un profilo dell'istanza: `CreateInstanceProfile`
- Applicare tag a un profilo dell'istanza: `TagInstanceProfile`
- Elencare i tag su un profilo dell'istanza: `ListInstanceProfileTags`
- Rimuovere i tag da un profilo dell'istanza: `UntagInstanceProfile`
- Per aggiungere un ruolo a un profilo dell'istanza: `AddRoleToInstanceProfile`
- Per elencare i profili delle istanze: `ListInstanceProfiles`, `ListInstanceProfilesForRole`
- Per ottenere informazioni su un profilo dell'istanza: `GetInstanceProfile`
- Per rimuovere un ruolo da un profilo dell'istanza: `RemoveRoleFromInstanceProfile`
- Per eliminare un profilo dell'istanza: `DeleteInstanceProfile`

È anche possibile collegare un ruolo a un'istanza EC2 già in esecuzione, richiamando le operazioni riportate di seguito. Per ulteriori informazioni, consulta Ruoli IAM per Amazon EC2.

- Collegare un profilo dell'istanza con ruolo a un'istanza EC2 in funzione o arrestata: `AssociateIamInstanceProfile`
- Ottenere informazioni su un profilo dell'istanza collegato a un'istanza EC2: `DescribeIamInstanceProfileAssociations`

- Distaccare un profilo dell'istanza con ruolo da un'istanza EC2 in funzione o arrestata: `DisassociateIamInstanceProfile`

Fornitori di identità e federazione in AWS

Come best practice, ti consigliamo di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere alle AWS risorse invece di creare singoli utenti IAM nel tuo Account AWS. Con un provider di identità (IdP), puoi gestire le tue identità utente all'esterno AWS e concedere a queste identità utente esterne le autorizzazioni per utilizzare le AWS risorse del tuo account. Questa funzione è utile se la tua organizzazione dispone già di un proprio sistema di gestione delle identità, come ad esempio una directory aziendale degli utenti. È utile anche se stai creando un'app mobile o un'applicazione web che richiede l'accesso alle risorse. AWS

> **Note**
>
> Consigliamo inoltre di gestire gli utenti nel Centro identità IAM con un provider di identità SAML esterno anziché utilizzare la federazione SAML in IAM. La federazione di IAM Identity Center con un provider di identità consente di consentire alle persone di accedere a più AWS account dell'organizzazione e a più AWS applicazioni. Per informazioni su situazioni specifiche in cui è richiesto un utente IAM, consulta la sezione Quando creare un utente IAM invece di un ruolo.
> Puoi anche utilizzare l'account access manager , una funzionalità IAM che ti consente di assegnare ruoli IAM esistenti negli account della tua organizzazione agli utenti e ai gruppi di IAM Identity Center. Puoi utilizzarla insieme ai set di autorizzazioni o da sola.

Se preferisci utilizzare un singolo AWS account senza abilitare IAM Identity Center, puoi utilizzare IAM con un IdP esterno che fornisce informazioni sull'identità AWS utilizzando OpenID Connect (OIDC) o SAML 2.0 (Security Assertion Markup Language 2.0). L'OIDC collega le applicazioni, come le azioni, che non vengono GitHub eseguite sulle risorse. AWS AWS Tra i gestori dell'identità digitale SAML noti figurano Shibboleth e Active Directory Federation Services.

Quando utilizzi un provider di identità, non devi creare un codice di accesso personalizzato né gestire le tue identità utente. Tale operazione viene eseguita dall'IdP. Gli utenti esterni accedono tramite un IdP e tu puoi concedere a tali identità esterne le autorizzazioni per utilizzare AWS le risorse del tuo account. I provider di identità ti aiutano a Account AWS proteggerti perché non devi distribuire o incorporare credenziali di sicurezza a lungo termine, come le chiavi di accesso, nell'applicazione.

Consulta la tabella seguente per determinare quale tipo di federazione IAM è il migliore per il tuo caso d'uso: IAM, Centro identità IAM o Amazon Cognito. I riepiloghi e la tabella seguenti forniscono una panoramica dei metodi che gli utenti possono utilizzare per ottenere l'accesso federato alle risorse. AWS

Tipo di federazione IAM	Tipo di account	Gestione degli accessi di...	Origine di identità supportata
Federazione con il Centro identità IAM	Account multipli gestiti da AWS Organizations	Utenti umani della forza lavoro	• SAML 2.0 • Active Directory gestita • Directory del Centro identità
Federazione con IAM	Singolo account autonomo	• Utenti umani impegnati in implementazioni a breve termine su piccola scala • Utenti di macchine	• SAML 2.0 • OIDC
Federazione con pool di identità Amazon Cognito	Qualsiasi	Utenti di app che richiedono l'autorizzazione IAM per accedere alle risorse	• SAML 2.0 • OIDC • Selezionare un gestore di identità digitali social OAuth 2.0

Federazione con il Centro identità IAM

Per una gestione centralizzata degli accessi degli utenti umani, consigliamo di utilizzare Centro identità IAM per gestire l'accesso ai tuoi account e le autorizzazioni all'interno di questi account. Agli utenti di IAM Identity Center vengono concesse credenziali a breve termine per AWS le tue risorse. Puoi utilizzare Active Directory, un provider di identità esterno (IdP) o una directory IAM Identity Center come fonte di identità per utenti e gruppi per assegnare l'accesso alle tue AWS risorse.

IAM Identity Center supporta la federazione delle identità con SAML (Security Assertion Markup Language) 2.0 per fornire un accesso Single Sign-on federato agli utenti autorizzati a utilizzare le applicazioni all'interno del portale di accesso. AWS Gli utenti possono quindi accedere in modalità Single Sign-On ai servizi che supportano SAML, incluse le applicazioni di terze parti, come Microsoft 365, SAP Concur Console di gestione AWS e Salesforce.

Federazione con IAM

Anche se consigliamo vivamente di gestire gli utenti umani nel Centro identità IAM, è possibile abilitare l'accesso ai principali federati con IAM per gli utenti umani impegnati in implementazioni a breve termine su piccola scala. IAM consente di utilizzare SAML 2.0 e Open ID Connect (OIDC) separati e di utilizzare attributi principali federati per il controllo degli accessi. IdPs Con IAM, puoi trasferire gli attributi utente, come centro di costo, titolo o lingua, dal tuo IdPs a e implementare autorizzazioni di accesso AWS granulari basate su questi attributi.

Un carico di lavoro è una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione o un processo di backend. Il tuo carico di lavoro può richiedere un'identità IAM per effettuare richieste a AWS servizi, applicazioni, strumenti operativi e componenti. Queste identità includono macchine in esecuzione nei tuoi AWS ambienti, come le istanze o le funzioni di Amazon EC2. AWS Lambda

Puoi anche gestire identità computer per soggetti esterni che necessitano di accesso. Per concedere l'accesso alle identità computer, puoi utilizzare i ruoli IAM. I ruoli IAM dispongono di autorizzazioni specifiche e forniscono un modo per accedere AWS facendo affidamento su credenziali di sicurezza temporanee con una sessione di ruolo. Inoltre, potresti avere macchine esterne AWS che necessitano di accedere ai tuoi ambienti. AWS Per le macchine che funzionano all'esterno dell' AWS utente, puoi usare IAM Roles Anywhere. Per ulteriori informazioni sui ruoli, consulta Ruoli IAM. Per dettagli su come utilizzare i ruoli per delegare l'accesso Account AWS, consultaTutorial IAM: Delega l'accesso attraverso AWS account che utilizzano ruoli IAM.

Per collegare un IdP direttamente a IAM, crei un'entità provider di identità per stabilire una relazione di fiducia tra il tuo Account AWS e l'IdP. Supporti IAM IdPs compatibili con OpenID Connect (OIDC) o SAML 2.0 (Security Assertion Markup Language 2.0). Per ulteriori informazioni sull'utilizzo di uno di questi IdPs con AWS, consulta le seguenti sezioni:

- Federazione OIDC
- Federazione SAML 2.0

Federazione con pool di identità Amazon Cognito

Amazon Cognito è progettato per gli sviluppatori che desiderano autenticare e autorizzare gli utenti nelle proprie app mobili e Web. I pool di utenti di Amazon Cognito aggiungono funzionalità di accesso e registrazione all'app e i pool di identità forniscono credenziali IAM che consentono agli utenti di accedere alle risorse protette gestite in AWS. I pool di identità acquisiscono le credenziali per le sessioni temporanee tramite il funzionamento dell'operazione API `AssumeRoleWithWebIdentity`.

Amazon Cognito funziona con gestori dell'identità digitale esterni che supportano SAML e OpenID Connect e con gestori di identità social come Facebook, Google e Amazon. L'app può far effettuare l'accesso a un utente con un pool di utenti o un gestore dell'identità digitale esterno e recuperare in seguito le risorse per suo conto con sessioni temporanee personalizzate con un ruolo IAM.

Risorse aggiuntive

- Per una dimostrazione su come creare un proxy federativo personalizzato che abiliti il single sign-on (SSO) nell' Console di gestione AWS utilizzo del sistema di autenticazione dell'organizzazione, vedere. Abilita l'accesso personalizzato del broker di identità al AWS console
- Per istruzioni su come consentire ai AWS carichi di lavoro di accedere a servizi esterni senza memorizzare le credenziali, consulta. Federando AWS Identità verso servizi esterni

Scenari comuni

> **Note**
>
> Ti consigliamo di richiedere agli utenti umani di utilizzare credenziali temporanee per l'accesso AWS. Hai preso in considerazione l'utilizzo Centro identità AWS IAM? Puoi utilizzare IAM Identity Center per gestire centralmente l'accesso a più account Account AWS e fornire agli MFA-protected utenti l'accesso Single Sign-on a tutti gli account assegnati da un'unica posizione. Con IAM Identity Center puoi creare e gestire le identità degli utenti in IAM Identity Center o connetterti facilmente al tuo attuale gestore dell'identità digitale (IdP) compatibile con SAML 2.0. Per ulteriori informazioni, consulta Cos'è IAM Identity Center? nella Guida per l'utente di Centro identità AWS IAM .

Puoi utilizzare un gestore dell'identità digitale (IdP) esterno per gestire le identità degli utenti al di fuori di AWS e dell'IdP esterno. Un IdP esterno può fornire informazioni sull'identità AWS

utilizzando OpenID Connect (OIDC) o Security Assertion Markup Language (SAML). L'OIDC viene comunemente utilizzato quando un'applicazione che non viene eseguita richiede l'accesso alle risorse. AWS AWS

Quando si desidera configurare la federazione con un IdP esterno, è necessario creare un provider di identità IAM per fornire AWS informazioni sull'IdP esterno e sulla sua configurazione. Questo stabilisce un rapporto di fiducia tra il tuo IdP Account AWS e quello esterno. I seguenti argomenti forniscono scenari comuni per l'utilizzo dei provider di identità IAM.

Argomenti

- Amazon Cognito per applicazioni per dispositivi mobili
- Federazione OIDC per applicazioni per dispositivi mobili

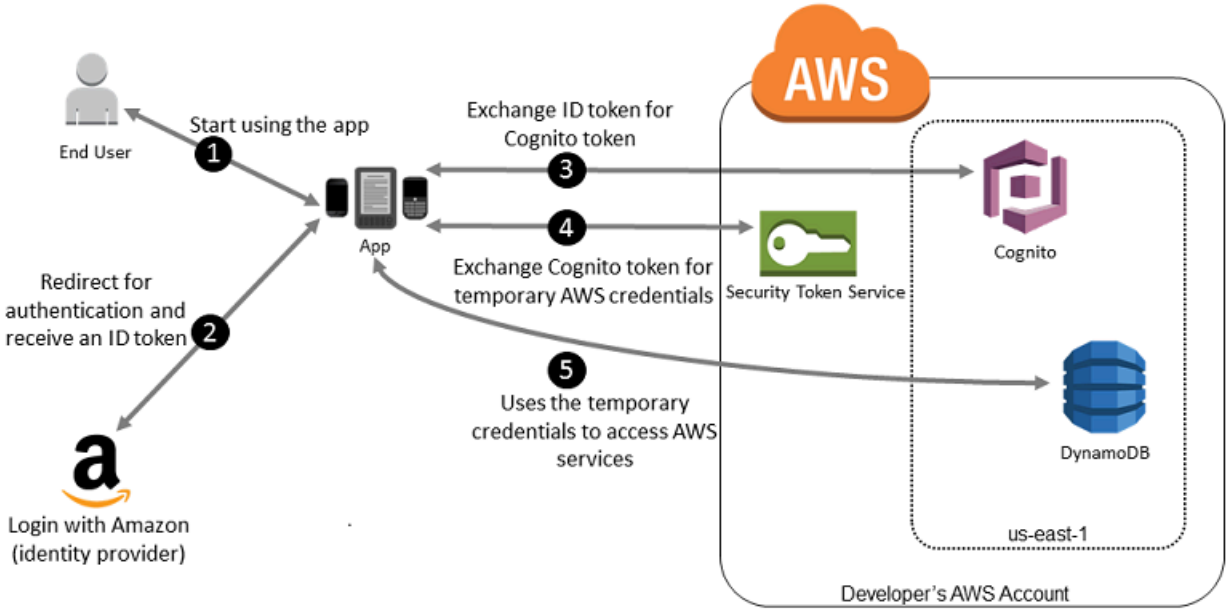
Amazon Cognito per applicazioni per dispositivi mobili

Il modo migliore per utilizzare la federazione OIDC è utilizzare Amazon Cognito. Ad esempio, Adele sta sviluppando un gioco per un dispositivo mobile in cui i dati dell'utente, quali punteggi e profili, vengono memorizzati in Amazon S3 e Amazon DynamoDB. Adele potrebbe archiviare questi dati anche in locale sul dispositivo e utilizzare Amazon Cognito per mantenerli sincronizzati su tutti i dispositivi. Tuttavia, Adele è consapevole che, per motivi di sicurezza e manutenzione, le credenziali di sicurezza AWS a lungo termine non dovrebbero essere distribuite con il gioco. Adele sa anche che il gioco potrebbe avere un gran numero di utenti. Per tutti questi motivi, non desidera creare nuove identità utente in IAM per ciascun giocatore. In alternativa, decide di sviluppare il gioco in modo che gli utenti possano effettuare l'accesso utilizzando un'identità già stabilita con un noto gestore dell'identità digitale (IdP) esterno, ad esempio Accedi con Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC). Il suo gioco può sfruttare il meccanismo di autenticazione di uno di questi provider per convalidare l'identità dell'utente.

Per consentire all'app mobile di accedere alle sue AWS risorse, Adele si registra innanzitutto per ottenere un ID sviluppatore a sua scelta. IdPs Configura anche l'applicazione con ciascuno di questi provider. Nella versione Account AWS che contiene il bucket Amazon S3 e la tabella DynamoDB per il gioco, Adele utilizza Amazon Cognito per creare ruoli IAM che definiscono con precisione le autorizzazioni necessarie al gioco. Se utilizza un IdP OIDC, crea anche un'entità di provider di identità IAM OIDC per stabilire la fiducia tra un pool di identità Amazon Cognito presente in lei e l'IdP. https://docs.aws.amazon.com/cognito/latest/developerguide/external-identity-providers.html Account AWS

Nel codice dell'app, Adele chiama l'interfaccia di accesso per il provider di identità che ha configurato in precedenza. Il provider di identità gestisce tutti i dettagli relativi all'accesso dell'utente e l'app riceve un token di accesso OAuth o un token ID OIDC dal provider. L'app di Adele può scambiare queste informazioni di autenticazione con una serie di credenziali di sicurezza temporanee costituite da un ID chiave di accesso, una chiave di AWS accesso segreta e un token di sessione. L'app può quindi utilizzare queste credenziali per accedere ai servizi web offerti da. AWS L'app è limitata alle autorizzazioni definite nel ruolo che assume.

La figura riportata di seguito mostra un flusso semplificato su come questo processo potrebbe funzionare, usando Login with Amazon come provider di identità. Per il passaggio 2, l'app può anche utilizzare Facebook, Google o qualsiasi OIDC-compatible IdP, ma non è mostrato qui.



1. Un cliente avvia l'app su un dispositivo mobile. L'app richiede all'utente di effettuare l'accesso.
2. L'app utilizza il login con le risorse di Login with Amazon per accettare le credenziali dell'utente.
3. L'app utilizza le operazioni dell'API Amazon Cognito `GetId` e `GetCredentialsForIdentity` per scambiare il token ID di Login with Amazon con un token di Amazon Cognito. Amazon Cognito, che è stato configurato per rendere attendibile il tuo progetto Login with Amazon, genera un token che scambia con credenziali di sessione temporanee per AWS STS.
4. L'app riceve le credenziali di sicurezza temporanee da Amazon Cognito. La tua app può anche utilizzare il flusso di lavoro Basic (Classic) in Amazon Cognito per recuperare i token dall'utilizzo. AWS STS `AssumeRoleWithWebIdentity` Per ulteriori informazioni, consulta Flusso di

autenticazione dei pool di identità (identità federate) nella Guida per gli sviluppatori di Amazon Cognito.

5. Le credenziali di sicurezza temporanee possono essere utilizzate dall'app per accedere alle risorse AWS richieste dall'applicazione per funzionare. Il ruolo associato alle credenziali di sicurezza temporanee e alle relative policy assegnate determina a quali elementi è possibile accedere.

Utilizza la seguente procedura per configurare la tua app in modo che utilizzi Amazon Cognito per autenticare gli utenti e consentire all'app di accedere alle risorse. AWS Per le operazioni specifiche per realizzare questo scenario, consulta la documentazione di Amazon Cognito.

1. (Facoltativo) Registrati come sviluppatore con Login with Amazon, Facebook, Google o qualsiasi altro provider di identità compatibile con OpenID Connect (OIDC) e configura una o più app con il provider. Questa fase è facoltativa poiché Amazon Cognito supporta anche l'accesso non autenticato (guest) per gli utenti.
2. Accedi ad Amazon Cognito in. Console di gestione AWS Utilizza la procedura guidata di Amazon Cognito per creare un pool di identità, ovvero un container che Amazon Cognito utilizza per mantenere le identità degli utenti finali organizzate per le app. Puoi condividere i pool di identità tra le app. Quando configuru un pool di identità, Amazon Cognito crea uno o due ruoli IAM (uno per le identità autenticate e uno per le identità "guest" non autenticate) che definiscono le autorizzazioni per gli utenti di Amazon Cognito.
3. Integra AWS Amplify con l'app e importa i file necessari per utilizzare Amazon Cognito.
4. Crea un'istanza del provider di credenziali di Amazon Cognito, passando l'ID del pool di identità, il numero del tuo Account AWS e l'Amazon Resource Name (ARN) dei ruoli associati al pool di identità. La procedura guidata di Amazon Cognito in Console di gestione AWS fornisce un codice di esempio per aiutarti a iniziare.
5. Quando l'app accede a una AWS risorsa, passa l'istanza del fornitore di credenziali all'oggetto client, che trasmette le credenziali di sicurezza temporanee al client. Le autorizzazioni per le credenziali si basano sul ruolo o sui ruoli definiti in precedenza.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Accedi (Android) nella documentazione del AWS Amplify Framework.
- Accedi (iOS) nella documentazione del AWS Amplify framework.

Federazione OIDC per applicazioni per dispositivi mobili

Per risultati ottimali, utilizza Amazon Cognito come gestore di identità per quasi tutti gli scenari di federazione OIDC. Amazon Cognito è semplice da utilizzare e fornisce funzionalità aggiuntive quali l'accesso anonimo (non autenticato) e la sincronizzazione dei dati degli utenti tra più dispositivi e provider. Tuttavia, se hai già creato un'app che utilizza la federazione OIDC chiamando manualmente l'API `AssumeRoleWithWebIdentity`, puoi continuare a utilizzarla e le app funzioneranno comunque correttamente.

Il processo per l'utilizzo della federazione OIDC senza Amazon Cognito segue questa struttura generale:

1. Effettuare la registrazione come sviluppatore al provider di identità (IdP) esterno e configurare l'app con tale provider, che fornisce un ID univoco per l'app. (Diversi IdPs utilizzano una terminologia diversa per questo processo. Questo schema utilizza il termine configura per il processo di identificazione dell'app con l'IdP.) Ogni IdP ti fornisce un ID app univoco per quell'IdP, quindi se configuri la stessa app con più ID di app IdPs, la tua app avrà più ID di app. È possibile configurare più app con ciascun provider.

   I seguenti link esterni forniscono informazioni sull'utilizzo di alcuni dei provider di identità più utilizzati ()IdPs:

   - Centro Sviluppatori di Login with Amazon
   - Aggiunta dell'accesso a Facebook a un'app o a un sito Web sul sito degli sviluppatori di Facebook.
   - Utilizzo di OAuth 2.0 per l'accesso (OpenID Connect) sul sito degli sviluppatori di Google.

   > **Important**
   >
   > Se utilizzi un provider di identità OIDC di Google, Facebook o Amazon Cognito, non creare un provider di identità IAM separato nel. Console di gestione AWS AWS dispone di questi provider di identità OIDC integrati e disponibili per l'uso. Ignora la fase seguente e passa direttamente alla creazione di nuovi ruoli utilizzando il provider di identità.

2. Se utilizzi un IdP di identità diverso da Google, Facebook o Amazon Cognito compatibile con OIDC, crea un'entità provider di identità IAM.
3. In IAM, crea uno o più ruoli. Per ogni ruolo, definisci chi può assumere il ruolo (policy di attendibilità) e quali autorizzazioni concedere agli utenti dell'app (policy di autorizzazione). Di solito, è necessario creare un ruolo per ogni provider di identità supportato da un'app. Ad esempio,

puoi creare un ruolo che può essere assunto da un'applicazione quando l'utente effettua l'accesso tramite Login with Amazon, un secondo ruolo per la stessa applicazione in cui l'utente effettua l'accesso tramite Facebook e un terzo ruolo per l'applicazione in cui l'utente effettua l'accesso tramite Google. Per la relazione di fiducia, specifica l'IdP (like Amazon.com) come `Principal` (l'entità attendibile) e includi uno `Condition` che corrisponda all'ID app assegnato all'IdP. Esempi di ruoli per diversi provider sono descritti più in Creazione di un ruolo per un provider di identità di terze parti.

4. Nell'applicazione, autenticare gli utenti con il provider di identità. Le specifiche della procedura variano sia in base al provider di identità in uso (Login with Amazon, Facebook o Google) sia in base alla piattaforma su cui viene eseguita l'app. Ad esempio, il metodo di autenticazione di un'app Android può differire da quello di un'app iOS o di un'app JavaScript-based Web.

   In genere, se l'utente non ha già effettuato l'accesso, il provider di identità si occupa di visualizzare una pagina di accesso. Dopo aver autenticato l'utente, il provider di identità restituisce all'app un token di autenticazione con le informazioni sull'utente. Le informazioni incluse dipendono dagli elementi esposti dal provider di identità e dalle informazioni che l'utente è disposto a condividere. Queste informazioni possono essere utilizzate nell'app.

5. Nell'app, effettuare una chiamata non firmata all'operazione `AssumeRoleWithWebIdentity` per richiedere le credenziali di sicurezza provvisorie. Nella richiesta, passi il token di autenticazione dell'IdP e specifichi l'Amazon Resource Name (ARN) per il ruolo IAM che hai creato per quell'IdP. AWS verifica che il token sia affidabile e valido e, in tal caso, restituisce le credenziali di sicurezza temporanee all'app che dispongono delle autorizzazioni per il ruolo indicato nella richiesta. La risposta include anche i metadati relativi all'utente forniti dal provider di identità, ad esempio l'ID utente univoco che il provider associa all'utente.

6. Utilizzando le credenziali di sicurezza temporanee della `AssumeRoleWithWebIdentity` risposta, l'app invia richieste firmate alle operazioni API. AWS Le informazioni sull'ID utente fornite dall'IdP possono distinguere gli utenti nella tua app. Ad esempio, è possibile inserire in cartelle Amazon S3 oggetti che includono l'ID utente come prefisso o suffisso. Ciò consente di creare policy di controllo degli accessi che bloccano la cartella in modo che solo l'utente con l'ID specificato possa accedervi. Per ulteriori informazioni, consulta AWS STS principi utente federati.

7. L'app dovrebbe memorizzare nella cache le credenziali di sicurezza temporanee in modo da non doverne ottenere di nuove ogni volta che ha bisogno di effettuare una richiesta ad AWS. Come impostazione predefinita, le credenziali sono valide per un'ora. Quando scadono (o prima), devi effettuare un'altra chiamata ad `AssumeRoleWithWebIdentity` per ottenere un nuovo set di credenziali di sicurezza temporanee. A seconda del provider di identità e di come gestisce i token, potrebbe essere necessario aggiornare il token del provider prima di effettuare una nuova

chiamata ad `AssumeRoleWithWebIdentity`, dato che anche i token di solito scadono dopo un determinato periodo di tempo. Se utilizzi l' AWS SDK per iOS o l' AWS SDK per Android, puoi utilizzare l'AmazonSTSCredentialsProviderazione, che gestisce le credenziali temporanee IAM, incluso l'aggiornamento se necessario.

Federazione OIDC

Immagina di creare un'applicazione che accede alle AWS risorse, come GitHub Actions che utilizza i flussi di lavoro per accedere ad Amazon S3 e DynamoDB.

Quando si utilizzano questi flussi di lavoro, si effettuano richieste ai AWS servizi che devono essere firmate con una chiave di accesso. AWS Tuttavia, consigliamo vivamente di non archiviare AWS le credenziali a lungo termine in applicazioni esterne. AWS Configura invece le tue applicazioni per richiedere credenziali di AWS sicurezza temporanee in modo dinamico quando necessario utilizzando la federazione OIDC. Le credenziali temporanee fornite sono associate a un AWS ruolo che dispone solo delle autorizzazioni necessarie per eseguire le attività richieste dall'applicazione.

Con la federazione OIDC, non è necessario creare il codice di accesso personalizzato o gestire le proprie identità utente personalizzate. È invece possibile utilizzare OIDC in applicazioni, come GitHub Actions o qualsiasi altro IdP compatibile con OpenID Connect (OIDC), con cui autenticarsi. AWS Ricevono un token di autenticazione, noto come JSON Web Token (JWT), e quindi scambiano quel token con credenziali di sicurezza temporanee in AWS quella mappa con un ruolo IAM con le autorizzazioni per utilizzare risorse specifiche nel tuo. Account AWS L'utilizzo di un IdP ti aiuta a mantenere la tua Account AWS sicurezza perché non devi incorporare e distribuire credenziali di sicurezza a lungo termine nella tua applicazione.

La federazione OIDC supporta sia l'autenticazione da macchina a macchina (come CI/CD pipeline, script automatici e applicazioni serverless) che l'autenticazione degli utenti umani. Nei casi di autenticazione di utenti umani in cui è necessario gestire la registrazione, l'accesso e i profili dell'utente, valuta l'utilizzo di Amazon Cognito come gestore di identità. Per informazioni dettagliate sull'utilizzo di Amazon Cognito con OIDC, consulta Amazon Cognito per applicazioni per dispositivi mobili.

> **Note**
>
> I JSON Web Tokens (JWT) emessi dai provider di identità OpenID Connect (OIDC) contengono una data di scadenza nell'attestazione `exp` che specifica quando scade il token. IAM offre una finestra di cinque minuti oltre la data di scadenza specificata nel JWT per

tenere conto dell'alterazione del clock, come consentito dallo standard OpenID Connect (OIDC) Core 1.0. Ciò significa che i JWT OIDC ricevuti da IAM dopo la scadenza, ma entro questo intervallo di cinque minuti, vengono accettati per un'ulteriore valutazione ed elaborazione.

Argomenti

- Risorse aggiuntive per la federazione OIDC
- Creare un provider di identità OpenID Connect (OIDC) in IAM
- Ottenere l'impronta digitale per un provider di identità OpenID Connect
- Identity-provider controlli per provider OIDC condivisi

Risorse aggiuntive per la federazione OIDC

Le risorse seguenti possono fornire ulteriori informazioni sulla federazione OIDC:

- Usa OpenID Connect all'interno dei tuoi flussi di lavoro configurando OpenID Connect in Amazon Web Services GitHub https://docs.github.com/en/actions/deployment/security-hardening-your-deployments/configuring-openid-connect-in-amazon-web-services
- Amazon Cognito Identity nella Guida alle librerie Amplify per Android e Guida all'identità di Amazon Cognito nella Guida alle librerie Amplify per Swift.
- Come utilizzare un ID esterno per concedere l'accesso alle AWS risorse sul AWS Security Blog fornisce indicazioni sulla configurazione sicura dell'accesso tra account e della federazione delle identità esterne.

Creare un provider di identità OpenID Connect (OIDC) in IAM

I provider di identità OIDC IAM sono entità in IAM che descrivono un servizio del provider di identità (IdP) esterno in grado di supportare lo standard OpenID Connect (OIDC), come Google o Salesforce. Utilizzi un provider di identità IAM OIDC quando desideri stabilire un rapporto di fiducia tra un OIDC-compatible IdP e il tuo. Account AWS Ciò è utile quando si crea un'app mobile o un'applicazione web che richiede l'accesso alle AWS risorse, ma non si desidera creare un codice di accesso personalizzato o gestire le proprie identità utente. Per ulteriori informazioni su questo scenario, consulta the section called "Federazione OIDC".

Puoi creare e gestire un provider di identità IAM OIDC utilizzando Console di gestione AWS gli AWS Command Line Interface strumenti per Windows PowerShell o l'API IAM.

Una volta creato un provider di identità OIDC IAM, dovrai creare uno o più ruoli IAM. Un ruolo è un'identità AWS che non dispone di credenziali proprie (come avviene per un utente). Tuttavia, in questo contesto, un ruolo viene assegnato in modo dinamico a un principale federato OIDC autenticato dall'IdP dell'organizzazione. Il ruolo consente al provider di identità dell'organizzazione di richiedere credenziali di sicurezza provvisorie per l'accesso a AWS. Le politiche assegnate al ruolo determinano le operazioni consentite agli utenti. AWS Per creare un ruolo per un provider di identità di terze parti, consulta Creazione di un ruolo per un provider di identità di terze parti.

> **Important**
>
> Quando si configurano politicy basate sull'identità per operazioni che supportano risorse `oidc-provider`, IAM valuta l'URL completo del provider di identità OIDC, inclusi i percorsi specificati. Se l'URL del tuo provider di identità OIDC ha un percorso, devi includere quel percorso nell'ARN `oidc-provider` come valore dell'elemento `Resource`. È inoltre possibile aggiungere una barra in avanti e un carattere jolly (`/*`) al dominio URL o usare caratteri jolly (`*` e `?`) in qualsiasi punto del percorso dell'URL. Se l'URL del provider di identità OIDC nella richiesta non corrisponde al valore impostato nell'elemento `Resource` della policy, la richiesta ha esito negativo.

Per risolvere i problemi più comuni relativi alla federazione IAM OIDC, consulta Risolvere gli errori relativi all'OIDC su re:POST. AWS

Argomenti

- Prerequisiti: convalida la configurazione del tuo provider di identità
- Creazione e gestione di un provider OIDC (Console)
- Creazione e gestione di un provider di identità OIDC IAM (AWS CLI)
- Creazione e gestione di un provider di identità OIDC (AWS API)

Prerequisiti: convalida la configurazione del tuo provider di identità

Prima di poter creare un provider di identità OIDC IAM, è necessario disporre delle seguenti informazioni dal proprio IdP. Per ulteriori informazioni su come ottenere le informazioni di configurazione del provider OIDC, consulta la documentazione dell'IdP.

1. Determina l'URL disponibile pubblicamente del tuo provider di identità OIDC. L'URL deve iniziare con https://. Secondo lo standard OIDC, i componenti del percorso sono consentiti ma i parametri di interrogazione no. In genere, l'URL è costituito solo da un nome host, ad esempio https://server.example.org o https://example.com. L'URL non deve contenere un numero di porta.
2. Aggiungi /.well- known/openid -configuration alla fine dell'URL del tuo provider di identità OIDC per visualizzare il documento di configurazione e i metadati del provider disponibili al pubblico. È necessario disporre di un documento di rilevamento in formato JSON con il documento di configurazione e i metadati del provider che possono essere recuperati dall'URL dell'endpoint di rilevamento del provider OpenID Connect.
3. Verifica che i seguenti valori siano inclusi nelle informazioni di configurazione del tuo provider. Se nella configurazione openid manca uno di questi campi, è necessario aggiornare il documento di rilevamento. Questo processo può variare in base al provider di identità, quindi segui la documentazione del tuo IdP per completare questa attività.

   - emittente: l'URL del dominio.
   - jwks_uri: l'endpoint JSON Web Key Set (JWKS) da cui IAM ottiene le chiavi pubbliche. Il provider di identità deve includere un endpoint JSON Web Key Set (JWKS) nella configurazione openid. Questo URI definisce dove ottenere le chiavi pubbliche utilizzate per verificare i token firmati dal provider di identità.

     > **Note**
     >
     > Il JSON Web Key Set (JWKS) deve contenere almeno una chiave e può avere un massimo di 100 chiavi RSA e 100 chiavi EC. Se il JWKS del provider di identità OIDC contiene più di 100 chiavi RSA o 100 chiavi EC, verrà restituita un'`InvalidIdentityToken`eccezione quando si utilizza l'operazione AssumeRoleWithWebIdentity API con un JWT firmato da un tipo di chiave che supera il limite di 100 chiavi. Ad esempio, se un JWT è firmato con l'algoritmo RSA e nel JWKS del provider sono presenti più di 100 chiavi RSA, verrà restituita un'eccezione `InvalidIdentityToken`.

   - claims_supported: informazioni sull'utente che ti aiutano a garantire che le risposte di autenticazione OIDC del tuo IdP contengano gli attributi richiesti utilizzati nelle politiche IAM per verificare le autorizzazioni per i principali federati OIDC. AWS Per un elenco delle chiavi di condizione IAM che possono essere utilizzate per le attestazioni, consulta Chiavi disponibili per AWS Federazione OIDC.

  - aud: devi determinare il valore dell'attestazione del pubblico dal tuo IdP nei JSON Web Tokens (JWT). L'attestazione del pubblico (aud) è specifica dell'applicazione e identifica i destinatari previsti del token. Quando registri un'app mobile o Web con un provider OpenID Connect, viene stabilito un ID client che identifica l'applicazione. L'ID client è un identificatore univoco per l'app passato nell'attestazione aud per l'autenticazione. Quando crei il tuo provider di identità OIDC IAM, l'attestazione aud deve corrispondere al valore Pubblico.
  - iat: le attestazioni devono includere un valore per `iat` che rappresenti l'ora in cui viene emesso il token ID.
  - iss: l'URL del provider di identità. L'URL deve iniziare con https://e deve corrispondere all'URL del provider fornito a IAM. In conformità con lo standard OIDC, i componenti del percorso sono consentiti, ma i parametri di query non lo sono. In genere, l'URL è costituito solo da un nome host, ad esempio o L'URL non deve contenere un numero di porta. https://server.example.org https://example.com.
- response_types_supported: id_token
- subject_types_supported: public
- id_token_signing_alg_values_supported: RS256, RS384, RS512, ES256, ES384, ES512

> **Note**
>
> Puoi includere ulteriori attestazioni come `my_custom_claim` nell'esempio seguente; tuttavia, la rivendicazione AWS STS verrà ignorata.

```
{
  "issuer": "https://example-domain.com",
  "jwks_uri": "https://example-domain.com/jwks/keys",
  "claims_supported": [
    "aud",
    "iat",
    "iss",
    "name",
    "sub",
    "my_custom_claim"
  ],
  "response_types_supported": [
    "id_token"
  ],
```

```
  "id_token_signing_alg_values_supported": [
    "RS256",
    "RS384",
    "RS512",
    "ES256",
    "ES384",
    "ES512"
  ],
  "subject_types_supported": [
    "public"
  ]
}
```

Creazione e gestione di un provider OIDC (Console)

Segui queste istruzioni per creare e gestire un provider di identità OIDC IAM nella Console di gestione AWS.

> **Important**
>
> Se utilizzi un provider di identità OIDC di Google, Facebook o Amazon Cognito, non creare un provider di identità IAM separato utilizzando questa procedura. Questi provider di identità OIDC sono già integrati AWS e sono disponibili per l'uso. Segui invece i passaggi per creare nuovi ruoli per il provider di identità e consulta Creare un ruolo per la federazione OpenID Connect (console).

Come creare un provider di identità OIDC IAM (console)

1. Prima di creare un provider di identità OIDC IAM, occorre registrare l'applicazione sul provider di identità per ricevere un ID client. L'ID client (noto anche come destinatario) è un identificatore univoco per l'app rilasciato durante la registrazione dell'app sul provider di identità. Per ulteriori informazioni su come ottenere un ID client, consulta la documentazione per l'IdP.

   > **Note**
   >
   > AWS protegge la comunicazione con i provider di identità OIDC (IdPs) utilizzando la nostra libreria di autorità di certificazione (CA) affidabili per verificare il certificato TLS dell'endpoint JSON Web Key Set (JWKS). Se il tuo IdP OIDC si basa su un certificato

non firmato da una di queste CA affidabili, solo allora proteggiamo la comunicazione utilizzando le impronte digitali impostate nella configurazione dell'IdP. AWS ricorreremo alla verifica tramite impronta digitale se non siamo in grado di recuperare il certificato TLS o se è richiesto TLS v1.3.

2. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
3. Nel riquadro di navigazione, scegli Provider di identità, quindi seleziona Aggiungi provider.
4. Per Configura provider, scegli OpenID Connect.
5. In Provider URL (URL provider), digitare l'URL del provider di identità. L'URL deve soddisfare queste restrizioni:
   - L'URL rileva la distinzione tra lettere maiuscole e minuscole.
   - L'URL deve iniziare con **`https://`**.
   - L'URL non deve contenere un numero di porta.
   - All'interno del tuo Account AWS, ogni provider di identità IAM OIDC deve utilizzare un URL univoco. Se provi a inviare un URL che è già stato utilizzato per un provider OpenID Connect nel Account AWS, riceverai un errore.
6. Per Audience, digita l'ID client dell'applicazione che hai registrato con l'IdP e che hai ricevuto e alla quale fai le richieste. Step 1 AWS Se si dispone di ID client aggiuntivi (noti anche come destinatari) per questo provider di identità, è possibile aggiungerli in un secondo momento nella pagina dei dettagli del provider.

   > **Note**
   >
   > Se il tuo token IdP JWT include l'attestazione `azp`, inserisci questo valore come valore Pubblico.
   > Se il tuo provider di identità OIDC imposta entrambi `aud` i `azp` claim nel token, AWS STS utilizzerà il valore del `azp` claim come claim. `aud`

7. (Facoltativo) Per Aggiungi tag, puoi aggiungere coppie chiave-valore per aiutarti a identificare e organizzare i tuoi. IdPs È inoltre possibile utilizzare i tag per controllare l'accesso alle risorse AWS . Per ulteriori informazioni sul tagging dei provider di identità OIDC IAM, consulta Aggiungere tag ai provider di identità OpenID Connect (OIDC). Seleziona Aggiungi tag. Immetti i valori per ogni coppia chiave-valore del tag.

8. Controlla le informazioni inserite. Quando hai finito, scegli Aggiungi provider. IAM proverà a recuperare e utilizzare l'impronta digitale della CA intermedia del certificato server IdP OIDC per creare il provider di identità OIDC IAM.

   > **Note**
   >
   > La catena di certificati del provider di identità OIDC deve iniziare con l'URL del dominio o dell'emittente, quindi il certificato intermedio e deve terminare con il certificato root. Se l'ordine della catena di certificati è diverso o se include certificati duplicati o aggiuntivi, riceverai un errore di mancata corrispondenza della firma e STS non riuscirà a convalidare il JSON Web Token (JWT). Correggi l'ordine dei certificati nella catena restituita dal server per risolvere l'errore. Per ulteriori informazioni sugli standard della catena di certificati, consulta certificate_list nella RFC 5246 sul sito Web della serie RFC.

9. Assegna un ruolo IAM al tuo provider di identità per concedere alle identità di utenti esterni gestite dal tuo provider di identità le autorizzazioni per accedere alle AWS risorse del tuo account. Per ulteriori informazioni sulla creazione di ruoli per la federazione delle identità, consulta Creazione di un ruolo per un provider di identità di terze parti.

   > **Note**
   >
   > L'OIDC IdPs utilizzato in una politica di fiducia del ruolo deve appartenere allo stesso account del ruolo che lo considera affidabile.

Come aggiungere o rimuovere un'identificazione personale per un provider di identità OIDC IAM (console)

> **Note**
>
> AWS protegge le comunicazioni con i provider di identità OIDC (IdPs) utilizzando la nostra libreria di autorità di certificazione (CA) affidabili per verificare il certificato TLS dell'endpoint JSON Web Key Set (JWKS). Se il tuo IdP OIDC si basa su un certificato non firmato da una di queste CA affidabili, solo allora proteggiamo la comunicazione utilizzando le impronte digitali impostate nella configurazione dell'IdP. AWS ricorreremo alla verifica tramite impronta digitale se non siamo in grado di recuperare il certificato TLS o se è richiesto TLS v1.3.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Identity providers (Provider di identità). Scegli quindi il nome del provider di identità IAM che desideri aggiornare.
3. Scegli la scheda Verifica dell'endpoint, quindi nella sezione Impronte digitali, scegli Gestisci. Per immettere un nuovo valore di identificazione personale, scegli Aggiungi identificazione personale. Per rimuovere un'identificazione personale, scegli Rimuovi accanto all'elemento che desideri rimuovere.

   > **Note**
   >
   > Un provider di identità OIDC IAM deve avere un numero di identificazioni personale compreso tra 1 e 5.

   Al termine, scegli Salva modifiche.

Come aggiungere un destinatario per un provider di identità OIDC IAM (console)

1. Nel pannello di navigazione, scegli Provider di identità, quindi scegli il nome del provider di identità IAM che desideri aggiornare.
2. Nella sezione Destinatari, scegli Operazioni e seleziona Aggiungi destinatario.
3. Digita l'ID client dell'applicazione che hai registrato con l'IdP e che hai ricevuto e alla quale verranno inoltrate le richiesteStep 1. AWS Quindi scegli Aggiungi destinatari.

   > **Note**
   >
   > Un provider di identità OIDC IAM deve avere un numero di audience compreso tra 1 e 100.

Come rimuovere un destinatario da un provider di identità OIDC IAM (console)

1. Nel pannello di navigazione, scegli Provider di identità, quindi scegli il nome del provider di identità IAM che desideri aggiornare.
2. Nella sezione Destinatari, seleziona il pulsante di opzione accanto al destinatario che desideri rimuovere, quindi seleziona Operazioni.

3. Scegli Rimuovi destinatario. Viene visualizzata una nuova finestra.
4. Se rimuovi un destinatario, le identità a esso federate non possono assumere ruoli associati al destinatario. Nella finestra, leggi l'avviso e conferma di volere rimuovere il destinatario digitando la parola `remove` nel campo.
5. Scegli Rimuovi per rimuovere il destinatario.

Come eliminare un provider di identità OIDC IAM (console)

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Identity providers (Provider di identità).
3. Seleziona la casella di controllo accanto al provider di identità IAM che desideri eliminare. Viene visualizzata una nuova finestra.
4. Conferma che desideri eliminare il provider digitando la parola `delete` nel campo. Quindi, scegli Elimina.

Creazione e gestione di un provider di identità OIDC IAM (AWS CLI)

È possibile utilizzare i seguenti AWS CLI comandi per creare e gestire i provider di identità OIDC IAM.

Per creare un provider di identità IAM OIDC (AWS CLI)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , emetti il seguente comando:
   - `aws iam list-open-id-connect-providers`
2. Per creare un nuovo provider di identità OIDC IAM, esegui il comando:
   - `aws iam create-open-id-connect-provider`

Per aggiornare l'elenco delle impronte digitali dei certificati del server per un provider di identità IAM OIDC esistente (AWS CLI)

- Per aggiornare l'elenco di identificazioni personali del certificato del server per un provider di identità OIDC IAM, esegui il comando:
  - `aws iam update-open-id-connect-provider-thumbprint`

Per etichettare un provider di identità IAM OIDC esistente (AWS CLI)

- Per aggiungere i tag a un provider di identità OIDC IAM esistente, esegui il comando:
  - `aws iam tag-open-id-connect-provider`

Per elencare i tag per un provider di identità IAM OIDC esistente (AWS CLI)

- Per elencare i tag per un provider di identità OIDC IAM esistente, esegui il comando:
  - `aws iam list-open-id-connect-provider-tags`

Per rimuovere i tag su un provider di identità IAM OIDC (AWS CLI)

- Per rimuovere i tag da un provider di identità OIDC IAM esistente, esegui il comando:
  - `aws iam untag-open-id-connect-provider`

Per aggiungere o rimuovere un ID cliente da un provider di identità IAM OIDC esistente (AWS CLI)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , emetti il seguente comando:
   - `aws iam list-open-id-connect-providers`
2. (Facoltativo) Per ottenere informazioni dettagliate su un provider di identità OIDC IAM, esegui il comando:
   - `aws iam get-open-id-connect-provider`
3. Per aggiungere un nuovo ID client a un provider di identità OIDC IAM esistente, esegui il comando:
   - `aws iam add-client-id-to-open-id-connect-provider`
4. Per rimuovere un client da un provider di identità OIDC IAM esistente, esegui il comando:
   - `aws iam remove-client-id-from-open-id-connect-provider`

Per eliminare un provider di identità IAM OIDC (AWS CLI)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , emetti il seguente comando:
   - `aws iam list-open-id-connect-providers`
2. (Facoltativo) Per ottenere informazioni dettagliate su un provider di identità OIDC IAM, esegui il comando:
   - `aws iam get-open-id-connect-provider`
3. Per eliminare un provider di identità OIDC IAM, esegui il comando:
   - `aws iam delete-open-id-connect-provider`

Creazione e gestione di un provider di identità OIDC (AWS API)

Puoi utilizzare i seguenti comandi dell'API IAM per creare e gestire provider OIDC.

Per creare un provider di identità IAM OIDC (AWS API)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , chiama la seguente operazione:
   - `ListOpenIDConnectProviders`
2. Per creare un nuovo provider di identità OIDC IAM, chiama la seguente operazione:
   - `CreateOpenIDConnectProvider`

Per aggiornare l'elenco delle impronte digitali dei certificati del server per un provider di identità IAM OIDC esistente (AWS API)

- Per aggiornare l'elenco di identificazioni personali del certificato del server per un provider di identità OIDC IAM, chiama la seguente operazione:
  - `UpdateOpenIDConnectProviderThumbprint`

Per etichettare un provider di identità IAM OIDC esistente (AWS API)

- Per aggiungere i tag a un provider di identità OIDC IAM, richiama la seguente operazione:

- `TagOpenIDConnectProvider`

Per elencare i tag per un provider di identità IAM OIDC esistente (AWS API)

- Per elencare i tag per un provider di identità OIDC IAM esistente, richiama la seguente operazione:
  - `ListOpenIDConnectProviderTags`

Per rimuovere i tag su un provider di identità IAM OIDC esistente (AWS API)

- Per rimuovere i tag da un provider di identità OIDC IAM esistente, richiama la seguente operazione:
  - `UntagOpenIDConnectProvider`

Per aggiungere o rimuovere un ID cliente da un provider di identità IAM OIDC esistente (AWS API)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , chiama la seguente operazione:
   - `ListOpenIDConnectProviders`
2. (Facoltativo) Per ottenere informazioni dettagliate su un provider di identità OIDC IAM, chiama la seguente operazione:
   - `GetOpenIDConnectProvider`
3. Per aggiungere un nuovo ID client a un provider di identità OIDC IAM esistente, chiama la seguente operazione:
   - `AddClientIDToOpenIDConnectProvider`
4. Per rimuovere un ID client da un provider di identità OIDC IAM esistente, chiama la seguente operazione:
   - `RemoveClientIDFromOpenIDConnectProvider`

Per eliminare un provider di identità IAM OIDC (AWS API)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , chiama la seguente operazione:
   - `ListOpenIDConnectProviders`
2. (Facoltativo) Per ottenere informazioni dettagliate su un provider di identità OIDC IAM, chiama la seguente operazione:
   - `GetOpenIDConnectProvider`
3. Per eliminare un provider di identità OIDC IAM, chiama la seguente operazione:
   - `DeleteOpenIDConnectProvider`

Ottenere l'impronta digitale per un provider di identità OpenID Connect

Durante la creazione di un provider di identità OpenID Connect (OIDC) in IAM, è necessario fornire un'impronta digitale per l'autorità di certificazione (CA) intermedia che ha firmato il certificato utilizzato dal gestore dell'identità digitale esterno. L'impronta digitale è una firma del certificato della CA utilizzata per emettere il certificato per l' OIDC-compatible IdP. Quando crei un provider di identità OIDC IAM, stai affidando alle identità autenticate da quell'IdP l'accesso al tuo. Account AWS Utilizzando l'impronta digitale del certificato della CA, si considera attendibile qualsiasi certificato emesso dalla CA con lo stesso nome DNS di quello registrato. Questo elimina la necessità di aggiornare i trust in ogni account quando si rinnova il certificato di firma del provider di identità.

> **Important**
>
> Nella maggior parte dei casi, il server di federazione utilizza due certificati diversi:
>
> - Il primo stabilisce una connessione HTTPS tra e il tuo IdP. AWS Questo dovrebbe essere emesso da un CA root pubblica nota, ad esempio AWS Certificate Manager. Ciò consente al cliente di verificare l'affidabilità e lo stato del certificato.
> - Il secondo è usato per crittografare i token e deve essere firmato da un CA radice privato o pubblico.

Puoi creare un provider di identità IAM OIDC con gli AWS Command Line Interface strumenti per Windows PowerShell o l'API IAM. Quando si utilizzano questi metodi, è possibile fornire un'impronta

digitale manualmente. Se scegli di non includere un'impronta digitale, IAM recupererà l'impronta della CA intermedia superiore del certificato server IdP OIDC. Se scegli di includere un'impronta digitale, sarà necessario ottenere l'impronta manualmente e fornirla ad AWS.

Quando crei un provider di identità OIDC IAM nella console IAM, IAM prova a recuperare l'impronta digitale della CA intermedia del certificato server IdP OIDC per tuo conto.

Consigliamo di ottenere manualmente anche l'impronta digitale dell'IdP OIDC e di verificare che IAM abbia recuperato l'impronta digitale corretta. Per ulteriori informazioni su come ottenere impronte digitali dei certificati, consulta le seguenti sezioni.

> **Note**
>
> AWS protegge la comunicazione con i provider di identità OIDC (IdPs) utilizzando la nostra libreria di autorità di certificazione (CA) affidabili per verificare il certificato TLS dell'endpoint JSON Web Key Set (JWKS). Se il tuo IdP OIDC si basa su un certificato non firmato da una di queste CA affidabili, solo allora proteggiamo la comunicazione utilizzando le impronte digitali impostate nella configurazione dell'IdP. AWS ricorreremo alla verifica tramite impronta digitale se non siamo in grado di recuperare il certificato TLS o se è richiesto TLS v1.3.

> **Important**
>
> Se il tuo provider di identità OIDC utilizza host diversi per l'endpoint di rilevamento (l'URL dell'emittente) e l'endpoint JWKS (`jwks_uri`), devi includere l'impronta personale di entrambi gli endpoint nell'elenco delle impronte digitali del tuo provider OIDC. AWS convalida l'impronta digitale di entrambi gli endpoint quando si ricorre alla verifica tramite impronta digitale. Se nell'elenco configurato manca una delle impronte digitali, l'autenticazione ha esito negativo.
>
> La maggior parte dei provider OIDC ospita entrambi gli endpoint sullo stesso dominio, quindi un'unica impronta digitale copre entrambi. È necessario aggiungere un'impronta personale separata solo se il provider utilizza host o certificati diversi per i due endpoint.

Ottenere l'impronta digitale del certificato

Utilizzare un browser Web e lo strumento a riga di comando OpenSSL per ottenere l'impronta digitale per un provider OIDC. Tuttavia, non è necessario ottenere manualmente l'impronta digitale del

certificato per creare un provider di identità OIDC IAM. Puoi utilizzare la procedura seguente per ottenere l'impronta digitale del certificato del provider OIDC.

Per ottenere l'identificazione personale per un provider di identità OIDC

1. Prima di poter ottenere l'identificazione personale per un provider di identità OIDC, è necessario ottenere lo strumento a riga di comando OpenSSL. È possibile utilizzare questo strumento per scaricare la catena di certificati del provider di identità OIDC e produrre un'identificazione personale del certificato finale nella catena di certificati. Se è necessario installare e configurare OpenSSL, seguire le istruzioni in Installare OpenSSL e Configurare OpenSSL.
2. Inizia con l'URL del provider di identità OIDC (ad esempio, `https://server.example.com`) e quindi aggiungi `/.well-known/openid-configuration` per formare l'URL per il documento di configurazione del provider di identità, nel modo seguente:

   `https://server.example.com/.well-known/openid-configuration`

   Apri questo URL in un browser web, sostituendolo *`server.example.com`* con il nome del tuo server IdP.
3. Nel documento visualizzato, utilizza l'opzione Trova del browser Web per individuare il testo `"jwks_uri"`. Subito dopo il testo `"jwks_uri"` sono presenti due punti (:) seguiti da un URL. Copiare il nome di dominio completo dell'URL. Non includere il percorso `https://` o qualsiasi altro percorso dopo il dominio di primo livello.

   ```
   {
    "issuer": "https://accounts.example.com",
    "authorization_endpoint": "https://accounts.example.com/o/oauth2/v2/auth",
    "device_authorization_endpoint": "https://oauth2.exampleapis.com/device/code",
    "token_endpoint": "https://oauth2.exampleapis.com/token",
    "userinfo_endpoint": "https://openidconnect.exampleapis.com/v1/userinfo",
    "revocation_endpoint": "https://oauth2.exampleapis.com/revoke",
    "jwks_uri": "https://www.exampleapis.com/oauth2/v3/certs",
   ...
   ```

   > **Note**
   >
   > Se gli endpoint discovery e JWKS utilizzano host diversi, annota il nome di dominio completo riportato nell'URL dell'emittente (l'URL in cui hai utilizzatoStep 2, senza il percorso). `/.well-known/openid-configuration` Se questo dominio è diverso

dal `jwks_uri` dominio che hai copiato in questa fase, esegui la procedura Step 6 per entrambi i domini e includi entrambe Step 4 le impronte digitali quando crei il tuo provider OIDC.

4. Utilizzare lo strumento a riga di comando OpenSSL per eseguire il seguente comando. Sostituiscilo *`keys.example.com`* con il nome di dominio che hai ottenuto. Step 3

```
openssl s_client -servername keys.example.com -showcerts -
connect keys.example.com:443
```

5. Nella finestra di comando, scorrere verso l'alto fino a visualizzare un certificato simile al seguente esempio. Se viene visualizzato più di un certificato, individua l'ultimo certificato visualizzato (nella parte inferiore dell'output di comando). Contiene il certificato della migliore CA intermedia nella catena della certification authority.

```
-----BEGIN CERTIFICATE-----
 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
 YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
 21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
 rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
 nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
 NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
 -----END CERTIFICATE-----
```

Copiare il certificato (incluse le righe `-----BEGIN CERTIFICATE-----` e `-----END CERTIFICATE-----`) e incollarlo in un file di testo. Salvare quindi il file con il nome **`certificate.crt`**.

> **Note**
>
> La catena di certificati del provider di identità OIDC deve iniziare con l'URL del dominio o dell'emittente, includere eventuali certificati intermedi (se presenti) e terminare con

il certificato root. Se l'ordine della catena di certificati è diverso o se include certificati duplicati o aggiuntivi, riceverai un errore di mancata corrispondenza della firma e STS non riuscirà a convalidare il JSON Web Token (JWT). Correggi l'ordine dei certificati nella catena restituita dal server per risolvere l'errore. Per ulteriori informazioni sugli standard della catena di certificati, consulta certificate_list nella RFC 5246 sul sito Web della serie RFC.

6. Utilizzare lo strumento a riga di comando OpenSSL per eseguire il seguente comando.

```
openssl x509 -in certificate.crt -fingerprint -sha1 -noout
```

La finestra di comando visualizza l'identificazione personale del certificato, simile a quella dell'esempio seguente:

```
SHA1 Fingerprint=99:0F:41:93:97:2F:2B:EC:F1:2D:DE:DA:52:37:F9:C9:52:F2:0D:9E
```

Rimuovere i due punti (:) da questa stringa per ottenere l'identificazione personale finale, come segue:

```
990F4193972F2BECF12DDEDA5237F9C952F20D9E
```

> **Note**
>
> Se il tuo endpoint di scoperta utilizza un host diverso dall'endpoint JWKS, Step 4 ripeti l'Step 6utilizzo del dominio dell'endpoint di scoperta (il dominio URL dell'emittente). Includi entrambe le impronte digitali quando crei o aggiorni il tuo provider di identità OIDC utilizzando o. CreateOpenIDConnectProvider UpdateOpenIDConnectProviderThumbprint

7. Se stai creando il provider di identità OIDC IAM con Tools for Windows o l' AWS CLI API IAM PowerShell, fornire un'impronta personale è facoltativo. Se scegli di non includere un'impronta digitale durante la creazione, IAM recupererà l'impronta della CA intermedia superiore del certificato server IdP OIDC. Dopo aver creato il provider di identità OIDC IAM, potrai confrontare questa impronta digitale con quella recuperata da IAM.

   Se stai creando il provider di identità OIDC IAM nella console IAM, la console prova a recuperare l'impronta digitale della CA intermedia del certificato server IdP OIDC. Puoi confrontare questa

impronta digitale con quella recuperata da IAM. Dopo aver creato il provider di identità OIDC IAM, puoi visualizzare l'impronta digitale del provider di identità OIDC IAM nella scheda Verifica dell'endpoint nella pagina Riepilogo della console del provider OIDC.

> **Important**
>
> Se l'impronta digitale ottenuta non corrisponde a quella riportata nei dettagli del provider di identità OIDC IAM, il provider OIDC non dovrebbe essere utilizzato. Sarà necessario invece eliminare il provider OIDC creato, quindi riprovare a creare il provider OIDC dopo un po' di tempo. Verifica che le impronte digitali corrispondano prima di utilizzare il provider. Se dopo un secondo tentativo non vi è comunque corrispondenza tra le identificazioni personali, accedere al forum di IAM per contattare AWS.

Installare OpenSSL

Se OpenSSL non è già installato, segui le istruzioni in questa sezione.

Come installare OpenSSL su Linux e Unix

1. Vai a OpenSSL: Source, Tarballs (). https://openssl.org/source/
2. Scarica l'origine più recente e compila il pacchetto.

Per installare OpenSSL in ambiente Windows

1. Vai a OpenSSL: Binary Distributions (https://wiki.openssl.org/index.php/Binaries) per un elenco di siti da cui è possibile installare la versione per Windows.
2. Segui le istruzioni sul sito selezionato per avviare l'installazione.
3. Se viene richiesto di installare Microsoft Visual C++ 2008 Redistributables e questo non è già installato sul tuo sistema, scegli il link di download appropriato per il tuo ambiente. Segui le istruzioni fornite dalla Installazione guidata di Microsoft Visual C++ 2008 Redistributable.

> **Note**
>
> Se non sei sicuro che Microsoft Visual C++ 2008 Redistributables sia già installato nel sistema, puoi provare a installare prima OpenSSL. Il programma di installazione di OpenSSL visualizza un avviso se Microsoft Visual C++ 2008 Redistributables non è

ancora installato. Assicurati di installare l'architettura (32 bit o 64 bit) che corrisponde alla versione di OpenSSL installata.

4. Dopo aver installato Microsoft Visual C++ 2008 Redistributables, seleziona la versione appropriata dei file binari OpenSSL per l'ambiente e salva il file in locale. Avvia l'Installazione guidata di OpenSSL.
5. Segui le istruzioni descritte in Installazione guidata di OpenSSL.

Configurare OpenSSL

Prima di utilizzare i comandi OpenSSL, è necessario configurare il sistema operativo in modo che contenga le informazioni sulla posizione in cui è installato OpenSSL.

Come configurare OpenSSL su Linux o Unix

1. Alla riga di comando, imposta la variabile `OpenSSL_HOME` sulla posizione dell'installazione di OpenSSL:

```
$ export OpenSSL_HOME=path_to_your_OpenSSL_installation
```

2. Configura il percorso in modo da includere l'installazione di OpenSSL:

```
$ export PATH=$PATH:$OpenSSL_HOME/bin
```

> **Note**
>
> Eventuali modifiche apportate alle variabili di ambiente con il comando `export` sono valide solo per la sessione corrente. Puoi apportare modifiche permanenti alle variabili di ambiente impostandole nel file di configurazione della shell. Per ulteriori informazioni, consulta la documentazione relativa al sistema operativo in uso.

Come configurare OpenSSL su Windows

1. Apri una finestra del prompt dei comandi.
2. Configura la variabile `OpenSSL_HOME` sulla posizione dell'installazione di OpenSSL:

```
C:\> set OpenSSL_HOME=path_to_your_OpenSSL_installation
```

3. Imposta la variabile `OpenSSL_CONF` sulla posizione del file di configurazione nell'installazione di OpenSSL:

```
C:\> set OpenSSL_CONF=path_to_your_OpenSSL_installation\bin\openssl.cfg
```

4. Configura il percorso in modo da includere l'installazione di OpenSSL:

```
C:\> set Path=%Path%;%OpenSSL_HOME%\bin
```

> **Note**
>
> Eventuali modifiche apportate alle variabili di ambiente Windows in una finestra del prompt dei comandi sono valide solo per la sessione della riga di comando corrente. È possibile apportare modifiche permanenti alle variabili di ambiente impostandole come proprietà di sistema. Le procedure esatte dipendono dalla versione di Windows in uso. (Ad esempio, su Windows 7, apri Pannello di controllo, Sistema e sicurezza, Sistema. Quindi scegli Impostazioni di sistema avanzate, scheda Avanzate, Variabili di ambiente.) Per ulteriori informazioni, consulta la documentazione di Windows.

Identity-provider controlli per provider OIDC condivisi

Per i provider di identità OpenID Connect (OIDC) condivisi riconosciuti, IAM richiede una valutazione esplicita di affermazioni specifiche nelle policy di trust dei ruoli. IdPs Queste attestazioni obbligatorie, chiamate controlli del provider di identità, vengono valutate da IAM durante la creazione dei ruoli e gli aggiornamenti delle policy di attendibilità. Se la policy di attendibilità dei ruoli non valuta i controlli richiesti dall'IdP OIDC condiviso, non sarà possibile creare o aggiornare ruoli. Questo garantisce che solo le identità autorizzate dell'organizzazione prevista possano assumere ruoli e accedere alle risorse AWS . Questo controllo di sicurezza è fondamentale quando i provider OIDC sono condivisi tra più clienti AWS .

Identity-provider i controlli non verranno valutati da IAM per le politiche di trust dei ruoli OIDC esistenti. Per qualsiasi modifica alla policy di attendibilità dei ruoli per i ruoli OIDC esistenti, IAM richiederà che i controlli relativi ai provider di identità siano inclusi nella policy stessa.

Tipi di provider OIDC

IAM classifica i provider di identità OIDC in due tipi distinti: privati e condivisi. Un IdP OIDC privato può essere di proprietà e gestito da una singola organizzazione o può essere un tenant di un provider SaaS, con l'URL dell'emittente OIDC che funge da identificatore univoco specifico per tale organizzazione. Al contrario, un IdP OIDC condiviso viene impiegato da più organizzazioni e può avere lo stesso URL dell'emittente OIDC per tutte quelle che lo utilizzano.

La tabella seguente illustra le principali differenze tra i provider OIDC privati e condivisi:

Caratteristica	Provider OIDC privato	Provider OIDC condiviso
Emittente	Univoco per l'organizzazione	Condiviso tra più organizza zioni
Informazioni di tenancy	Comunicate tramite un emittente unico	Comunicate tramite attestazi oni nei JWT
Requisiti della policy di attendibilità	Non è richiesta la valutazione di un'attestazione specifica	È richiesta la valutazione di attestazioni specifiche

Provider di identità OIDC condivisi con controlli dei provider di identità

Quando crei o modifichi un provider OIDC in IAM, il sistema identifica e valuta automaticamente le attestazioni richieste per i provider OIDC condivisi e riconosciuti. Se i controlli del provider di identità non sono configurati nella policy di fiducia dei ruoli, la creazione o l'aggiornamento del ruolo avrà esito negativo e verrà generato un errore. MalformedPolicyDocument

La tabella seguente elenca i provider OIDC condivisi che richiedono i controlli del provider di identità nelle policy di attendibilità dei ruoli e informazioni aggiuntive per aiutarti a configurare i controlli del provider di identità.

IdP OIDC	URL OIDC	Attestazione di tenancy	Attestazioni obbligato rie
Amazon Cognito	`cognito-i dentity.a mazonaws.com`	aud	`cognito-i dentity.a`

IdP OIDC	URL OIDC	Attestazione di tenancy	Attestazioni obbligatorie
			`mazonaws.com:aud`
Azure Sentinel	`https://sts.windows.net/33e01921-4d64-4f8c-a055-5bdaffd5e33d`	`sts:RoleSessionName`	`sts:RoleSessionName`
Buildkite	`https://agent.buildkite.com`	sub	`agent.buildkite.com:sub`
Codefresh SaaS	`https://oidc.codefresh.io`	sub	`oidc.codefresh.io:sub`
DVC Studio	`https://studio.datachain.ai/api`	sub	`studio.datachain.ai/api:sub`

IdP OIDC	URL OIDC	Attestazione di tenancy	Attestazioni obbligatorie

GitHub azioni	`https://token.actions.githubusercontent.com`	sub o job_workflow_ref	`token.actions.githubusercontent.com:sub`
			
			or
			
			`token.actions.githubusercontent.com:job_workflow_ref`

GitHub registro di controllo (streaming)	`https://oidc-configuration.audit-log.githubusercontent.com`	sub	`oidc-configuration.audit-log.githubusercontent.com:sub`
GitHub vstoken	`https://vstoken.actions.githubusercontent.com`	sub	`vstoken.actions.githubusercontent.com:sub`
GitLab	`https://gitlab.com`	sub	`gitlab.com:sub`

IdP OIDC	URL OIDC	Attestazione di tenancy	Attestazioni obbligatorie
IBM Turbonomic SaaS*	• https://rh-oidc.s3.us-east-1.amazonaws.com/22ejnvnnturfmt6km08iidd0nt4hekbn7 • https://rh-oidc.s3.us-east-1.amazonaws.com/23e3sd27sju1hoou6ohfs68vbno607tr • https://rh-oidc.s3.us-east-1.amazonaws.com/23ne21h005qjl3n33d8dui5dlrmv2tmg • https://rh-oidc.s3.us-east-1.amazonaws.com/24jrf12m5dj7ljlfb4ta2frhrcoadm26	sub	• rh-oidc.s3.us-east-1.amazonaws.com/22ejnvnnturfmt6km08idd0nt4hekbn7:sub • rh-oidc.s3.us-east-1.amazonaws.com/23e3sd27sju1hoou6ohfs68vbno607tr:sub • rh-oidc.s3.us-east-1.amazonaws.com/23ne21h005qjl3n33d8dui5dlrmv2tmg:sub • rh-oidc.s3.us-east-1.amazonaws.com/24jrf12m5dj7ljlfb4ta2frhrcoadm26:sub

IdP OIDC	URL OIDC	Attestazione di tenancy	Attestazioni obbligatorie
	• https://oidc.op1.openshiftapps.com/2f785sojlpb85i7402pk3qogugim5nfb • https://oidc.op1.openshiftapps.com/2c51blsaqa9gkjt0o9rt11mle8mmropu		• oidc.op1.openshiftapps.com/2f785sojlpb85i7402pk3qogugim5nfb:sub • oidc.op1.openshiftapps.com/2c51blsaqa9gkjt0o9rt11mle8mmropu:sub
Pulumi Cloud	https://api.pulumi.com/oidc	aud	api.pulumi.com/oidc:aud
sandboxes.cloud	https://sandboxes.cloud	aud	sandboxes.cloud:aud
Scalr	https://scalr.io	sub	scalr.io:sub
Shisho Cloud	https://tokens.cloud.shisho.dev	sub	tokens.cloud.shisho.dev:sub
Terraform Cloud	https://app.terraform.io	sub	app.terraform.io:sub

IdP OIDC	URL OIDC	Attestazione di tenancy	Attestazioni obbligatorie
Upbound	`https://proidc.upbound.io`	sub	`proidc.upbound.io:sub`
Endpoint globale Vercel	`https://oidc.vercel.com`	aud	`oidc.vercel.com:aud`

* IBM Turbonomic aggiorna periodicamente l'URL dell'emittente OIDC con nuove versioni della piattaforma. Se necessario, aggiungeremo altri emittenti OIDC Turbonomic come provider condiviso.

Per ogni nuovo OIDC IdPs che IAM identifica come condiviso, i controlli richiesti dai provider di identità per le politiche di fiducia dei ruoli verranno documentati e applicati in modo simile.

Risorse aggiuntive

Risorse aggiuntive:

- Per ulteriori informazioni sulla creazione di un ruolo IAM per la federazione OIDC, consulta Creare un ruolo per la federazione OpenID Connect (console).
- Per un elenco delle chiavi di condizione IAM che possono essere utilizzate per le attestazioni, consulta Chiavi disponibili per AWS Federazione OIDC.

Federazione SAML 2.0

AWS supporta la federazione delle identità con SAML 2.0 (Security Assertion Markup Language 2.0), uno standard aperto utilizzato da molti provider di identità (IdPs). Questa funzionalità abilita il single sign-on (SSO) federato, in modo che gli utenti possano accedere Console di gestione AWS o chiamare le operazioni AWS API senza dover creare un utente IAM per tutti i membri dell'organizzazione. Utilizzando SAML, puoi semplificare il processo di configurazione della federazione con AWS, poiché puoi utilizzare il servizio dell'IdP invece di scrivere un codice proxy di identità personalizzato. https://docs.aws.amazon.com/STS/latest/UsingSTS/CreatingFedTokens.html

> **Note**
>
> La federazione delle identità SAML IAM supporta risposte SAML crittografate da provider di identità SAML-based federati (). IdPs Il Centro identità IAM e Amazon Cognito non supportano le asserzioni SAML crittografate dei provider di identità IAM SAML.
> Puoi aggiungere indirettamente il supporto per le asserzioni SAML crittografate alla federazione dei pool di identità di Amazon Cognito con i pool di utenti di Amazon Cognito. I pool di utenti dispongono di una federazione SAML che è indipendente dalla federazione IAM SAML e supporta la firma e la crittografia SAML. Sebbene questa funzionalità non si estenda direttamente ai pool di identità, i pool di utenti possono riguardare i pool IdPs di identità. Per utilizzare la crittografia SAML con pool di identità, aggiungi un provider SAML con crittografia a un pool di utenti che è un IdP a un pool di identità.
> Il tuo provider SAML deve essere in grado di crittografare le asserzioni SAML con una chiave fornita dal tuo pool di utenti. I pool di utenti non accetteranno asserzioni crittografate con un certificato fornito da IAM.

La federazione IAM supporta i casi d'uso indicati di seguito.

- Accesso federato per consentire a un utente o a un'applicazione dell'organizzazione di chiamare le operazioni AWS API. Questo caso d'uso è discusso nella sezione seguente. viene utilizzata un'asserzione SAML (come parte della risposta di autenticazione) generata nell'organizzazione per ottenere credenziali di sicurezza temporanee. Questo scenario è analogo ad altri scenari di federazione supportati da IAM, come descritto in Richiedere credenziali di sicurezza temporanee e Federazione OIDC. Tuttavia, SAML 2.0, basato IdPs nell'organizzazione, gestisce molti dettagli in fase di esecuzione per eseguire l'autenticazione e il controllo delle autorizzazioni.
- Web-basedsingle sign-on (SSO) da inviare all'organizzazione. Console di gestione AWS Gli utenti possono accedere a un portale dell'organizzazione ospitato da un IdP compatibile con SAML 2.0, selezionare un'opzione a cui accedere ed essere reindirizzati alla AWS console senza dover fornire ulteriori informazioni di accesso. Puoi utilizzare un IdP SAML di terze parti per stabilire l'accesso SSO alla console o creare un IdP personalizzato per consentire l'accesso alla console per utenti esterni. Per ulteriori informazioni sulla creazione di un provider di identità personalizzato, consulta Abilita l'accesso personalizzato del broker di identità al AWS console.

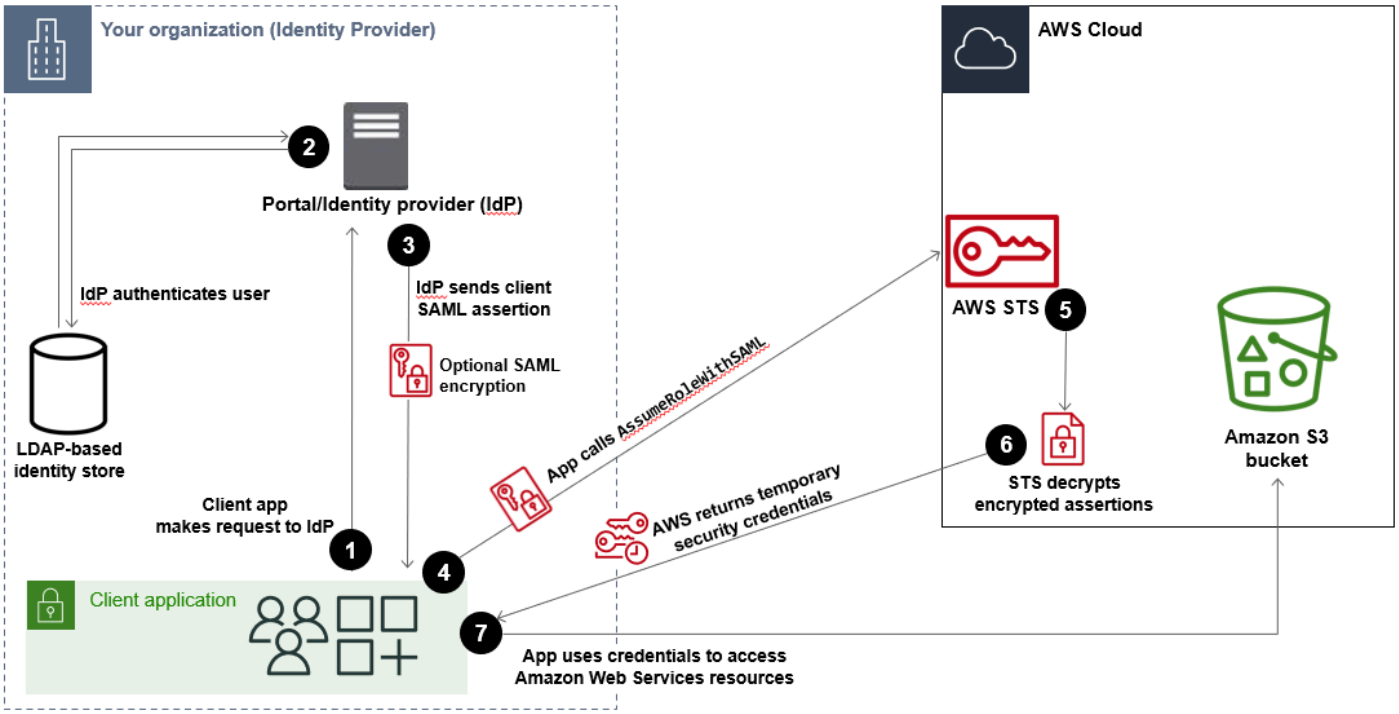
Argomenti

- Utilizzo della federazione per l'accesso tramite API a SAML-based AWS

- Panoramica della configurazione della federazione basata su SAML 2.0
- Panoramica del ruolo per consentire l'accesso al tuo SAML-federated AWS risorse
- Identificazione univoca degli SAML-based utenti nella federazione
- Creare un provider di identità SAML in IAM
- Configurare il provider di identità SAML 2.0 con una relazione di attendibilità della parte affidabile e aggiunta di attestazioni
- Integra fornitori di soluzioni SAML di terze parti con AWS
- Configurare le asserzioni SAML per la risposta di autenticazione
- Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS
- Visualizzare una risposta SAML nel browser

Utilizzo della federazione per l'accesso tramite API a SAML-based AWS

Supponi di voler fornire ai dipendenti un modo per copiare i dati dai loro computer a una cartella di backup. È possibile creare un'applicazione che gli utenti possono eseguire sui propri computer. Sul backend, l'applicazione legge e scrive oggetti in un bucket Amazon S3. Gli utenti non hanno accesso diretto a AWS. viene utilizzato invece il seguente processo:



1. Un utente dell'organizzazione utilizza un'app client per richiedere l'autenticazione dal provider di identità della propria organizzazione.
2. Il provider di identità autentica l'utente rispetto all'archivio identità organizzazione.
3. Il provider di identità crea un'asserzione SAML con informazioni sull'utente e invia l'asserzione all'app client. Quando abiliti la crittografia SAML per il tuo IdP IAM SAML, questa asserzione viene crittografata dal tuo IdP esterno.
4. L'app client chiama l' AWS STS `AssumeRoleWithSAML`API, trasmettendo l'ARN del provider SAML, l'ARN del ruolo da assumere e l'asserzione SAML da IdP. Se la crittografia è abilitata, l'asserzione trasmessa tramite l'app client rimane crittografata durante il transito.
5. (Facoltativo) AWS STS utilizza la chiave privata caricata dal tuo IdP esterno per decrittografare l'asserzione SAML crittografata.
6. La risposta dell'API all'app client include credenziali di sicurezza temporanee.
7. L'app client utilizza le credenziali di sicurezza temporanee per chiamare le operazioni dell'API di Amazon S3.

Panoramica della configurazione della federazione basata su SAML 2.0

Prima di poter utilizzare la federazione basata su SAML 2.0 come descritto nello scenario e nel diagramma precedenti, devi configurare l'IdP dell'organizzazione e il tuo in modo che si fidino l'uno dell'altro. Account AWS Di seguito è descritta la procedura generale per la configurazione di questo tipo di attendibilità. All'interno dell'organizzazione, è necessario disporre di un provider di identità che supporti SAML 2.0, come Microsoft Active Directory Federation Service (ADFS, parte di Windows Server), Shibboleth o di un altro provider compatibile con SAML 2.0.

> **Note**
>
> Per migliorare la resilienza della federazione, ti consigliamo di configurare l'IdP e la federazione AWS per supportare più endpoint di accesso SAML. Per informazioni dettagliate, consulta l'articolo del AWS Security Blog How to use regional SAML endpoint for failover.

Configura l'IdP della tua organizzazione e AWS fidarsi l'uno dell'altro

1. Registrati AWS come fornitore di servizi (SP) presso l'IdP della tua organizzazione. Utilizzo del documento dei metadati SAML generato da `https://region-code.signin.aws.amazon.com/static/saml-metadata.xml`

Per un elenco dei *region-code* valori possibili, consulta la colonna Regione negli AWS Sign-In endpoints endpoints

Facoltativamente, puoi utilizzare il documento di metadati SAML di. `https://signin.aws.amazon.com/static/saml-metadata.xml`

2. Utilizzando il provider di identità dell'organizzazione, genera un file XML di metadati SAML equivalenti che possono descrivere l'IdP come provider di identità IAM in AWS. Deve includere il nome dell'emittente, una data di creazione, una data di scadenza e le chiavi che AWS possono essere utilizzate per convalidare le risposte di autenticazione (asserzioni) dell'organizzazione.

   Se consenti l'invio di asserzioni SAML crittografate dal tuo IdP esterno, devi generare un file di chiave privata utilizzando l'IdP della tua organizzazione e caricare questo file nella tua configurazione SAML IAM in formato .pem. AWS STS ha bisogno di questa chiave privata per decrittografare le risposte SAML che corrispondono alla chiave pubblica caricata sul tuo IdP.

   > **Note**
   >
   > Come definito dal SAML V2.0 Metadata Interoperability Profile Version 1.0, IAM non valuta né interviene sulla scadenza dei certificati nei documenti di metadati SAML. X.509 Se sei preoccupato per i X.509 certificati scaduti, ti consigliamo di monitorare le date di scadenza dei certificati e di ruotare i certificati in base alle politiche di governance e sicurezza della tua organizzazione.

3. Nella console IAM, crea un'entità provider di identità SAML. Come parte di questo processo, è possibile caricare il documento dei metadati SAML e la chiave di decrittografia privata prodotto dal provider di identità nell'organizzazione in Step 2. Per ulteriori informazioni, consulta Creare un provider di identità SAML in IAM.
4. In IAM, vengono creati uno o più ruoli IAM. Nella politica di fiducia del ruolo, imposti il provider SAML come principale, il che stabilisce una relazione di fiducia tra la tua organizzazione e. AWS La policy di autorizzazione del ruolo stabilisce le operazioni che gli utenti dell'organizzazione possono effettuare in AWS. Per ulteriori informazioni, consulta Creazione di un ruolo per un provider di identità di terze parti.

> **Note**
>
> Gli IDP SAML utilizzati in una policy di attendibilità dei ruoli devono appartenere allo stesso account in cui si trova il ruolo.

5. Nel provider di identità dell'organizzazione, si definiscono asserzioni che associano utenti o gruppi dell'organizzazione ai ruoli IAM. Nota che i diversi utenti e gruppi dell'organizzazione potrebbero essere mappati a diversi ruoli IAM. La procedura per eseguire la mappatura dipende dal provider di identità che si sta utilizzando. Nello scenario precedente che utilizza una cartella Amazon S3 per gli utenti, è possibile che tutti gli utenti dispongano della mappatura allo stesso ruolo che fornisce le autorizzazioni Amazon S3. Per ulteriori informazioni, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

   Se il tuo IdP abilita l'SSO AWS sulla console, puoi configurare la durata massima delle sessioni della console. Per ulteriori informazioni, consulta Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS.

6. Nell'applicazione che stai creando, chiami l' Servizio di token di sicurezza AWS `AssumeRoleWithSAML`API, passandole l'ARN del provider SAML in cui hai creatoStep 3, l'ARN del ruolo in cui assumi la creazione e l'asserzione SAML sull'utente corrente che ricevi dal tuo IdP. Step 4 AWS si assicura che la richiesta di assumere il ruolo provenga dall'IdP a cui fa riferimento il provider SAML.

   Per ulteriori informazioni, consulta AssumeRoleWithSAML nella documentazione di riferimento dell'API Servizio di token di sicurezza AWS .

7. Se la richiesta ha esito positivo, l'API restituisce una serie di credenziali di sicurezza temporanee, che l'applicazione può utilizzare per inviare richieste firmate ad AWS. L'applicazione contiene informazioni sull'utente corrente e può accedere a cartelle specifiche dell'utente in Amazon S3, come descritto nello scenario precedente.

Panoramica del ruolo per consentire l'accesso al tuo SAML-federated AWS risorse

I ruoli creati in IAM definiscono ciò che i principali federati SAML dell'organizzazione possono fare in AWS. Quando si creano policy di affidabilità per il ruolo, è necessario specificare il provider SAML creato in precedenza come `Principal`. È inoltre possibile estendere la policy di affidabilità con un elemento `Condition` per permettere solo agli utenti che corrispondono ad alcuni attributi SAML di accedere al ruolo. Ad esempio, puoi specificare che solo gli utenti la cui affiliazione SAML è

`staff` (come affermato dahttps://openidp.feide.no) possono accedere al ruolo, come illustrato dalla seguente politica di esempio:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"Federated": "arn:aws:iam::111122223333:saml-provider/
ExampleOrgSSOProvider"},
    "Action": "sts:AssumeRoleWithSAML",
    "Condition": {
      "StringEquals": {
        "saml:aud": "https://us-east-1.signin.aws.amazon.com/saml",
        "saml:iss": "https://openidp.feide.no"
      },
      "ForAllValues:StringLike": {"saml:edupersonaffiliation": ["staff"]}
    }
  }]
}
```

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws-cn:iam::111122223333:saml-provider/
ExampleOrgSSOProvider"
            },
            "Action": "sts:AssumeRoleWithSAML",
            "Condition": {
                "StringEquals": {
                    "saml:aud": "https://cn-north-1.signin.amazonaws.cn/saml",
                    "saml:iss": "https://openidp.feide.no"
                },
                "ForAllValues:StringLike": {
```

```
                    "saml:edupersonaffiliation": [
                        "staff"
                    ]
                }
            }
        }
    ]
}
```

> **Note**
>
> Gli IDP SAML utilizzati in una policy di attendibilità dei ruoli devono appartenere allo stesso account in cui si trova il ruolo.

La chiave di contesto `saml:aud` nella policy specifica l'URL visualizzato dal browser quando si accede alla console. L'URL dell'endpoint di accesso deve corrispondere all'attributo destinatario del provider di identità. Puoi includere gli URL di accesso all'interno di determinate aree geografiche. AWS consiglia di utilizzare gli endpoint regionali anziché gli endpoint globali per migliorare la resilienza della federazione. Se hai configurato un solo endpoint, non sarai in grado di federarti AWS nell'improbabile eventualità che l'endpoint diventi non disponibile. Per un elenco dei *`region-code`* valori possibili, consulta la colonna Regione negli AWS Sign-In endpoint endpoint.

L'esempio seguente mostra il formato dell'URL di accesso con l'opzione obbligatoria. `region-code`

`https://`*`region-code`*`.signin.aws.amazon.com/saml`

Se è richiesta la crittografia SAML, l'URL di accesso deve includere l'identificatore univoco che AWS assegna al provider SAML, che puoi trovare nella pagina dei dettagli del provider di identità. Nell'esempio seguente, l'URL di accesso include l'identificatore univoco dell'IdP, per cui è necessario aggiungere /acs/ al percorso di accesso.

`https://`*`region-code`*`.signin.aws.amazon.com/saml/acs/`*`IdP-ID`*

Per la policy di autorizzazione nel ruolo, è necessario specificare le autorizzazioni come per qualsiasi ruolo. Ad esempio, se gli utenti della tua organizzazione sono autorizzati ad amministrare istanze Amazon Elastic Compute Cloud, devi consentire esplicitamente le azioni di Amazon EC2 nella politica delle autorizzazioni, come quelle nella policy gestita. AmazonEC2FullAccess

Per ulteriori informazioni sulle chiavi SAML che è possibile verificare in una policy, consulta Chiavi disponibili per SAML-based AWS STS federazione.

Identificazione univoca degli SAML-based utenti nella federazione

Quando si creano policy di accesso in IAM, spesso è utile poter specificare le autorizzazioni in base all'identità degli utenti. Ad esempio, per gli utenti che sono stati federati tramite SAML, un'applicazione potrebbe voler mantenere le informazioni in Amazon S3 utilizzando una struttura come questa:

```
amzn-s3-demo-bucket/app1/user1
amzn-s3-demo-bucket/app1/user2
amzn-s3-demo-bucket/app1/user3
```

Puoi creare il bucket (`amzn-s3-demo-bucket`) e la cartella (`app1`) tramite la console Amazon S3 o il AWS CLI, poiché si tratta di valori statici. Tuttavia, le cartelle specifiche dell'utente (*`user1`*,, *`user2user3`*, ecc.) devono essere create in fase di esecuzione utilizzando il codice, poiché il valore che identifica l'utente non è noto fino alla prima volta che l'utente accede tramite il processo di federazione.

Per scrivere policy che fanno riferimento ai dettagli specifici dell'utente come parte di un nome di risorsa, l'identità dell'utente deve essere disponibile nelle chiavi SAML che possono essere utilizzate nelle condizioni di policy. Le seguenti chiavi sono disponibili per la federazione basata su SAML 2.0 da utilizzare nelle policy IAM. È possibile utilizzare i valori restituiti dalle chiavi seguenti per creare identificativi utente univoci per risorse come le cartelle Amazon S3.

- `saml:namequalifier`. Un valore hash basato sulla concatenazione del valore della risposta `Issuer` (`saml:iss`) e una stringa con l'ID account `AWS` e il nome descrittivo (l'ultima parte dell'ARN) del provider SAML in IAM. La concatenazione dell'ID account e del nome descrittivo del provider SAML è disponibile per le policy IAM sotto forma di chiave `saml:doc`. L'ID dell'account e il nome del provider devono essere separati da una '/' come in "123456789012/provider_name». Per ulteriori informazioni, consulta la chiave `saml:doc` in Chiavi disponibili per SAML-based AWS STS federazione.

  La combinazione di `NameQualifier` e `Subject` può essere utilizzata per identificare in modo univoco un principale federato SAML. Il seguente pseudocodice mostra come viene calcolato questo valore. In questo pseudocodice + indica la concatenazione, `SHA1` rappresenta una funzione che produce un digest di messaggio utilizzando e `Base64` rappresenta una funzione che produce Base-64 una SHA-1 versione codificata dell'output hash.

```
Base64 ( SHA1 ( "https://example.com/saml" + "123456789012" + "/
MySAMLIdP" ) )
```

Per ulteriori informazioni sulle chiavi di policy disponibili per la federazione, vedere. SAML-based Chiavi disponibili per SAML-based AWS STS federazione

- `saml:sub` (Stringa). Questo è l'oggetto della richiesta, che include un valore che identifica in modo univoco un singolo utente in un'organizzazione (ad esempio, `_cbb88bf52c2510eabe00c1642d4643f41430fe25e3`).
- `saml:sub_type` (Stringa). Questa chiave può essere `persistent`, `transient` o l'URI `Format` completo degli elementi `Subject` e `NameID` utilizzati nell'asserzione SAML. Il valore `persistent` indica che il valore in `saml:sub` è lo stesso per un utente in tutte le sessioni. Se il valore è `transient`, l'utente dispone di un valore `saml:sub` diverso per ogni sessione. Per ulteriori informazioni sull'attributo `Format` dell'elemento `NameID`, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

L'esempio seguente mostra una policy di autorizzazione che utilizza le chiavi precedenti per concedere le autorizzazioni a una cartella specifica per utente in Amazon S3. La policy presuppone che gli oggetti Amazon S3 vengano identificati utilizzando un prefisso che include sia `saml:namequalifier` che `saml:sub`. Si noti che l'elemento `Condition` include un test per assicurarsi che `saml:sub_type` sia impostato su `persistent`. Se è impostato su `transient`, il valore `saml:sub` per l'utente può essere diverso per ogni sessione e la combinazione di valori non deve essere utilizzata per identificare le cartelle specifiche dell'utente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:PutObject",
      "s3:DeleteObject"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket-org-data/backup/${saml:namequalifier}/
${saml:sub}",
```

```
      "arn:aws:s3:::amzn-s3-demo-bucket-org-data/backup/${saml:namequalifier}/
${saml:sub}/*"
    ],
    "Condition": {"StringEquals": {"saml:sub_type": "persistent"}}
  }
}
```

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:PutObject",
      "s3:DeleteObject"
    ],
    "Resource": [
      "arn:aws-cn:s3:::amzn-s3-demo-bucket-org-data/backup/${saml:namequalifier}/
${saml:sub}",
      "arn:aws-cn:s3:::amzn-s3-demo-bucket-org-data/backup/${saml:namequalifier}/
${saml:sub}/*"
    ],
    "Condition": {"StringEquals": {"saml:sub_type": "persistent"}}
  }
}
```

Per ulteriori informazioni sulle asserzioni di mappatura dal provider di identità alle chiavi di policy, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

Creare un provider di identità SAML in IAM

Un provider di identità SAML 2.0 IAM è un'entità in IAM che descrive un servizio del provider di identità (IdP) esterno che supporta lo standard SAML 2.0 (Security Assertion Markup Language 2.0). Si utilizza un provider di identità IAM quando si desidera stabilire la fiducia tra un SAML-compatible IdP come Shibboleth o Active Directory Federation Services e AWS in modo che gli utenti possano accedere alle risorse. AWS I provider di identità SAML IAM vengono utilizzati come principali nelle policy di attendibilità IAM.

Per ulteriori informazioni su questo scenario, consulta Federazione SAML 2.0.

Puoi creare e gestire un provider di identità IAM tramite Console di gestione AWS o con AWS CLI Tools for Windows PowerShell o chiamate API. AWS

Una volta creato un provider SAML, dovrai creare uno o più ruoli IAM. Un ruolo è un'identità AWS che non dispone di credenziali proprie (come avviene per un utente). Tuttavia, in questo contesto, un ruolo viene assegnato in modo dinamico a un principale federato SAML autenticato dal tuo IdP. Il ruolo consente all'IdP di richiedere credenziali di sicurezza provvisorie per l'accesso a AWS. Le politiche assegnate al ruolo determinano le operazioni consentite agli utenti. AWS Per creare un ruolo per una federazione SAML consultare Creazione di un ruolo per un provider di identità di terze parti.

Infine, dopo aver creato il ruolo, completi il trust SAML configurando il tuo IdP con le informazioni AWS e i ruoli che desideri vengano utilizzati dai tuoi principali federati SAML. Questa operazione viene definita configurazione di una relazione di trust fra IdP e AWS. Per configurare una relazione di trust, consultare Configurare il provider di identità SAML 2.0 con una relazione di attendibilità della parte affidabile e aggiunta di attestazioni.

Argomenti

- Prerequisiti
- Creare e gestire un provider di identità SAML IAM (console)
- Gestire le chiavi di crittografia SAML
- Crea e gestisci un IAM SAML Identity Provider (AWS CLI)
- Creare e gestire un provider di identità SAML IAM (AWS API)
- Fasi successive

Prerequisiti

Prima di creare un provider di identità SAML, è necessario disporre delle seguenti informazioni dal proprio IdP.

- Scarica il documento di metadati SAML dal tuo IdP. Questo documento include il nome dell'approvatore, le informazioni sulla scadenza e le chiavi che possono essere utilizzate per convalidare la risposta di autenticazione SAML (asserzioni) che vengono ricevute dal provider di identità. Per generare il documento di metadati, utilizza il software di gestione delle identità fornito dall'IdP esterno.

> **Important**
>
> Questo file di metadati include il nome dell'approvatore, le informazioni sulla scadenza e le chiavi che possono essere utilizzate per convalidare la risposta di autenticazione SAML (asserzioni) ricevute dal provider di identità. Il file di metadati deve essere codificato in un UTF-8 formato senza un byte order mark (BOM). Per rimuovere la distinta base, puoi codificare il file UTF-8 utilizzando uno strumento di modifica del testo, come Notepad++. Il X.509 certificato incluso nel documento di metadati SAML deve utilizzare una dimensione della chiave di almeno 1024 bit. Inoltre, il X.509 certificato deve essere privo di estensioni ripetute. È possibile utilizzare le estensioni, ma possono essere visualizzate una sola volta nel certificato. Se il X.509 certificato non soddisfa nessuna delle due condizioni, la creazione dell'IdP non riesce e restituisce un errore «Impossibile analizzare i metadati». Come definito dal SAML V2.0 Metadata Interoperability Profile Version 1.0, IAM non valuta né interviene sulla scadenza dei certificati nei documenti di metadati SAML. X.509 Se sei preoccupato per i X.509 certificati scaduti, ti consigliamo di monitorare le date di scadenza dei certificati e di ruotare i certificati in base alle politiche di governance e sicurezza della tua organizzazione.

- Quando si sceglie di abilitare la crittografia SAML, è necessario generare un file di chiave privata utilizzando l'IdP e caricare questo file nella configurazione SAML di IAM in formato .pem. AWS STS necessita di questa chiave privata per decrittografare le risposte SAML che corrispondono alla chiave pubblica utilizzata dall'IdP. Sono supportati i seguenti algoritmi:
  - Algoritmi di crittografia
    - AES-128
    - AES-256
    - RSA-OAEP
  - Algoritmi di trasporto della chiave
    - AES-CBC
    - AES-GCM

  Consulta la documentazione del provider di identità per le istruzioni sulla generazione di una chiave privata.

> **Note**
>
> Il Centro identità IAM e Amazon Cognito non supportano le asserzioni SAML crittografate dei provider di identità IAM SAML. Puoi aggiungere indirettamente il supporto per le asserzioni SAML crittografate alla federazione dei pool di identità di Amazon Cognito con i pool di utenti di Amazon Cognito. I pool di utenti dispongono di una federazione SAML che è indipendente dalla federazione IAM SAML e supporta la firma e la crittografia SAML. Sebbene questa funzionalità non si estenda direttamente ai pool di identità, i pool di utenti possono riguardare i pool IdPs di identità. Per utilizzare la crittografia SAML con pool di identità, aggiungi un provider SAML con crittografia a un pool di utenti che è un IdP a un pool di identità.
> Il tuo provider SAML deve essere in grado di crittografare le asserzioni SAML con una chiave fornita dal tuo pool di utenti. I pool di utenti non accetteranno asserzioni crittografate con un certificato fornito da IAM.

Per istruzioni su come configurare molti dei dati disponibili con cui IdPs lavorare AWS, incluso come generare il documento di metadati SAML richiesto, consulta. Integra fornitori di soluzioni SAML di terze parti con AWS

Per assistenza sulla federazione SAML, consulta Risoluzione dei problemi della federazione SAML.

Creare e gestire un provider di identità SAML IAM (console)

Puoi utilizzare il Console di gestione AWS per creare, aggiornare ed eliminare i provider di identità SAML IAM. Per assistenza sulla federazione SAML, consulta Risoluzione dei problemi della federazione SAML.

Come creare un provider di identità SAML IAM (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Provider di identità, quindi seleziona Aggiungi provider.
3. Per Configura provider, scegli SAML.
4. Digitare un nome per il provider di identità.
5. In Documento metadati, fai clic su Scegli file e specifica il documento di metadati SAML scaricato in the section called "Prerequisiti".

> **Note**
>
> La data di scadenza valida visualizzata per il provider di identità è impostata da IAM al momento della creazione del provider. Non riflette `cacheDuration` gli attributi `validUntil` o presenti nel documento di metadati SAML. IAM non utilizza questi attributi di metadati per determinare la data di scadenza valida.
> IAM non valuta né interviene sulla scadenza del documento di metadati SAML o dei X.509 certificati nei documenti di metadati SAML durante l'autenticazione SAML. La validità dell'autenticazione SAML viene applicata a livello di asserzione, utilizzando l'attributo nell'asserzione e negli elementi. `NotOnOrAfter` `Conditions` `SubjectConfirmationData` Se sei preoccupato per i X.509 certificati scaduti, ti consigliamo di monitorare le date di scadenza dei certificati e di ruotare i certificati in base alle politiche di governance e sicurezza della tua organizzazione.

6. (Facoltativo) Per la Crittografia SAML, seleziona Scegli file e seleziona il file della chiave privata creato in the section called "Prerequisiti". Scegli Richiedi crittografia per accettare solo le richieste crittografate dal tuo IdP.
7. (Facoltativo) Per Aggiungi tag, puoi aggiungere coppie chiave-valore per aiutarti a identificare e organizzare i tuoi. IdPs È inoltre possibile utilizzare i tag per controllare l'accesso alle risorse AWS . Per ulteriori informazioni sull'applicazione di tag ai provider di identità SAML, vedere Aggiungere tag ai provider di identità SAML per IAM.

   Seleziona Aggiungi tag. Immetti i valori per ogni coppia chiave-valore del tag.
8. Controlla le informazioni inserite. Quando hai finito, scegli Aggiungi provider.
9. Assegna un ruolo IAM al tuo provider di identità. Questo ruolo fornisce alle identità degli utenti esterni gestite dal tuo provider di identità le autorizzazioni per accedere alle AWS risorse del tuo account. Per ulteriori informazioni sulla creazione di ruoli per la federazione delle identità, consulta Creazione di un ruolo per un provider di identità di terze parti.

> **Note**
>
> Gli IDP SAML utilizzati in una policy di attendibilità dei ruoli devono appartenere allo stesso account in cui si trova il ruolo.

Per eliminare un provider SAML (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel pannello di navigazione, scegli Identity providers (Provider di identità).
3. Seleziona la casella di controllo accanto al provider di identità che desideri eliminare.
4. Scegli Elimina. Viene visualizzata una nuova finestra.
5. Conferma che desideri eliminare il provider digitando la parola `delete` nel campo. Quindi, scegli Elimina.

Gestire le chiavi di crittografia SAML

Puoi configurare i provider SAML IAM per ricevere asserzioni crittografate nella risposta SAML dal tuo IdP esterno. Gli utenti possono assumere un ruolo nelle AWS asserzioni SAML crittografate chiamando. `sts:AssumeRoleWithSAML`

La crittografia SAML garantisce la sicurezza delle asserzioni quando vengono trasmesse tramite intermediari o terze parti. Inoltre, questa funzionalità consente di soddisfare il FedRAMP o qualsiasi requisito della policy di conformità interna che impone la crittografia delle asserzioni SAML.

Per configurare un provider di identità SAML IAM, consulta Creare un provider di identità SAML in IAM. Per assistenza sulla federazione SAML, consulta Risoluzione dei problemi della federazione SAML.

Ruotare la chiave di crittografia SAML

IAM utilizza la chiave privata che hai caricato sul provider SAML IAM per decrittografare le asserzioni SAML crittografate del tuo IdP. Puoi salvare fino a due file di chiave privata per ogni provider di identità, consentendoti di ruotare le chiavi private secondo necessità. Quando vengono salvati due file, ogni richiesta tenterà innanzitutto di decrittografare con la data riportata in Aggiunto il più recente, quindi IAM proverà a decrittografare la richiesta con la data riportata in Aggiunto il più vecchia.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione, scegli Provider di identità, quindi seleziona il tuo provider dall'elenco.
3. Scegli la scheda Crittografia SAML e seleziona Aggiungi nuova chiave.

4. Seleziona Scegli file e carica la chiave privata scaricata dal tuo IdP come file .pem, quindi seleziona Aggiungi chiave.
5. Nella sezione Chiavi private per la decrittografia SAML, seleziona il file di chiave privata scaduto e scegli Rimuovi. Ti consigliamo di rimuovere la chiave privata scaduta dopo aver aggiunto una nuova chiave privata per assicurarti che il primo tentativo di decrittografare l'asserzione abbia esito positivo.

Crea e gestisci un IAM SAML Identity Provider (AWS CLI)

Puoi utilizzarlo AWS CLI per creare, aggiornare ed eliminare i provider SAML. Per assistenza sulla federazione SAML, consulta Risoluzione dei problemi della federazione SAML.

Per creare un provider di identità IAM e caricare un documento di metadati (AWS CLI)

- Eseguire il comando: `aws iam create-saml-provider`

Per aggiornare un provider di identità SAML IAM (AWS CLI)

Puoi aggiornare il file di metadati, le impostazioni di crittografia SAML e ruotare i file di decrittografia a chiave privata per il tuo provider SAML IAM. Per ruotare le chiavi private, aggiungi la tua nuova chiave privata e rimuovi la vecchia chiave in una richiesta separata. Per ulteriori informazioni sulla rotazione delle chiavi private, consulta Gestire le chiavi di crittografia SAML.

- Eseguire il comando: `aws iam update-saml-provider`

Per etichettare un provider di identità IAM esistente (AWS CLI)

- Eseguire il comando: `aws iam tag-saml-provider`

Per elencare i tag per il provider di identità IAM esistente (AWS CLI)

- Eseguire il comando: `aws iam list-saml-provider-tags`

Per rimuovere i tag su un provider di identità IAM esistente (AWS CLI)

- Eseguire il comando: `aws iam untag-saml-provider`

Per eliminare un provider di identità SAML IAM (AWS CLI)

1. (Facoltativo) Per elencare le informazioni di tutti i provider (ad esempio l'ARN, la data di creazione e la scadenza), eseguire il seguente comando:

   - `aws iam list-saml-providers`
2. (Facoltativo) Per ottenere informazioni su un provider specifico (ad esempio l'ARN, la data di creazione, la scadenza, le impostazioni di crittografia e le informazioni sulla chiave privata), esegui il seguente comando:

   - `aws iam get-saml-provider`
3. Per eliminare un provider di identità IAM, esegui il comando:

   - `aws iam delete-saml-provider`

Creare e gestire un provider di identità SAML IAM (AWS API)

Puoi utilizzare l' AWS API per creare, aggiornare ed eliminare i provider SAML. Per assistenza sulla federazione SAML, consulta Risoluzione dei problemi della federazione SAML.

Per creare un provider di identità IAM e caricare un documento di metadati (AWS API)

- Richiamare l'operazione: `CreateSAMLProvider`

Per aggiornare un provider di identità SAML IAM (AWS API)

Puoi aggiornare il file di metadati, le impostazioni di crittografia SAML e ruotare i file di decrittografia a chiave privata per il tuo provider SAML IAM. Per ruotare le chiavi private, aggiungi la tua nuova chiave privata e rimuovi la vecchia chiave in una richiesta separata. Per ulteriori informazioni sulla rotazione delle chiavi private, consulta Gestire le chiavi di crittografia SAML.

- Richiamare l'operazione: `UpdateSAMLProvider`

Per etichettare un provider di identità IAM esistente (AWS API)

- Richiamare l'operazione: `TagSAMLProvider`

Per elencare i tag per un provider di identità IAM esistente (AWS API)

- Richiamare l'operazione: `ListSAMLProviderTags`

Per rimuovere i tag su un provider di identità IAM esistente (AWS API)

- Richiamare l'operazione: `UntagSAMLProvider`

Per eliminare un provider di identità IAM (AWS API)

1. (Facoltativo) Per elencare le informazioni relative a tutti IdPs, come l'ARN, la data di creazione e la scadenza, esegui la seguente operazione:
   - `ListSAMLProviders`
2. (Facoltativo) Per ottenere informazioni su un provider specifico (ad esempio l'ARN, la data di creazione, la data di scadenza, le impostazioni di crittografia e le informazioni sulla chiave privata), esegui il seguente comando:
   - `GetSAMLProvider`
3. Per eliminare un IdP, richiamare la seguente operazione:
   - `DeleteSAMLProvider`

Fasi successive

Dopo aver creato un provider di identità SAML, configura l'attendibilità della parte con il tuo IdP. È inoltre possibile utilizzare le attestazioni della risposta di autenticazione del tuo IdP nelle policy per controllare l'accesso a un ruolo.

- È necessario comunicare all'IdP la AWS qualità di fornitore di servizi. Questa relazione è nota come relazione di attendibilità tra il provider di identità e AWS. L'esatto processo per aggiungere una relazione di trust dipende dall'IdP utilizzato. Per informazioni dettagliate, vedi Configurare il provider di identità SAML 2.0 con una relazione di attendibilità della parte affidabile e aggiunta di attestazioni.
- Quando l'IdP invia la risposta contenente le attestazioni a AWS, molte delle richieste in entrata vengono associate a AWS chiavi di contesto. Puoi utilizzare queste chiavi di contesto nelle policy IAM utilizzando l'elemento Condition per controllare l'accesso a un ruolo. Per maggiori dettagli, consultare Configurare le asserzioni SAML per la risposta di autenticazione.

Configurare il provider di identità SAML 2.0 con una relazione di attendibilità della parte affidabile e aggiunta di attestazioni

Quando crei un provider di identità IAM e un ruolo per l'accesso SAML, indichi ad AWS il provider di identità (IdP) esterno e le operazioni che i rispettivi utenti sono autorizzati a effettuare. Il passaggio successivo consiste quindi nel comunicare all'IdP la AWS qualità di fornitore di servizi. Questa relazione è nota come relazione di trust tra il provider di identità e AWS. L'esatto processo per aggiungere una relazione di trust dipende dall'IdP utilizzato. Per ulteriori informazioni, consulta la documentazione relativa al tuo software di gestione delle identità.

Molti IdPs consentono di specificare un URL dal quale l'IdP può leggere un documento XML che contiene informazioni e certificati sui soggetti affidatari. Per AWS, usa l'URL dell'endpoint di accesso. L'esempio seguente mostra il formato dell'URL con l'opzione `region-code`

`https://region-code.signin.aws.amazon.com/static/saml-metadata.xml`

Se è richiesta la crittografia SAML, l'URL deve includere l'identificatore univoco AWS assegnato al tuo provider SAML, che puoi trovare nella pagina dei dettagli del provider di identità. L'esempio seguente mostra un URL di accesso regionale che include un identificatore univoco.

`https://region-code.signin.aws.amazon.com/static/saml/IdP-ID/saml-metadata.xml`

Per un elenco dei *region-code* valori possibili, consulta la colonna Regione negli endpoint degli endpoint. AWS Sign-In Per il AWS valore, puoi anche utilizzare l'endpoint non regionale. `https://signin.aws.amazon.com/saml`

Se non è possibile specificare direttamente un URL, scaricare il documento XML dall'URL precedente e importarlo nel software dell'IdP.

È inoltre necessario creare regole di reclamo appropriate nel proprio IdP che specifichino AWS come relying party. Quando l'IdP invia una risposta SAML all' AWS endpoint, include un'asserzione SAML che contiene una o più attestazioni. Un'attestazione consiste in un set di informazioni sull'utente e sui rispettivi gruppi. Una regola di attestazione mappa tali informazioni negli attributi SAML. Ciò ti consente di assicurarti che le risposte di autenticazione SAML del tuo IdP contengano gli attributi necessari AWS utilizzati nelle politiche IAM per verificare le autorizzazioni per i principali federati SAML. Per ulteriori informazioni, consulta i seguenti argomenti:

- Panoramica del ruolo per consentire l'accesso al tuo SAML-federated AWS risorse. Questo argomento illustra l'uso SAML-specific delle chiavi nelle policy IAM e come utilizzarle per limitare le autorizzazioni per i committenti federati SAML.
- Configurare le asserzioni SAML per la risposta di autenticazione. In questo argomento viene descritto come configurare le attestazioni SAML che includono informazioni sull'utente. Le attestazioni sono raggruppate in un'asserzione SAML e incluse nella risposta SAML inviata ad AWS. È necessario assicurarsi che le informazioni necessarie alle AWS policy siano incluse nell'asserzione SAML in una forma che possa essere riconosciuta e utilizzata. AWS
- Integra fornitori di soluzioni SAML di terze parti con AWS. Questo argomento fornisce collegamenti alla documentazione fornita da organizzazioni di terze parti su come integrare soluzioni di identità con AWS.

> **Note**
>
> Per migliorare la resilienza della federazione, ti consigliamo di configurare l'IdP e la federazione AWS per supportare più endpoint di accesso SAML. Per informazioni dettagliate, consulta l'articolo del AWS Security Blog How to use regional SAML endpoint for failover.

Integra fornitori di soluzioni SAML di terze parti con AWS

> **Note**
>
> Ti consigliamo di richiedere agli utenti umani di utilizzare credenziali temporanee per l'accesso. AWS Hai preso in considerazione l'utilizzo Centro identità AWS IAM? Puoi utilizzare IAM Identity Center per gestire centralmente l'accesso a più account Account AWS e fornire agli MFA-protected utenti l'accesso Single Sign-on a tutti gli account assegnati da un'unica posizione. Con IAM Identity Center puoi creare e gestire le identità degli utenti in IAM Identity Center o connetterti facilmente al tuo attuale gestore dell'identità digitale (IdP) compatibile con SAML 2.0. Per ulteriori informazioni, consulta Cos'è IAM Identity Center? nella Guida per l'utente di Centro identità AWS IAM .

I seguenti collegamenti consentono di configurare soluzioni di provider di identità (IdP) SAML 2.0 di terze parti in modo che funzionino con la federazione. AWS Rivolgiti al tuo provider di identità per

determinare se supporta la crittografia con token SAML. Per i requisiti di crittografia SAML, consulta Gestire le chiavi di crittografia SAML.

> **Tip**
>
> AWS I tecnici dell'assistenza possono assistere i clienti che dispongono di piani di supporto aziendali ed aziendali con alcune attività di integrazione che coinvolgono software di terze parti. Per un elenco aggiornato delle piattaforme e delle applicazioni supportate, consulta Qual è il software per terze parti supportato? in Domande frequenti sul supporto di AWS .

Soluzione	Ulteriori informazioni
Auth0	Integrazione con Amazon Web Services: questa pagina del sito Web di documentazione Auth0 contiene collegamenti a risorse che descrivono come configurare il Single Sign-on (SSO) con Console di gestione AWS e include un esempio. JavaScript È possibile configurare Auth0 per passare i tag di sessione. Per ulteriori informazioni, consulta Auth0 Announces Partnership with for IAM Session Tags. AWS
Microsoft Entra	Tutorial: integrazione SSO di Microsoft Entra con AWS Single-Account Access: questo tutorial sul sito Web Microsoft descrive come configurare Microsoft Entra (precedentemente noto come Azure AD) come provider di identità (IdP) utilizzando la federazione SAML.
Centrify	Configura Centrify e usa SAML per SSO AWS: questa pagina del sito Web di Centrify spiega come configurare Centrify per utilizzare SAML per SSO. AWS
CyberArk	Configura CyberArk per fornire l'accesso ad Amazon Web Services (AWS) agli utenti che accedono tramite SAML Single Sign-on (SSO) dal Portale utenti. CyberArk
ForgeRock	La ForgeRock piattaforma Identity si integra con. AWSÈ possibile configurare il passaggio ForgeRock dei tag ???

Soluzione	Ulteriori informazioni
	di sessione. Per ulteriori informazioni, consulta Attribute Based Access Control for Amazon Web Services.
Google Workspace	Applicazione cloud Amazon Web Services: questo articolo sul sito di assistenza per amministratori di Google Workspace descrive come configurare Google Workspace come IdP SAML 2.0 con AWS fornitore di servizi.
IBM	È possibile configurare IBM per passare i tag di sessione. Per ulteriori informazioni, consulta IBM Cloud Identity IDaaS, uno dei primi a supportare i tag di sessione. AWS
JumpCloud	Concessione dell'accesso tramite ruoli IAM per Single Sign On (SSO) con Amazon AWS: questo articolo sul JumpCloud sito Web descrive come configurare e abilitare l'SSO in base ai ruoli IAM per. AWS
Matrix42	MyWorkspace Guida introduttiva: questa guida descrive come integrare i servizi di AWS identità con Matrix42. MyWorkspace
Microsoft Active Directory Federation Services (AD FS)	Note sul campo: Integrazione di Active Directory Federation Service con Centro identità AWS IAM — Questo post sul blog di AWS architettura spiega il flusso di autenticazione tra AD FS e Centro identità AWS IAM (IAM Identity Center). IAM Identity Center supporta la federazione delle identità con SAML 2.0, consentendo l'integrazione con le soluzioni AD FS. Gli utenti possono accedere al portale di IAM Identity Center con le proprie credenziali aziendali, riducendo il sovraccarico amministrativo dovuto al mantenimento di credenziali separate. Inoltre, puoi configurare AD FS per passare i tag di sessione. Per ulteriori informazioni, consulta Utilizzo del controllo accessi basato sugli attributi con AD FS per semplificare la gestione delle autorizzazioni IAM.

Soluzione	Ulteriori informazioni
miniOrange	SSO per AWS: questa pagina del sito web miniOrange descrive come stabilire un accesso sicuro AWS per le aziende e il pieno controllo sull'accesso alle applicazioni. AWS
Okta	Integrazione dell'interfaccia a riga di comando di Amazon Web Services tramite Okta: da questa pagina del sito del supporto di Okta è possibile ottenere informazioni su come configurare Okta per l'utilizzo con AWS. È possibile configurare Okta per passare i tag di sessione. Per ulteriori informazioni, consulta Okta and AWS Partner to Simplify Access Tramite Session Tag.
Okta	AWS Federazione degli account: questa sezione del sito web di Okta descrive come configurare e abilitare IAM Identity Center per. AWS
OneLogin	Nella OneLogin Knowledgebase, **`SAML AWS`** cerca un elenco di articoli che spieghino come configurare la funzionalità di IAM Identity Center tra OneLogin e AWS per scenari a ruolo singolo e multiruolo. Puoi configurare il passaggio dei tag di sessione. OneLogin ??? Per ulteriori informazioni, vedere OneLogin e Session Tags: Attribute-Based Access Control for AWS Resources.
Identità Ping	PingFederate AWS Connettore: visualizza i dettagli sul PingFederate AWS Connector, un modello di connessio ne rapida per configurare facilmente una connessione Single Sign-on (SSO) e di provisioning. Leggi la documenta zione e scarica la versione più recente di PingFederate AWS Connector per le integrazioni con. AWSÈ possibile configurare Ping Identity per passare i tag di sessione. Per ulteriori informazioni, vedi Annuncio del supporto di Ping Identity per il controllo degli Attribute-Based accessi in. AWS

Soluzione	Ulteriori informazioni
RadiantLogic	Partner tecnologici Radiant Logic: il RadiantOne Federated Identity Service di Radiant Logic si integra AWS per fornire un hub di identità per SSO. SAML-based
RSA	Amazon Web Services - Guida all'implementazione di RSA Ready fornisce linee guida per l'integrazione di AWS e RSA. Per ulteriori informazioni sulla configurazione di SAML, consulta Amazon Web Services - Configurazione SSO della mia pagina SAML - Guida all'implementazione di RSA Ready.
Salesforce.com	Come configurare l'SSO da Salesforce a AWS: questo articolo pratico sul sito per Salesforce.com sviluppatori descrive come configurare un provider di identità (IdP) in Salesforce e configurarlo come fornitore di servizi. AWS
SecureAuth	AWS - SecureAuth SAML SSO: questo articolo del SecureAuth sito Web descrive come configurare l'integra zione SAML con per un'appliance. AWS SecureAuth
Shibboleth	Come utilizzare Shibboleth for SSO per Console di gestione AWS: questo articolo del AWS Security Blog fornisce un tutorial dettagliato su come configurare Shibboleth e configurarlo come provider di identità per. AWSÈ possibile configurare Shibboleth per passare i tag di sessione.

Per maggiori dettagli, consulta la pagina IAM Partners sul sito web. https://aws.amazon.com/iam/partners/ AWS

Configurare le asserzioni SAML per la risposta di autenticazione

Dopo aver verificato l'identità di un utente nella tua organizzazione, il provider di identità esterno (IdP) invia una risposta di autenticazione all'URL dell'endpoint di accesso. AWS Questa risposta è una richiesta POST che include un token SAML che aderisce al binding POST HTTP per lo standard SAML 2.0 e che contiene i seguenti elementi o attestazioni. Puoi configurare queste attestazioni

nel tuo IdP. SAML-compatible Per ulteriori informazioni, consulta la documentazione del provider di identità per istruzioni su come inserire queste attestazioni.

Quando l'IdP invia la risposta contenente le attestazioni a AWS, molte delle attestazioni in entrata vengono associate a AWS chiavi di contesto. Queste chiavi di contesto possono essere controllate nelle policy IAM utilizzando l'elemento `Condition`. L'elenco delle mappature disponibili è incluso nella sezione Mappatura degli attributi SAML su AWS chiavi di contesto della politica di fiducia.

`Oggetto` e NameID

La risposta deve includere esattamente un elemento `SubjectConfirmation` con un elemento `SubjectConfirmationData` che include sia l'attributo `NotOnOrAfter` sia l'attributo `Recipient`. L'attributo Destinatario deve includere un valore che corrisponda all'URL dell'endpoint di AWS accesso. Il tuo IdP può utilizzare il termine `ACS` o `Target` per fare riferimento a questo attributo. `Recipient`

Se è richiesta la crittografia SAML, l'URL di accesso deve includere l'identificatore univoco assegnato da AWS al provider SAML e consultabile nella pagina dei dettagli del provider di identità. L'esempio seguente mostra il formato dell'URL di accesso con l'opzione obbligatorio `region-code`

`https://`*`region-code`*`.signin.aws.amazon.com/saml`

Nell'esempio seguente, l'URL di accesso include un identificatore univoco, per cui è necessario aggiungere /acs/ al percorso di accesso.

`https://`*`region-code`*`.signin.aws.amazon.com/saml/acs/`*`IdP-ID`*

Per un elenco dei *`region-code`* valori possibili, vedi la colonna Regione negli AWS Sign-In endpoints endpoints Per il AWS valore, puoi anche utilizzare l'endpoint di accesso globale. `https://signin.aws.amazon.com/saml`

Gli elementi `NameID` possono avere il valore persistente, transitorio o oppure possono essere costituiti dall'URI formato completo, come fornito dalla soluzione IdP. Un valore permanente indica che il valore in `NameID` è lo stesso per un utente da una sessione all'altra. Se il valore è transitorio, l'utente dispone di un valore `NameID` diverso per ogni sessione. Le interazioni Single Sign-on supportano i seguenti tipi di identificatori:

- `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`
- `urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
- `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`

- `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`
- `urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName`
- `urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName`
- `urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos`
- `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`

Di seguito viene riportato un estratto di esempio. Sostituire con i propri valori i valori contrassegnati.

```
<Subject>
  <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">_cbb88bf52c2510eabe00c1642d4643f41430fe25e3</NameID>
  <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <SubjectConfirmationData NotOnOrAfter="2013-11-05T02:06:42.876Z"
 Recipient="https://region-code.signin.aws.amazon.com/saml/SAMLSP4SHN3UIS2D558H46"/>
  </SubjectConfirmation>
</Subject>
```

> **Important**
>
> Al contrario, la chiave di contesto `saml:aud` proviene dall'attributo recipient SAML perché è l'equivalente SAML del campo del destinatario OIDC, ad esempio, `accounts.google.com:aud`.

Attributo SAML **`PrincipalTag`**

(Facoltativo) Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/PrincipalTag:{TagKey}`. Questo elemento consente di passare attributi come tag di sessione nell'asserzione SAML. Per ulteriori informazioni sui tag di sessione, consultare Passa i tag della sessione in AWS STS.

Per passare gli attributi come tag di sessione, includi l'elemento `AttributeValue` che specifica il valore del tag. Ad esempio, per passare la coppia chiave-valore del tag `Project` = `Marketing` e `CostCenter` = `12345`, utilizza il seguente attributo. Includi un elemento `Attribute` separato per ogni tag.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:Project">
```

```
  <AttributeValue>Marketing</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:CostCenter">
  <AttributeValue>12345</AttributeValue>
</Attribute>
```

Per impostare i tag sopra elencati come transitivi, includere un altro elemento `Attribute` con l'attributo `Name` impostato a `https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys`. Questo è un attributo con più valori opzionale che imposta i tag di sessione come transitivi. I tag transitivi persistono quando utilizzi la sessione SAML per assumere un altro ruolo in AWS. Questo è noto come concatenazione del ruolo. Ad esempio, per impostare entrambi i tag `CostCenter` e `Principal` come transitivi, utilizza il seguente attributo per specificare le chiavi.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys">
  <AttributeValue>Project</AttributeValue>
  <AttributeValue>CostCenter</AttributeValue>
</Attribute>
```

Attributo SAML del ruolo

Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/Role` Questo elemento contiene uno o più elementi `AttributeValue` che elencano il ruolo e il provider di identità IAM a cui l'utente è mappato dall'IdP. Il ruolo IAM e il provider di identità IAM sono specificati come una coppia di ARN delimitata da virgole nello stesso formato dei parametri `RoleArn` e `PrincipalArn` a cui vengono passati. AssumeRoleWithSAML Questo elemento deve contenere almeno una coppia ruolo-provider (elemento `AttributeValue`) e può contenere più coppie. Se l'elemento contiene più coppie, all'utente viene chiesto di scegliere il ruolo da assumere quando utilizza WebSSO per accedere alla Console di gestione AWS.

> **Important**
>
> Il valore dell'attributo `Name` nel tag `Attribute` è sensibile alla distinzione tra maiuscolo/minuscolo. Il valore deve essere impostato esattamente su `https://aws.amazon.com/SAML/Attributes/Role`.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/Role">
```

```
  <AttributeValue>arn:aws:iam::account-number:role/role-name1,arn:aws:iam::account-
number:saml-provider/provider-name</AttributeValue>
  <AttributeValue>arn:aws:iam::account-number:role/role-name2,arn:aws:iam::account-
number:saml-provider/provider-name</AttributeValue>
  <AttributeValue>arn:aws:iam::account-number:role/role-name3,arn:aws:iam::account-
number:saml-provider/provider-name</AttributeValue>
</Attribute>
```

Attributo SAML **RoleSessionName**

Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/RoleSessionName` Questo elemento contiene un elemento `AttributeValue` che fornisce un identificatore per le credenziali temporanee emesse quando viene assunto il ruolo. È possibile utilizzare questa opzione per associare le credenziali temporanee all'utente che utilizza l'applicazione. Questo elemento viene utilizzato per visualizzare le informazioni sull'utente in. Console di gestione AWS Il valore nell'elemento `AttributeValue` deve contenere tra 2 e 64 caratteri, può contenere solo caratteri alfanumerici, caratteri di sottolineatura e i seguenti caratteri: . , + = @ - (trattino). Non può contenere spazi. Il valore è in genere un ID utente (`john`) o un indirizzo e-mail (`johndoe@example.com`). Non deve essere un valore che include uno spazio, ad esempio il nome di visualizzazione di un utente (`John Doe`).

> **Important**
>
> Il valore dell'attributo `Name` nel tag `Attribute` è sensibile alla distinzione tra maiuscolo/minuscolo. Il valore deve essere impostato esattamente su `https://aws.amazon.com/SAML/Attributes/RoleSessionName`.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/RoleSessionName">
  <AttributeValue>user-id-name</AttributeValue>
</Attribute>
```

Attributo SAML **SessionDuration**

(Facoltativo) Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/SessionDuration`. Questo elemento contiene un `AttributeValue` elemento che specifica per quanto tempo l'utente può accedere Console di gestione AWS prima di dover richiedere nuove credenziali temporanee. Il valore è un numero intero che rappresenta il numero di secondi per la sessione. Il valore può variare da 900 secondi (15 minuti)

a 43.200 secondi (12 ore). Se questo attributo non è presente, la credenziale dura per un'ora (il valore predefinito del parametro `DurationSeconds` dell'API `AssumeRoleWithSAML`).

Per utilizzare questo attributo, è necessario configurare il provider SAML in modo da fornire l'accesso Single Sign-on all'endpoint web di accesso Console di gestione AWS tramite console all'indirizzo. `https://region-code.signin.aws.amazon.com/saml` Per un elenco dei *`region-code`* valori possibili, consulta la colonna Regione negli endpoint degli endpoint. AWS Sign-In Facoltativamente, puoi utilizzare il seguente URL:. `https://signin.aws.amazon.com/static/saml` Si noti che questo attributo estende le sessioni solo alla Console di gestione AWS. Non può estendere la durata di altre credenziali. Tuttavia, se è presente in una chiamata API `AssumeRoleWithSAML`, può essere utilizzato per abbreviare la durata della sessione. La durata predefinita delle credenziali restituite dalla chiamata è di 60 minuti.

Si noti inoltre che, se viene definito anche un attributo `SessionNotOnOrAfter`, il valore inferiore dei due attributi, `SessionDuration` o `SessionNotOnOrAfter`, stabilisce la durata massima della sessione della console.

Quando si abilitano le sessioni della console con una durata estesa, aumenta il rischio di compromissione delle credenziali. Per mitigare questo rischio, è possibile disabilitare immediatamente le sessioni della console attiva per tutti i ruoli, scegliendo Revoca sessioni nella pagina Riepilogo ruolo della console IAM. Per ulteriori informazioni, consulta Revocare le credenziali di sicurezza temporanee per i ruoli IAM.

> **Important**
>
> Il valore dell'attributo `Name` nel tag `Attribute` è sensibile alla distinzione tra maiuscolo/minuscolo. Il valore deve essere impostato esattamente su `https://aws.amazon.com/SAML/Attributes/SessionDuration`.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/SessionDuration">
  <AttributeValue>1800</AttributeValue>
</Attribute>
```

Attributo SAML **`SourceIdentity`**

(Facoltativo) Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/SourceIdentity`. Questo elemento contiene un elemento `AttributeValue` che fornisce un identificatore per la persona o l'applicazione che utilizza un ruolo

IAM. Il valore dell'identità di origine persiste quando si utilizza la sessione SAML per assumere un altro ruolo AWS noto come concatenamento di ruoli. Il valore per l'identità di origine è presente nella richiesta per ogni operazione eseguita durante la sessione del ruolo. Il valore impostato non può essere modificato durante la sessione del ruolo. Gli amministratori possono quindi utilizzare AWS CloudTrail i log per monitorare e verificare le informazioni sull'identità di origine per determinare chi ha eseguito azioni con ruoli condivisi.

Il valore dell'`AttributeValue`elemento deve avere una lunghezza compresa tra 2 e 256 caratteri, può contenere solo caratteri alfanumerici, trattini bassi e i seguenti caratteri:., + = @ - (trattino). Non può contenere spazi. Il valore è in genere un attributo associato all'utente, ad esempio un ID utente (`john`) o un indirizzo e-mail (`johndoe@example.com`). Non deve essere un valore che include uno spazio, ad esempio il nome di visualizzazione di un utente (`John Doe`). Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

> **Important**
>
> Se l'asserzione SAML è configurata per utilizzare l'attributo `SourceIdentity`, allora la policy di attendibilità del ruolo deve includere anche l'operazione `sts:SetSourceIdentity` altrimenti l'operazione di assunzione del ruolo avrà esito negativo. Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

Per inviare un attributo dell'identità di origine, includi l'elemento `AttributeValue` che specifica il valore dell'identità di origine. Ad esempio, per inviare `Diego` dell'identità di origine, utilizza il seguente attributo.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/SourceIdentity">
  <AttributeValue>Diego</AttributeValue>
```

Mappatura degli attributi SAML su AWS chiavi di contesto della politica di fiducia

Le tabelle in questa sezione elencano gli attributi SAML comunemente usati e il modo in cui sono mappati alle chiavi di contesto delle condizioni delle policy in AWS. Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi con i valori che sono inclusi nelle asserzioni incluse in una richiesta di accesso SAML.

> **Important**
>
> Queste chiavi sono disponibili solo nelle policy di affidabilità IAM (policy che determinano chi può assumere un ruolo) e non sono applicabili alle policy di autorizzazione.

Nella tabella degli attributi eduPerson e eduOrg, i valori vengono digitati come stringhe o come elenchi di stringhe. Per i valori di stringa, puoi testare questi valori nelle policy di attendibilità IAM utilizzando le condizioni `StringEquals` o `StringLike`. Per i valori che contengono un elenco di stringhe, è possibile utilizzare gli `ForAnyValue`operatori del set di policy`ForAllValues` e per un test dei valori delle policy di attendibilità.

> **Note**
>
> È necessario includere solo un'attestazione per chiave di AWS contesto. Se ne include più di una, verrà mappata una sola attestazione.

La tabella riportata di seguito mostra gli attributi eduPerson ed eduOrg.

Attributo eduPerson o eduOrg (chiave **`Name`**)	Si riferisce a questa chiave di AWS contesto (**`FriendlyName`** chiave)	Tipo
`urn:oid:1.3.6.1.4.1.5923.1.1.1.1`	`eduPersonAffiliation`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.2`	`eduPersonNickname`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.3`	`eduPersonOrgDN`	Stringa
`urn:oid:1.3.6.1.4.1.5923.1.1.1.4`	`eduPersonOrgUnitDN`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.5`	`eduPersonPrimaryAffiliation`	Stringa

Attributo eduPerson o eduOrg (chiave **`Name`**)	Si riferisce a questa chiave di AWS contesto (**`FriendlyName`** chiave)	Tipo
`urn:oid:1.3.6.1.4.1.5923.1.1.1.6`	`eduPersonPrincipalName`	Stringa
`urn:oid:1.3.6.1.4.1.5923.1.1.1.7`	`eduPersonEntitlement`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.8`	`eduPersonPrimaryOrgUnitDN`	Stringa
`urn:oid:1.3.6.1.4.1.5923.1.1.1.9`	`eduPersonScopedAffiliation`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.10`	`eduPersonTargetedID`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.11`	`eduPersonAssurance`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.2.1.2`	`eduOrgHomePageURI`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.2.1.3`	`eduOrgIdentityAuthNPolicyURI`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.2.1.4`	`eduOrgLegalName`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.2.1.5`	`eduOrgSuperiorURI`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.2.1.6`	`eduOrgWhitePagesURI`	Elenco di stringhe
`urn:oid:2.5.4.3`	`cn`	Elenco di stringhe

La tabella riportata di seguito mostra gli attributi di Active Directory.

Attributo AD	Si basa su questa chiave di AWS contesto	Tipo
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name`	`name`	Stringa
`http://schemas.xmlsoap.org/claims/CommonName`	`commonName`	Stringa
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname`	`givenName`	Stringa
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname`	`surname`	Stringa
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`	`mail`	Stringa
`http://schemas.microsoft.com/ws/2008/06/identity/claims/primarygroupsid`	`uid`	Stringa

La tabella seguente mostra X.500 gli attributi.

X.500 attributo	È mappato a questa chiave di AWS contesto	Tipo
`2.5.4.3`	`commonName`	Stringa
`2.5.4.4`	`surname`	Stringa
`2.4.5.42`	`givenName`	Stringa
`2.5.4.45`	`x500UniqueIdentifier`	Stringa

X.500 attributo	È mappato a questa chiave di AWS contesto	Tipo
`0.9.2342.19200300100.1.1`	`uid`	Stringa
`0.9.2342.19200300100.1.3`	`mail`	Stringa
`0.9.2342.19200300.100.1.45`	`organizationStatus`	Stringa

Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS

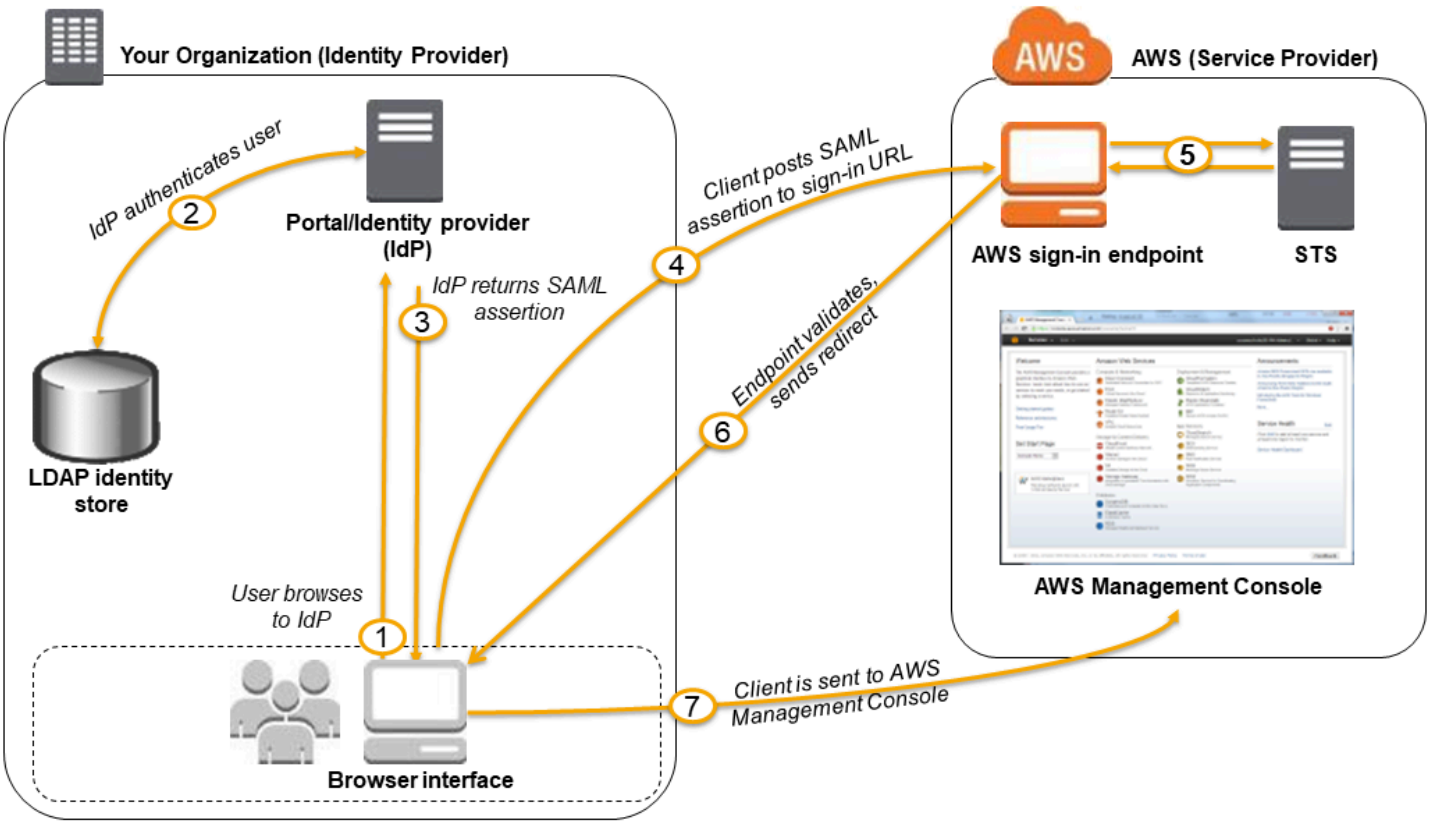
Puoi utilizzare un ruolo per configurare il tuo provider di identità (IdP) conforme a SAML 2.0 e consentire AWS ai principali federati SAML di accedere a. Console di gestione AWS Il ruolo concede all'utente le autorizzazioni per eseguire attività nella console. Se desideri offrire ai responsabili federati SAML altri modi di accesso, consulta uno di questi argomenti: AWS

- AWS CLI: Passa a un ruolo IAM (AWS CLI)
- Strumenti per Windows: PowerShell Passa a un ruolo IAM (Strumenti per Windows PowerShell)
- AWS API: Passa a un ruolo IAM (AWS API)

Panoramica di

Il diagramma seguente illustra il flusso per il SAML-enabled Single Sign-On.

> **Note**
>
> Questo uso specifico di SAML è diverso da quello più generale illustrato in precedenza Federazione SAML 2.0 perché questo flusso di lavoro viene aperto per conto dell'utente. Console di gestione AWS In questo caso occorre utilizzare l'endpoint di accesso ad AWS anziché richiamare direttamente l'API `AssumeRoleWithSAML`. L'endpoint richiama l'API per l'utente e restituisce un'URL che reindirizza automaticamente il browser dell'utente alla Console di gestione AWS.



Il diagramma illustra i passaggi seguenti:

1. L'utente accede al portale dell'organizzazione e seleziona l'opzione che consente di accedere alla Console di gestione AWS. Nella tua organizzazione, il portale è in genere una funzione del tuo IdP che gestisce lo scambio di fiducia tra l'organizzazione e. AWS Ad esempio, in Active Directory Federation Services, l'URL del portale è: `https://ADFSServiceName/adfs/ls/IdpInitiatedSignOn.aspx`
2. Il portale verifica l'identità dell'utente nell'organizzazione.
3. Il portale genera una risposta di autenticazione SAML che include asserzioni che identificano l'utente e includono gli attributi dell'utente. È anche possibile configurare il provider di identità per includere un attributo di asserzione SAML chiamato `SessionDuration` che specifica la durata della validità della sessione della console. È anche possibile configurare il provider di identità per passare gli attributi come tag di sessione. Il portale invia questa risposta al browser del client.
4. Il browser del client viene reindirizzato all'endpoint AWS Single Sign-On e pubblica l'asserzione SAML.
5. L'endpoint richiede le credenziali di sicurezza provvisorie per conto dell'utente e crea una URL di accesso alla console che utilizza tali credenziali.

6. AWS invia l'URL di accesso al client come reindirizzamento.
7. Il browser del client è reindirizzato alla Console di gestione AWS. Se la risposta di autenticazione SAML include attributi mappati a più ruoli IAM, all'utente viene chiesto di selezionare il ruolo per l'accesso alla console.

Dal punto di vista dell'utente, il processo avviene in modo trasparente: l'utente inizia dal portale interno dell'organizzazione e finisce al Console di gestione AWS, senza mai dover fornire alcuna credenziale. AWS

Consulta le seguenti sezioni per una panoramica della configurazione di questo comportamento insieme ai collegamenti alla procedura dettagliata.

Configura la tua rete come provider SAML per AWS

All'interno della rete dell'organizzazione, configurate il vostro archivio di identità (come Windows Active Directory) per lavorare con un SAML-based IdP come Windows Active Directory Federation Services, Shibboleth, ecc. Utilizzando il provider di identità, si genera un documento di metadati che descrive l'organizzazione come un IdP e include le chiavi di autenticazione. Inoltre, configuri il portale della tua organizzazione per indirizzare le richieste degli utenti all'endpoint SAML Console di gestione AWS per l'autenticazione utilizzando le AWS asserzioni SAML. Il modo in cui è possibile configurare il provider di identità per la produzione del file metadata.xml dipende dal provider di identità. Per ulteriori informazioni, consulta la documentazione del provider di identità, oppure consulta Integra fornitori di soluzioni SAML di terze parti con AWS per i collegamenti alla documentazione Web per molti dei fornitori di SAML supportati.

Creazione di un provider SAML in IAM

Successivamente, accedi Console di gestione AWS e vai alla console IAM. Qui si crea un nuovo provider SAML, ovvero un'entità in IAM che contiene informazioni sul provider di identità dell'organizzazione. Come parte di questo processo, è possibile caricare il documento di metadati prodotto dal software IdP nella propria organizzazione nella sezione precedente. Per informazioni dettagliate, vedi Creare un provider di identità SAML in IAM.

Configura le autorizzazioni in AWS per i presidi federati SAML

La fase successiva consiste nel creare un ruolo IAM che stabilisca una relazione di attendibilità tra IAM e il provider di identità dell'organizzazione. Questo ruolo deve identificare il tuo provider di identità come un principale (entità attendibile) ai fini della federazione. Il ruolo definisce anche le operazioni consentite agli utenti autenticati dall'IdP dell'organizzazione. AWSÈ possibile utilizzare

la console IAM per creare questo ruolo. Quando si crea la policy di attendibilità che indica chi può assumere il ruolo, specifica il provider SAML creato in precedenza in IAM. È inoltre possibile specificare uno o più attributi SAML a cui un utente deve corrispondere per poter assumere quel ruolo. Ad esempio, è possibile specificare che solo gli utenti il cui valore SAML `eduPersonOrgDN` è `ExampleOrg` sono autorizzati ad accedere. La procedura guidata relativa al ruolo aggiunge automaticamente una condizione per testare l'attributo `saml:aud` per assicurarsi che il ruolo venga assunto solo per l'accesso alla Console di gestione AWS.

Se è richiesta la crittografia SAML, l'URL di accesso deve includere l'identificatore univoco assegnato da AWS al provider SAML e consultabile nella pagina dei dettagli del provider di identità. La policy di affidabilità potrebbe apparire come segue:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:saml-provider/
ExampleOrgSSOProvider"
            },
            "Action": "sts:AssumeRoleWithSAML",
            "Condition": {
                "StringEquals": {
                    "saml:edupersonorgdn": "ExampleOrg",
                    "saml:aud": "https://region-code.signin.aws.amazon.com/saml/
acs/SAMLSP4SHN3UIS2D558H46"
                }
            }
        }
    ]
}
```

> **Note**
>
> Gli IDP SAML utilizzati in una policy di attendibilità dei ruoli devono appartenere allo stesso account in cui si trova il ruolo.

Ti consigliamo di utilizzare gli endpoint regionali per l'attributo `saml:aud` all'indirizzo `https://region-code.signin.aws.amazon.com/static/saml-metadata.xml`.

Per la policy di autorizzazione nel ruolo, è necessario specificare le autorizzazioni come per qualsiasi utente, gruppo o ruolo. Ad esempio, se gli utenti dell'organizzazione sono autorizzati ad amministrare le istanze Amazon EC2, consenti esplicitamente le operazioni Amazon EC2 nella policy di autorizzazione. A tale scopo, puoi assegnare una policy gestita, ad esempio la policy di accesso completo di Amazon EC2.

Per ulteriori informazioni sulla creazione di un ruolo per un provider di identità SAML, consulta Creare un ruolo per una federazione SAML 2.0 (console).

Fine della configurazione e creazione di asserzioni SAML

Informa il tuo IdP SAML AWS che è il tuo fornitore di servizi installando `saml-metadata.xml` il file disponibile `https://region-code.signin.aws.amazon.com/static/saml-metadata.xml` in o. `https://signin.aws.amazon.com/static/saml-metadata.xml` Se è richiesta la crittografia SAML, il file si trova in. `https://region-code.signin.aws.amazon.com/static/saml/SAMLSP4SHN3UIS2D558H46/saml-metadata.xml`

Il modo in cui installare tale file dipende dal provider di identità. Alcuni provider danno la possibilità di digitare l'URL, dopodiché il provider di identità ottiene e installa il file per conto dell'utente. Altri richiedono di scaricare il file dall'URL e quindi fornirlo come file locale. Per ulteriori informazioni, consulta la documentazione del provider di identità, oppure consulta Integra fornitori di soluzioni SAML di terze parti con AWS per i collegamenti alla documentazione Web per molti dei fornitori di SAML supportati.

È anche possibile configurare le informazioni che si desidera che il provider di identità passi come attributi SAML per AWS come parte della risposta di autenticazione. La maggior parte di queste informazioni viene visualizzata AWS come chiavi di contesto delle condizioni che puoi valutare nelle tue politiche. Queste chiavi di condizione garantiscono che solo agli utenti autorizzati nei giusti contesti vengono concesse le autorizzazioni per accedere alle risorse AWS . È possibile specificare le finestre di tempo che limitano l'utilizzo della console. È inoltre possibile specificare il tempo massimo (fino a 12 ore) durante il quale gli utenti possono accedere alla console prima di dover aggiornare le proprie credenziali. Per informazioni dettagliate, vedi Configurare le asserzioni SAML per la risposta di autenticazione.

Visualizzare una risposta SAML nel browser

Le procedure seguenti descrivono come visualizzare nel browser la risposta SAML del proprio provider di servizi durante la risoluzione di un problema relativo a SAML 2.0.

Per tutti i browser, passare alla pagina in cui è possibile riprodurre il problema. Quindi seguire i passaggi per il browser appropriato:

Argomenti

- Google Chrome
- Mozilla Firefox
- Apple Safari
- Cosa fare con la risposta Base64-encoded SAML

Google Chrome

Per visualizzare una risposta SAML in Chrome

Questi passaggi sono stati testati utilizzando la versione 106.0.5249.103 (versione ufficiale) (arm64) di Google Chrome. Se si utilizza una versione diversa, potrebbe essere necessario modificare i passaggi di conseguenza.

1. Premere F12 per avviare la console Strumenti per sviluppatori.
2. Selezionare la scheda Netowrk (Rete), quindi selezionare Preserve log (Conserva registro) nella parte superiore sinistra della finestra Developer Tools (Strumenti per sviluppatori).
3. Riprodurre il problema.
4. (Facoltativo) Se la colonna Metodo non è visibile nel riquadro del registro di Developer Tools Network, apri il menu contestuale (clic con il pulsante destro del mouse) per qualsiasi etichetta di colonna e scegli Metodo per aggiungere la colonna.
5. Cercare un SAML Post (Post SAML) nel pannello di registrazione Developer Tools (Strumenti per sviluppatori) Network (Rete). Selezionare la riga e quindi visualizzare la scheda Payload (Carico utile) nella parte superiore. Cercare l'elemento SAMLResponse che contiene la richiesta codificata. Il valore associato è la Base64-encoded risposta.

Mozilla Firefox

Per visualizzare una risposta SAML in Firefox

Questa procedura è stata testata con la versione 105.0.3 (64 bit) di Mozilla Firefox. Se si utilizza una versione diversa, potrebbe essere necessario modificare i passaggi di conseguenza.

1. Premere F12 per avviare la console Strumenti per sviluppatori Web.
2. Selezionare la scheda Network (Rete).
3. In alto a destra nella finestra Web Developer Tools (Strumenti per sviluppatori Web), fare clic sull'icona delle opzioni (il piccolo ingranaggio). Selezionare Persist logs (Preserva registri).
4. Riprodurre il problema.
5. (Facoltativo) Se la colonna Metodo non è visibile nel riquadro del registro di Web Developer Tools Network, apri il menu contestuale (clic con il pulsante destro del mouse) per qualsiasi etichetta di colonna e scegli Metodo per aggiungere la colonna.
6. Cercare un MESSAGGIO SAML nella tabella. Selezionare la riga e quindi visualizzare la scheda Request (Richiesta) e trovare l'elemento SAMLResponse. Il valore associato è la Base64-encoded risposta.

Apple Safari

Per visualizzare una risposta Safari

Questi passaggi sono stati testati utilizzando la versione 16.0 (17614.1.25.9.10, 17614) di Apple Safari. Se si utilizza una versione diversa, potrebbe essere necessario modificare i passaggi di conseguenza.

1. Abilitare Web Inspector in Safari. Aprire la finestra delle preferenze selezionare la scheda delle impostazioni avanzate e quindi selezionare l'opzione per mostrare il menu Sviluppo nella barra dei menu.
2. Ora è possibile aprire Web Inspector. Scegliere Develop (Sviluppo) nella barra dei menu, quindi selezionare Show Web Inspector (Mostra Web Inspector).
3. Selezionare la scheda Network (Rete).
4. Nella parte superiore sinistra della finestra Web Inspector, fare clic sull'icona delle opzioni (il piccolo cerchio con tre linee orizzontali). Selezionare Preserve Logs (Conserva registri).

5. (Facoltativo) Se la colonna Metodo non è visibile nel riquadro del registro di Web Inspector Network, apri il menu contestuale (clic con il pulsante destro del mouse) per qualsiasi etichetta di colonna e scegli Metodo per aggiungere la colonna.
6. Riprodurre il problema.
7. Cercare un MESSAGGIO SAML nella tabella. Selezionare la riga e quindi visualizzare la scheda Headers (Intestazioni).
8. Cercare l'elemento SAMLResponse che contiene la richiesta codificata. Scorrere per trovare l'elemento `Request Data` con nome `SAMLResponse`. Il valore associato è la risposta. Base64-encoded

Cosa fare con la risposta Base64-encoded SAML

Dopo aver trovato l'elemento di risposta Base64-encoded SAML nel browser, copialo e utilizza il tuo strumento di Base-64 decodifica preferito per estrarre la risposta con tag XML.

> **Suggerimento per la sicurezza**
>
> Poiché i dati di risposta SAML visualizzati potrebbero contenere dati di sicurezza sensibili, si consiglia di non utilizzare un decodificatore base64. Utilizzare invece uno strumento installato sul computer locale che non invia i dati SAML sulla rete.

Built-in opzione per sistemi Windows ()PowerShell:

```
PS C:
\> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("base64encodedtext
```

Built-in opzione per sistemi macOS e Linux:

```
$ echo "base64encodedtext" | base64 --decode
```

Rivedere i valori nel file decodificato

Rivedi i valori nel file di risposta SAML decodificato.

- Verifica che il valore dell'attributo saml:NameID corrisponda al nome utente dell'utente autenticato.
- Controlla il valore di ARN e SAML provider fanno distinzione tra maiuscole e minuscole e l'ARN deve corrispondere alla risorsa nel tuo account. https://aws.amazon.com/SAML/Attributes/Role.

- Rivedi il valore per https://aws.amazon.com/SAML/Attributes/RoleSessionName. Il valore deve corrispondere al valore nella regola di richiesta. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_create_saml_relying-party.html
- Se configuri il valore dell'attributo per un indirizzo e-mail o un nome di account, assicurati che i valori siano corretti. I valori devono corrispondere all'indirizzo e-mail o al nome dell'account dell'utente autenticato.

Verificare la presenza di errori e confermare la configurazione

Controlla se i valori contengono errori e conferma che le seguenti configurazioni siano corrette.

- Le regole di rivendicazione soddisfano gli elementi richiesti e tutti gli ARN sono corretti. Per ulteriori informazioni, consulta Configurare il provider di identità SAML 2.0 con una relazione di attendibilità della parte affidabile e aggiunta di attestazioni.
- Hai caricato il file di metadati più recente dal tuo IdP AWS nel tuo provider SAML. Per ulteriori informazioni, consulta Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS.
- La policy di attendibilità del ruolo IAM sia stata configurata correttamente. Per ulteriori informazioni, consulta Metodi per assumere un ruolo.

Federando AWS Identità verso servizi esterni

La federazione delle identità in uscita IAM consente ai AWS carichi di lavoro di accedere in modo sicuro ai servizi esterni senza archiviare credenziali a lungo termine. I tuoi AWS carichi di lavoro possono richiedere token Web JSON (JWT) di breve durata da AWS Security Token Service () chiamando l'API.AWS STS GetWebIdentityToken Questi token sono firmati crittograficamente, verificabili pubblicamente e contengono una serie completa di attestazioni che attestano l'identità del carico di lavoro ai servizi esterni. AWS Puoi utilizzare questi token con un'ampia gamma di provider cloud di terze parti, piattaforme SaaS e applicazioni ospitate autonomamente. I servizi esterni verificano l'autenticità del token utilizzando le chiavi AWS di verifica pubblicate su endpoint noti e utilizzano le informazioni contenute nei token per prendere decisioni di autenticazione e autorizzazione.

La federazione delle identità in uscita elimina la necessità di archiviare credenziali a lungo termine come chiavi API o password nel codice dell'applicazione o nelle variabili di ambiente, migliorando il livello di sicurezza. Puoi controllare l'accesso alla generazione dei token e applicare le proprietà

dei token come algoritmi di firma, destinatari consentiti e durata utilizzando le politiche IAM. Tutte le richieste di token vengono registrate AWS , fornendo audit trail completi per il monitoraggio della sicurezza e la reportistica sulla conformità. Puoi anche personalizzare i token con tag che appaiono come attestazioni personalizzate, consentendo ai servizi esterni di implementare un controllo degli accessi granulare e basato sugli attributi.

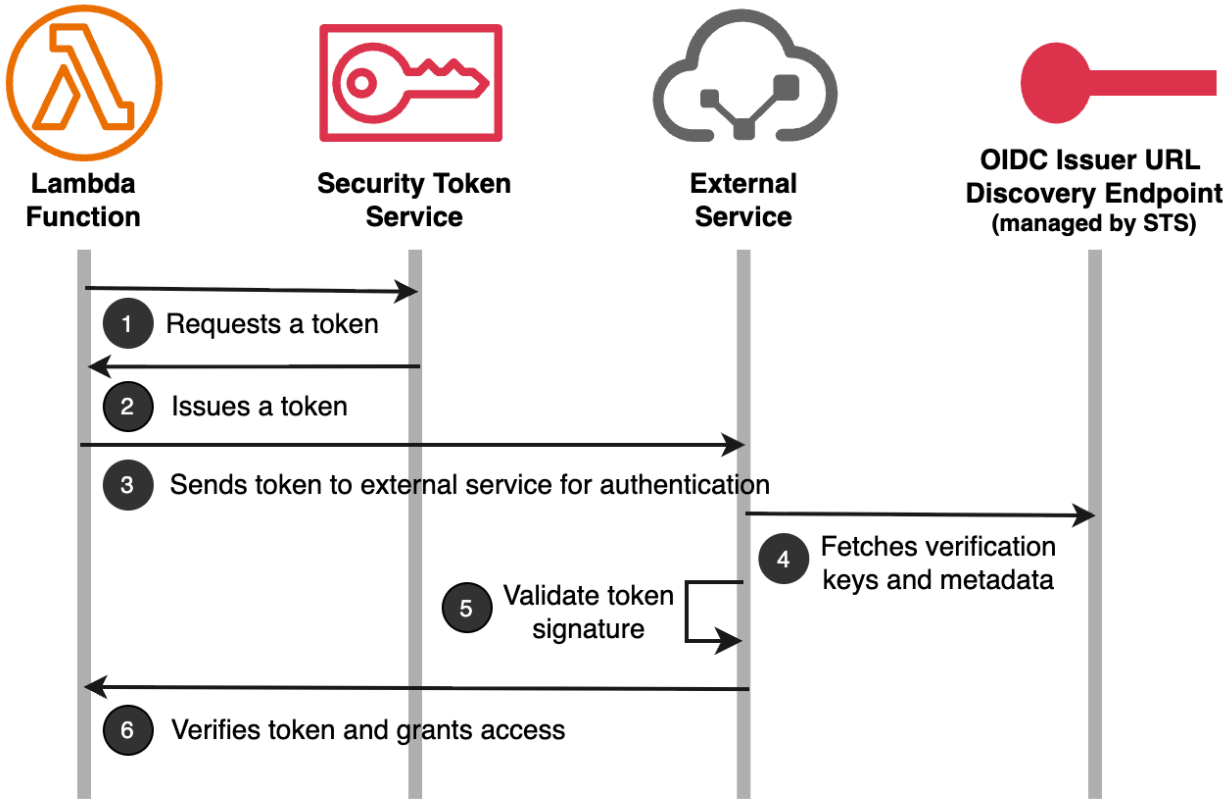
Per istruzioni sulla configurazione della federazione in uscita, consulta. Guida introduttiva alla federazione delle identità in uscita Per ulteriori informazioni sui contenuti dei token, vedereComprendere le dichiarazioni relative ai token. Per istruzioni sul controllo dei carichi di lavoro che possono richiedere i token, consulta. Controllo degli accessi con policy IAM

Casi di utilizzo comune

Utilizzando la federazione delle identità in uscita, i carichi di AWS lavoro possono in modo sicuro:

- Accedi a risorse e servizi in provider cloud esterni. Ad esempio, una funzione Lambda che elabora i dati può scrivere i risultati nel servizio di archiviazione di un provider di cloud esterno o interrogare il suo database.
- Effettua l'integrazione con fornitori esterni di software-as-a-service (SaaS) per analisi, elaborazione dati, monitoraggio ecc. Ad esempio, le tue funzioni Lambda possono inviare metriche a piattaforme di osservabilità.
- Effettua l'autenticazione con le tue applicazioni ospitate su provider cloud esterni o data center locali AWS, abilitando architetture ibride e multi-cloud sicure. Ad esempio, i tuoi AWS carichi di lavoro possono interagire con le applicazioni containerizzate in esecuzione nel tuo cluster Kubernetes locale.

Come funziona



1. La funzione Lambda chiama l'GetWebIdentityTokenAPI per richiedere un JSON Web Token (JWT) da Security Token Service (). AWS AWS STS
2. AWS STS convalida la richiesta e restituisce un JWT firmato alla funzione Lambda.
3. La funzione Lambda invia il JWT al servizio esterno.
4. Il servizio esterno estrae l'URL dell'emittente dal token, verifica che corrisponda a un emittente affidabile noto e recupera le chiavi di verifica e i metadati dall'endpoint AWS di rilevamento OIDC.
5. Il servizio esterno utilizza le chiavi di verifica per verificare la firma del token e convalida affermazioni quali ora di scadenza, oggetto e pubblico.
6. Dopo l'esito positivo della convalida, il servizio esterno concede l'accesso alla funzione Lambda.

Guida introduttiva alla federazione delle identità in uscita

Questa guida mostra come abilitare la federazione delle identità in uscita per il tuo AWS account e ottenere il tuo primo JSON Web Token (JWT) (utilizzando l'API). GetWebIdentityToken Attiverai la

funzionalità, stabilirai una relazione di fiducia con un servizio esterno, configurerai le autorizzazioni IAM e richiederai un token utilizzando la AWS CLI o l' AWS SDK for Python (Boto3).

Prerequisiti

Prima di iniziare, assicurati di avere:

- Ultima versione della AWS CLI o Python 3.8 (o successiva) e Boto3 installati (per esempi SDK) AWS
- Un account di servizio esterno in cui è possibile configurare le relazioni di fiducia (ad esempio un provider cloud esterno, un provider SaaS o un'applicazione di test)

> **Note**
>
> - L'`GetWebIdentityToken`API non è disponibile sull'endpoint STS Global.
> - I JSON Web Tokens (JWT) generati dall'`GetWebIdentityToken`API non possono essere utilizzati per la federazione di OpenID Connect (OIDC) in (tramite l'API). AWS AssumeRoleWithWebIdentity

Abilita la federazione delle identità in uscita per il tuo account

Devi abilitare la federazione delle identità in uscita prima di poter richiedere i token. È possibile abilitare la funzionalità utilizzando la console di AWS gestione o a livello di codice utilizzando l'API. EnableOutboundWebIdentityFederation

Utilizzo di AWS CLI

```
aws iam enable-outbound-web-identity-federation
```

Utilizzo AWS SDK for Python

```
import boto3

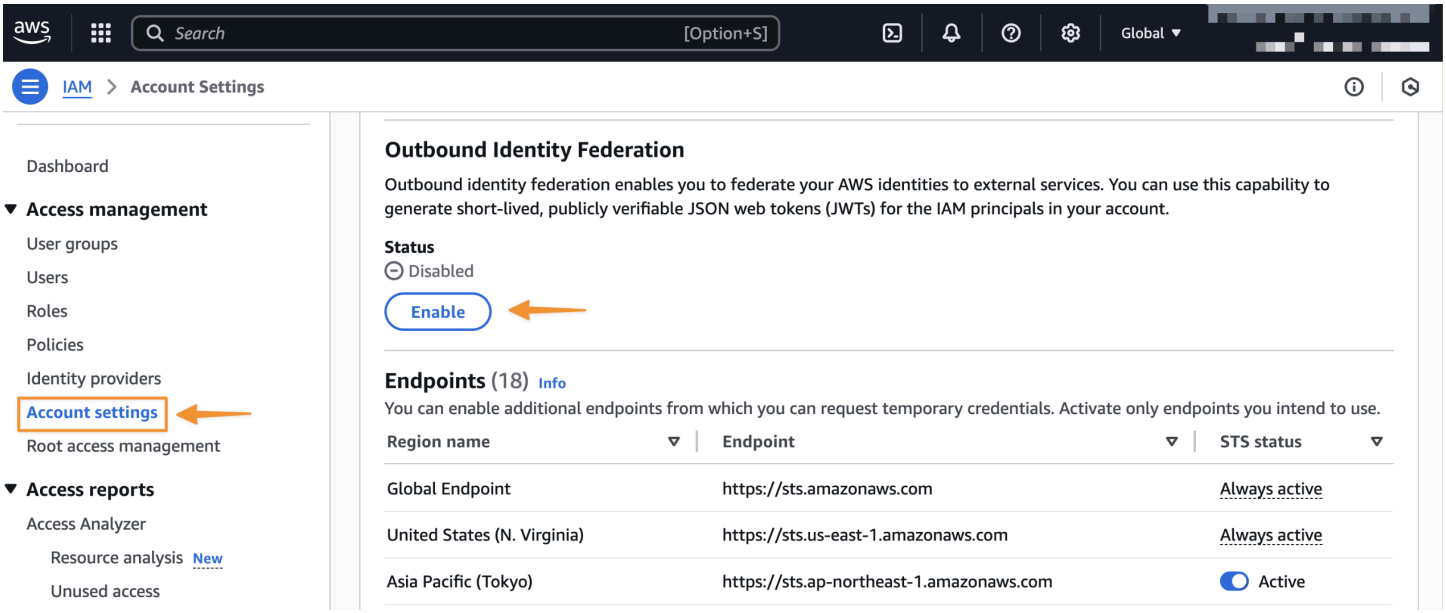
# Create IAM client
iam_client = boto3.client('iam')

# Enable outbound identity federation
response = iam_client.enable_outbound_web_identity_federation()
```

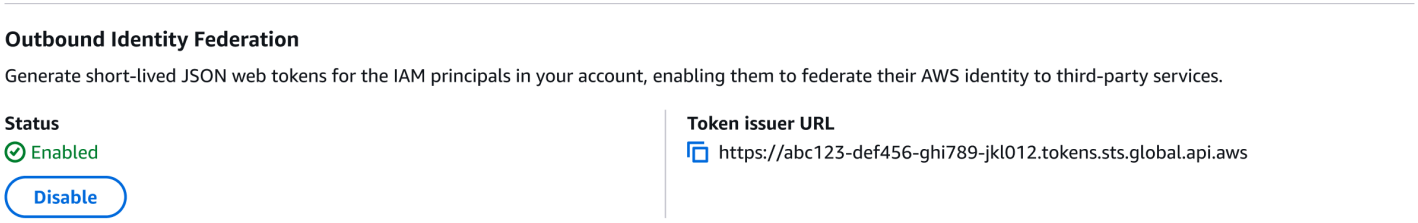
```
print(f"Feature enabled. Issuer URL: {response['IssuerUrl']}")
print(f"Status: {response['Status']}")
```

Utilizzo di AWS Console

Accedi a IAM e seleziona Impostazioni dell'account nella sezione Gestione degli accessi del menu di navigazione a sinistra



Una volta abilitata la funzione, annota l'URL dell'emittente specifico del tuo account. Utilizzerai questo URL per configurare le relazioni di fiducia nei servizi esterni. Puoi anche recuperare questo URL dell'emittente secondo necessità utilizzando l'API. GetOutboundWebIdentityFederationInfo



Stabilisci una relazione di fiducia nel servizio esterno

Configura il servizio esterno in modo che consideri attendibile e accetti i token emessi dal tuo AWS account. I passaggi specifici variano in base al servizio, ma generalmente comportano:

- Registrazione dell'URL AWS dell'emittente dell'account come provider di identità affidabile
- Configurazione delle affermazioni da convalidare (destinatario, modelli tematici)
- Mappatura delle dichiarazioni dei token alle autorizzazioni nel servizio esterno

Per istruzioni di configurazione dettagliate, consulta la documentazione del servizio esterno.

Configurazione delle autorizzazioni IAM

Crea una policy IAM che conceda il permesso di chiamare l'GetWebIdentityTokenAPI e allegare la policy a un ruolo IAM che deve generare token.

Questa policy di esempio garantisce l'accesso alla generazione di token con restrizioni specifiche. Consente di richiedere i token solo per "https://api.example.com" come pubblico e impone una durata massima dei token di 5 minuti (300 secondi). Fai riferimento a the section called "Chiavi di condizione IAM" per un elenco delle chiavi di condizione che puoi utilizzare per applicare le proprietà dei token.

Policy IAM di esempio

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:GetWebIdentityToken",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "sts:IdentityTokenAudience": "https://api.example.com"
                },
                "NumericLessThanEquals": {
                    "sts:DurationSeconds": 300
                }
            }
        }
    ]
}
```

Richiedi il tuo primo JSON Web Token (JWT)

Puoi richiedere un token Web JSON utilizzando l'API. GetWebIdentityToken È possibile specificare i seguenti parametri quando si chiama l'API:

- Destinatari (obbligatorio): il destinatario previsto del token. Questo valore compila l'attestazione «aud» nel JWT. I servizi esterni convalidano questa dichiarazione per assicurarsi che il token sia destinato a loro.

- SigningAlgorithm (obbligatorio): l'algoritmo crittografico utilizzato per firmare il token. I valori validi sono ES384 e RS256. Utilizzate ES384 per sicurezza e prestazioni ottimali o RS256 per una maggiore compatibilità con i sistemi che non supportano l'ECDSA.
- DurationSeconds (opzionale): la durata del token in secondi. I valori validi vanno da 60 a 3600. Il valore predefinito è 300 (5 minuti). Consigliamo una durata più breve dei token per una maggiore sicurezza.
- Tag (opzionale): un elenco di coppie chiave-valore da includere come attestazioni personalizzate nel token. I servizi esterni possono utilizzare queste attestazioni per un'autorizzazione dettagliata.

L'API restituisce i seguenti campi:

- IdentityToken: il JWT firmato come stringa codificata in base64url. Includi questo token nelle richieste a servizi esterni.
- Scadenza: il timestamp UTC relativo alla scadenza del token.

Utilizzo di AWS CLI

```
aws sts get-web-identity-token \
    --audience "https://api.example.com" \
    --signing-algorithm ES384 \
    --duration-seconds 300 \
    --tags Key=team,Value=data-engineering \
           Key=environment,Value=production \
           Key=cost-center,Value=analytics
```

Utilizzo AWS SDK for Python

```
import boto3

sts_client = boto3.client('sts')

response = sts_client.get_web_identity_token(
    Audience=['https://api.example.com'],
    DurationSeconds=300,
    SigningAlgorithm='RS256',
    Tags=[
        {'Key': 'team', 'Value': 'data-engineering'},
        {'Key': 'environment', 'Value': 'production'},
        {'Key': 'cost-center', 'Value': 'analytics'}
```

```
    ]
)

token = response['WebIdentityToken']
```

Puoi anche decodificare il JWT per controllarne il contenuto utilizzando librerie JWT standard come PyJWT, per Python, Nimbus JOSE+JWT Python-jose per Java o debugger come jwt.io. the section called "Comprendere le dichiarazioni relative ai token"Fai riferimento a per ulteriori informazioni sui claim inclusi nel token.

Usa il token con un servizio esterno

Dopo aver ricevuto il token, includilo nelle richieste al servizio esterno. Il metodo varia in base al servizio, ma la maggior parte dei servizi accetta i token nell'intestazione di autorizzazione. Il servizio esterno dovrebbe implementare una logica di convalida dei token che recupera le chiavi JWKS dal noto endpoint dell'emittente, verifica la firma del token e convalida le attestazioni essenziali prima di concedere l'accesso ai carichi di lavoro. AWS

Recupera le chiavi di verifica e i metadati dagli endpoint OpenID Connect (OIDC)

L'URL univoco dell'emittente del tuo AWS account ospita gli endpoint di rilevamento OpenID Connect (OIDC) che contengono le chiavi di verifica e i metadati necessari per la verifica dei token.

L'URL dell'endpoint di rilevamento OIDC contiene metadati utilizzati da alcuni provider per verificare i token. È disponibile all'indirizzo:

```
{issuer_url}/.well-known/openid-configuration
```

L'endpoint JWKS (JSON Web Key Set) contiene le chiavi utilizzate per verificare le firme dei token. È disponibile all'indirizzo:

```
{issuer_url}/.well-known/jwks.json
```

Recupera JWKS usando curl

```
curl https://{issuer_url}/.well-known/jwks.json
```

Risposta:

```
{
```

```
  "keys": [
    {
      "kty": "EC",
      "use": "sig",
      "kid": "key-id-1",
      "alg": "ES384",
      "crv": "P-384",
      "x": "base64-encoded-x-coordinate",
      "y": "base64-encoded-y-coordinate"
    },
    {
      "kty": "RSA",
      "use": "sig",
      "kid": "key-id-2",
      "n": "base64-encoded-modulus",
      "e": "AQAB"
    }
  ]
}
```

Utilizzo AWS SDK for Python

```
import requests

# Fetch Openid Configuration
open_id_config_response = requests.get("https://{issuer_url}/.well-known/openid-
configuration")
open_id_config = open_id_config_response.json()

# Fetch JWKS
jwks_response = requests.get("https://{issuer_url}/.well-known/jwks.json")
jwks = jwks_response.json()
```

Consigliamo di memorizzare nella cache queste chiavi per evitare di recuperarle per ogni verifica del token.

Convalide essenziali dei reclami

- Oggetto (sub): verifica che l'oggetto della rivendicazione contenga il modello ARN principale IAM previsto.
- Scadenza (exp): assicurati che il token non sia scaduto. Le librerie JWT in genere lo gestiscono automaticamente.

- Pubblico (aud): verifica che il pubblico corrisponda al valore previsto. Ciò impedisce che i token destinati ad altri servizi vengano utilizzati con i tuoi.
- Emittente (iss): verifica che l'emittente corrisponda agli AWS account di cui ti fidi. Mantieni un elenco di URL degli emittenti attendibili.

Ove possibile, dovresti convalidare ulteriori attestazioni AWS specifiche per implementare un controllo granulare degli accessi nel tuo servizio esterno. Ad esempio, convalida l'attestazione org_id per limitare l'accesso ai principali IAM nella tua AWS organizzazione, controlla principal_tags per applicare il controllo degli accessi basato sugli attributi (ad esempio consentendo solo ambienti di produzione o team specifici) o verifica le affermazioni sul contesto della sessione come lambda_source_function_arn o ec2_instance_source_vpc per limitare l'accesso in base alla risorsa di calcolo. Fai riferimento a Understanding token claims per un elenco completo dei claim inclusi nel token.

Comprendere le dichiarazioni relative ai token

Quando si chiama l'GetWebIdentityTokenAPI, AWS Security Token Service restituisce un JSON Web Token (JWT) firmato che contiene una serie di attestazioni che rappresentano l'identità del principale IAM. Questi token sono conformi alla RFC 7519. https://datatracker.ietf.org/doc/html/rfc7519 La comprensione della struttura e del contenuto di questi token consente di implementare flussi di autenticazione sicuri, configurare le convalide appropriate delle richieste in servizi esterni e utilizzare efficacemente le attestazioni personalizzate per un controllo granulare degli accessi.

Il JWT include dichiarazioni standard OpenID Connect (OIDC) come subject («sub»), audience («aud»), issuer («iss») per facilitare l'interoperabilità tra diversi servizi esterni. AWS STS compila il token con dichiarazioni AWS specifiche sull'identità (come l'ID AWS account e i tag Principal) e dichiarazioni relative al contesto della sessione (come gli ARN delle istanze EC2), se applicabili. Puoi anche aggiungere attestazioni personalizzate al token passandole come tag di richiesta all'API. GetWebIdentityToken Le attestazioni AWS specifiche dell'identità, le attestazioni relative al contesto di sessione e le attestazioni personalizzate sono annidate nello spazio dei nomi "https://sts.amazonaws.com/" del token.

Fai riferimento al token di esempio riportato di seguito per un elenco delle attestazioni incluse nel token. Tieni presente che tutte queste affermazioni potrebbero non essere presenti in un token contemporaneamente.

```
{
```

```
  "iss": "https://abc123-def456-ghi789-jkl012.tokens.sts.global.api.aws",
  "aud": "https://api.example.com",
  "sub": "arn:aws:iam::123456789012:role/DataProcessingRole",
  "iat": 1700000000,
  "exp": 1700000900,
  "jti": "xyz123-def456-ghi789-jkl012",
  "https://sts.amazonaws.com/": {
    "aws_account": "123456789012",
    "source_region": "us-east-1",
    "org_id": "o-abc1234567",
    "ou_path": "o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/",
    "principal_tags": {
      "environment": "production",
      "team": "data-engineering",
      "cost-center": "engineering"
    },
    "lambda_source_function_arn": "arn:aws:lambda:us-
east-1:123456789012:function:process-data",
    "request_tags": {
        "job-id": "job-2024-001",
        "priority": "high",
        "data-classification": "sensitive"
    }
  }
}
```

Reclami standard

I claim OIDC standard presenti nei token facilitano l'interoperabilità con un'ampia gamma di servizi esterni. Queste affermazioni possono essere convalidate utilizzando la maggior parte delle librerie JWT.

Richiedi	Nome	Description	Valore di esempio
è	Emittente	L'URL dell'emittente specifico dell'account. I servizi esterni convalidano questa dichiarazione per assicurarsi che corrisponda all'emittente attendibile.	https://abc123-def456-ghi789-jkl012.tokens.sts.global.api.aws

Richiedi	Nome	Description	Valore di esempio
aud	Destinatari	Il destinatario previsto del token specificato nella GetWebIdentityToken richiesta.	https://api.example.com
sub	Subject	L'ARN del principale IAM che ha richiesto il token.	arn:aws:iam: :123456789012:role/DataProcessingRole
iat	Emesso in	NumericDate valore che identifica l'ora in cui il JWT è stato emesso.	1700000000
exp	Scadenza	NumericDate valore che identifica l'ora di scadenza entro la quale il JWT NON DEVE essere accettato per l'elaborazione.	1700000900
kit	JWT È	Identificatore univoco per questa istanza di token.	xyz123-def456-ghi789-jkl012

Reclami personalizzati

Oltre alle dichiarazioni OIDC standard, AWS STS aggiunge affermazioni sull'identità e sul contesto della sessione, se applicabili. Puoi anche aggiungere i tuoi claim al token passandoli come tag di richiesta. Le attestazioni personalizzate sono annidate nel https://sts.amazonaws.com/ namespace.

AWS rivendicazioni di identità

Queste dichiarazioni forniscono informazioni dettagliate sull' AWS account, sulla struttura organizzativa e sul responsabile IAM.

Richiedi	Description	Maps to Condition Key	Valore di esempio
aws_account	L'ID del tuo account AWS	era: Principal Account	123456789012

Richiedi	Description	Maps to Condition Key	Valore di esempio
source_region	La AWS regione in cui è stato richiesto il token	era: Requested Region	us-east-1
org_id	L'ID della tua AWS organizzazione (se il tuo account fa parte di un'organizzazione)	aws: PrincipalOrgID	o-abc1234567
ou_path	Il percorso della tua unità organizzativa (se applicabile)	era: PrincipalOrgPaths	o- a1b2c3d4e5/r -ab12/ou -ab12- -ab12/ 11111111/ou
tag_principali	Tag allegati alla sessione con ruolo principale o presunto di IAM. Quando viene richiesto un token in cui il responsabile IAM richiedente ha sia tag principali che tag di sessione, i tag di sessione saranno presenti nel JWT.	era:/PrincipalTag<tag-key>	{"environment»: «produzione», «team»: «ingegneria dei dati», «cost-center» :"ingegneria "}

Dichiarazioni relative al contesto della sessione

Queste dichiarazioni forniscono informazioni sull'ambiente di calcolo e sulla sessione da cui ha avuto origine la richiesta del token. AWS STS include automaticamente queste affermazioni, se applicabili, in base al contesto della sessione del principale richiedente.

Richiedi	Description	Si associa a Condition Key	Valore di esempio
original_session_exp	Quando scadranno le credenziali della sessione di ruolo originale (per i ruoli assunti)	N/A	2024-01-15 T10:00:00Z

Richiedi	Description	Si associa a Condition Key	Valore di esempio
fornitore_federato	Il nome del provider di identità per le sessioni federate	era: Federated Provider	arn:aws:iam: :111122223333:oidc-provider/your_oidc_provider
identity_store_user_id	ID utente IAM Identity Center	archivio di identità: UserId	utente abc123def456
identity_store_arn	ARN dell'archivio di identità di Identity Center	archivio di identità: IdentityStoreArn	arn:aws:identitystore: :123456789012: -abc1234567 identitystore/d
ec2_source_instance_arn	ARN dell'istanza EC2 richiedente	ec2: SourceInstanceArn	arn:aws:ec2:us-east- 1:123456789012: -abc123def456 instance/i
ec2_instance_source_vpc	ID VPC in cui sono state fornite le credenziali del ruolo EC2	era: Ec2InstanceSourceVpc	vpc-abc123def456
ec2_instance_source_private_ipv4	Indirizzo IPv4 privato dell'istanza EC2	era: Ec2InstanceSourcePrivateIPv4	10.0.1.25
ec2_role_delivery	Versione del servizio di metadati dell'istanza	ec2: RoleDelivery	2
identità_origine	Identità di origine impostata dal principale	era: SourceIdentity	utente amministratore

Richiedi	Description	Si associa a Condition Key	Valore di esempio
lambda_source_function_arn	ARN della funzione Lambda chiamante	lambda: SourceFunctionArn	arn:aws:lambda:us-east- 1:123456789012:funzione:my-function
glue_credential_issuing_service	AWS Identificatore del servizio Glue per i lavori Glue	colla: CredentialIssuingService	colla.amazonaws.com

Richiedi tag

Puoi aggiungere attestazioni personalizzate ai token specificando i tag nella richiesta GetWebIdentityToken API. Queste attestazioni vengono visualizzate nel campo request_tags del token e consentono di trasmettere informazioni specifiche che i servizi esterni possono utilizzare per decisioni di autorizzazione dettagliate. Puoi specificare fino a 50 tag per richiesta.

Richiesta di esempio:

```
response = sts_client.get_web_identity_token(
    Audience=['https://api.example.com'],
    SigningAlgorithm='ES384',
    Tags=[
        {'Key': 'team', 'Value': 'data-engineering'},
        {'Key': 'cost-center', 'Value': 'analytics'},
        {'Key': 'environment', 'Value': 'production'}
    ]
)
```

Dichiarazioni risultanti in token:

```
{
  "request_tags": {
    "team": "data-engineering",
    "cost-center": "analytics",
    "environment": "production"
  }
```

```
}
```

Controllo degli accessi con policy IAM

IAM fornisce diversi tipi di policy per controllare l'accesso alla funzionalità di federazione delle identità in uscita. Puoi utilizzare politiche basate sull'identità per controllare quali principali IAM possono richiedere token e applicare proprietà specifiche dei token come audience, lifetimes e algoritmi di firma. Le politiche di controllo dei servizi (SCP) ti consentono di applicare restrizioni a livello di organizzazione sulla generazione di token in tutti gli account delle tue organizzazioni. AWS Le politiche di controllo delle risorse (RCP) controllano l'accesso a livello di risorsa. Puoi anche utilizzare le policy degli endpoint VPC per limitare i principali che possono accedere all' AWS STS `GetWebIdentityToken`API tramite i tuoi endpoint VPC, aggiungendo controlli a livello di rete al tuo livello di sicurezza. Questa sezione spiega come implementare controlli di accesso granulari utilizzando questi tipi di policy e le chiavi di condizione.

Per richiedere i token di identità, un responsabile IAM deve disporre dell'autorizzazione. `sts:GetWebIdentityToken` Concedi questa autorizzazione tramite politiche di identità allegate agli utenti o ai ruoli IAM. Per consentire il passaggio dei tag (chiavi, coppie di valori) alla GetWebIdentityToken chiamata, il responsabile IAM deve disporre dell'`sts:TagGetWebIdentityToken`autorizzazione.

- Usa la chiave sts: IdentityTokenAudience condition per limitare i servizi esterni che possono ricevere token.
- Usa la chiave sts: DurationSeconds condition per imporre la durata massima dei token.
- Usa la chiave sts: SigningAlgorithm condition per richiedere algoritmi crittografici specifici.
- Usa la chiave di RequestTag condizione aws: confronta la coppia chiave-valore del tag passata nella richiesta con la coppia di tag specificata nella policy.
- Usa la chiave aws: TagKeys condition per confrontare le chiavi dei tag in una richiesta con le chiavi specificate nella policy.

Fai riferimento a IAM e alle chiavi di AWS STS condizione per saperne di più sulle chiavi di condizione disponibili nelle policy IAM.

Questo esempio di policy di identità combina più chiavi di condizione:

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "AllowTokenGenerationWithRestrictions",
            "Effect": "Allow",
            "Action": "sts:GetWebIdentityToken",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "sts:IdentityTokenAudience": [
                        "https://api1.example.com",
                        "https://api2.example.com"
                    ]
                },
                "NumericLessThanEquals": {
                    "sts:DurationSeconds": 300
                },
                "StringEquals": {
                    "sts:SigningAlgorithm": "ES384"
                }
            }
        }
    ]
}
```

Best practice

Segui questi consigli per federare in modo sicuro le tue AWS identità con servizi esterni.

- Utilizza una durata breve dei token: richiedi i token con la durata più breve che soddisfi le tue esigenze operative.
- Implementa l'accesso con privilegi minimi e limita le proprietà dei token con le politiche IAM: concedi l'`sts:GetWebIdentityToken`autorizzazione solo ai principali IAM che la richiedono. Utilizza le chiavi di condizione per specificare gli algoritmi di firma, i destinatari consentiti dei token e la durata massima dei token, in base alle tue esigenze.
- Convalida le dichiarazioni nei servizi esterni: per motivi di sicurezza, convalida sempre le affermazioni pertinenti come oggetto («sub»), pubblico («aud») ecc. per assicurarti che corrispondano ai valori previsti. Convalida le attestazioni personalizzate quando possibile per consentire decisioni di autorizzazione dettagliate nei servizi esterni.

Credenziali di sicurezza temporanee in IAM

Puoi utilizzare Servizio di token di sicurezza AWS (AWS STS) per creare e fornire agli utenti attendibili credenziali di sicurezza temporanee in grado di controllare l'accesso alle tue AWS risorse. Le credenziali di sicurezza temporanee funzionano quasi esattamente come le credenziali delle chiavi di accesso a lungo termine, con le seguenti differenze:

- Le credenziali di sicurezza provvisorie sono a breve termine, come implica il nome. Possono essere configurate per durare ovunque per pochi minuti o diverse ore. Dopo la scadenza delle credenziali, AWS non le riconosce più né consente alcun tipo di accesso dalle richieste API effettuate con esse.
- Le credenziali di sicurezza temporanee non sono archiviate con l'utente, ma vengono generate dinamicamente e fornite all'utente quando richiesto. Quando (o anche prima) le credenziali di sicurezza temporanee scadono, l'utente può richiedere nuove credenziali, purché l'utente che le richiede abbia ancora le autorizzazioni per farlo.

Di conseguenza, le credenziali temporanee presentano i seguenti vantaggi rispetto alle credenziali a lungo termine:

- Non è necessario distribuire o incorporare credenziali di AWS sicurezza a lungo termine con un'applicazione.
- Puoi fornire l'accesso alle tue AWS risorse agli utenti senza dover definire un' AWS identità per loro. Le credenziali provvisorie sono la base dei ruoli e della federazione delle identità.
- Le credenziali di sicurezza temporanee hanno una durata limitata, perciò non è necessario aggiornarle o revocarle in modo esplicito quando non sono più necessarie. Dopo che le credenziali di sicurezza temporanee scadono, non possono essere riutilizzate. È possibile specificare quando scadono le credenziali, fino a un limite massimo.

AWS STS and AWS regioni

Le credenziali di sicurezza temporanee sono generate da AWS STS. Per impostazione predefinita, AWS STS è un servizio globale con un unico endpoint in`https://sts.amazonaws.com`. Tuttavia, puoi anche scegliere di effettuare chiamate AWS STS API agli endpoint in qualsiasi altra regione supportata. Ciò può ridurre la latenza (server lag) effettuando le richieste a server in una regione geograficamente più vicina a te. Indipendentemente dalla regione dalla quale provengono, le

credenziali funzionano a livello globale. Per ulteriori informazioni, consulta Manage (Gestione) AWS STS in un Regione AWS.

Scenari comuni per le credenziali temporanee

Le credenziali temporanee sono utili in scenari che interessano la federazione delle identità, la delega, l'accesso tra account e i ruoli IAM.

Federazione delle identità

Puoi gestire le tue identità utente in un sistema esterno AWS e concedere agli utenti che accedono da tali sistemi l'accesso per eseguire AWS attività e accedere alle tue AWS risorse. IAM supporta due tipi di federazione delle identità. In entrambi i casi, le identità vengono archiviate all'esterno di. AWS La differenza è dove risiede il sistema esterno: nel data center o una parte terza sul Web. Per confrontare le funzionalità delle credenziali di sicurezza temporanee per la federazione delle identità, consulta Confronta le AWS STS credenziali.

Per ulteriori informazioni sui provider di identità esterni, consultare Fornitori di identità e federazione in AWS.

- Federazione OpenID Connect (OIDC): puoi consentire agli utenti di accedere utilizzando un noto provider di identità di terze parti come Login with Amazon, Facebook, Google o qualsiasi provider. OIDC-compatible Per la tua applicazione mobile o web, non devi creare un codice di accesso personalizzato o gestire le tue identità utente. L'uso della federazione OIDC ti aiuta a mantenere la tua Account AWS sicurezza, perché non devi distribuire credenziali di sicurezza a lungo termine, come le chiavi di accesso degli utenti IAM, con la tua applicazione. Per ulteriori informazioni, consulta Federazione OIDC.

  AWS STS La federazione OIDC supporta l'accesso con Amazon, Facebook, Google e qualsiasi provider di identità compatibile con OpenID Connect (OIDC).

  > **Note**
  >
  > Per le applicazioni mobili, consigliamo di utilizzare Amazon Cognito. Puoi utilizzare questo servizio con AWS gli SDK per lo sviluppo mobile per creare identità univoche per gli utenti e autenticarli per un accesso sicuro alle tue risorse. AWS Amazon Cognito supporta gli stessi provider di identità e supporta anche l'accesso non autenticato (guest) e consente di migrare i dati degli utenti quando un utente accede. AWS STS Amazon Cognito fornisce inoltre operazioni API per la sincronizzazione dei dati utente in modo che vengano

conservati quando gli utenti passano da un dispositivo all'altro. Per ulteriori informazioni, consulta Autenticazione con Amplify nella documentazione di Amplify.

- Federazione SAML: puoi autenticare gli utenti nella rete della tua organizzazione e quindi fornire loro l'accesso AWS senza creare nuove AWS identità per loro e richiedere loro di accedere con credenziali di accesso diverse. Questo è noto come approccio Single Sign-on all'accesso temporaneo. AWS STS supporta standard aperti come Security Assertion Markup Language (SAML) 2.0, con cui è possibile utilizzare Microsoft AD FS per sfruttare Microsoft Active Directory. È inoltre possibile utilizzare SAML 2.0 per gestire la soluzione per la federazione delle identità dell'utente. Per ulteriori informazioni, consulta Federazione SAML 2.0.
  - Broker federativo personalizzato: puoi utilizzare il sistema di autenticazione della tua organizzazione per concedere l'accesso alle risorse. AWS Per uno scenario di esempio, consultare Abilita l'accesso personalizzato del broker di identità al AWS console.
  - Federazione tramite SAML 2.0: puoi utilizzare SAML e il sistema di autenticazione dell'organizzazione per concedere l'accesso alle risorse AWS . Per ulteriori informazioni e uno scenario di esempio, consultare Federazione SAML 2.0.

Ruoli per l'accesso tra account

Molte organizzazioni mantengono più di un Account AWS. Utilizzando ruoli e l'accesso tra account, è possibile definire le identità degli utenti in un account e utilizzare tali identità per accedere alle risorse AWS in altri account che appartengono all'organizzazione. Questo approccio è noto come delega all'accesso temporaneo. Per ulteriori informazioni sulla creazione di ruoli tra account, consulta la sezione Creazione di un ruolo per fornire le autorizzazioni a un utente IAM. Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Ruoli per Amazon EC2

Se esegui applicazioni sulle istanze Amazon EC2 e tali applicazioni devono accedere alle risorse AWS , puoi fornire le credenziali di sicurezza temporanee alle istanze al momento dell'avvio. Queste credenziali di sicurezza temporanee sono disponibili a tutte le applicazioni che vengono eseguite sull'istanza, perciò non è necessario archiviare nessuna delle credenziali a lungo termine sull'istanza. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2.

Per ulteriori informazioni sulle credenziali del ruolo IAM di Amazon EC2, consulta Ruoli IAM per Amazon EC2 nella Guida per l'utente di Elastic Compute Cloud.

Altro AWS services

È possibile utilizzare credenziali di sicurezza temporanee per accedere alla maggior parte dei AWS servizi. Per un elenco dei servizi che accettano le credenziali di sicurezza temporanee, consultare AWS servizi supportati da IAM.

Applicazioni di esempio che usano credenziali temporanee

È possibile utilizzare Servizio di token di sicurezza AWS (AWS STS) per creare e fornire agli utenti attendibili credenziali di sicurezza temporanee in grado di controllare l'accesso alle AWS risorse. Per ulteriori informazioni su AWS STS, vedereCredenziali di sicurezza temporanee in IAM. Per vedere come è possibile utilizzarle AWS STS per gestire le credenziali di sicurezza temporanee, è possibile scaricare le seguenti applicazioni di esempio che implementano scenari di esempio completi:

- Abilitazione della federazione all' AWS utilizzo di Windows Active Directory, ADFS e SAML 2.0. Dimostra come delegare l'accesso tramite la federazione aziendale all'utilizzo di Windows Active Directory (AD), Active Directory Federation Services (ADFS) 2.0 e SAML (Security Assertion Markup Language) 2.0. AWS
- Abilita l'accesso personalizzato del broker di identità al AWS console. Dimostra come creare un proxy di federazione personalizzato che abilita l'autenticazione unica (SSO) in modo che gli utenti esistenti di Active Directory possano accedere alla Console di gestione AWS.
- Come usare Shibboleth for Single per. Sign-On Console di gestione AWS. Mostra come utilizzare Shibboleth e SAML per fornire agli utenti l'accesso Single Sign-On (SSO) alla Console di gestione AWS.

Esempi per la federazione OIDC

Le seguenti applicazioni di esempio illustrano come utilizzare la federazione OIDC con provider come Login with Amazon, Amazon Cognito, Facebook o Google. Puoi scambiare l'autenticazione di questi provider con credenziali di AWS sicurezza temporanee per accedere ai servizi. AWS

- Tutorial su Amazon Cognito: ti consigliamo di utilizzare Amazon Cognito con gli SDK per lo AWS sviluppo mobile. Amazon Cognito offre il modo più semplice per gestire l'identità per le applicazioni per dispositivi mobili e offre funzionalità aggiuntive come la sincronizzazione e l'identità tra più

dispositivi. Per ulteriori informazioni su Amazon Cognito, consulta Autenticazione con Amplify nella documentazione di Amplify.

Risorse aggiuntive per le credenziali di sicurezza temporanee

I seguenti scenari e applicazioni possono essere utili per l'utilizzo di credenziali di sicurezza temporanee:

- Come effettuare l'integrazione con il tuo provider di identità AWS STS SourceIdentity . Questo post mostra come impostare l' AWS STS `SourceIdentity`attributo quando usi Okta, Ping o OneLogin come IdP.
- Federazione OIDC. In questa sezione viene descritto come configurare i ruoli IAM quando si utilizza la federazione OIDC e l'API `AssumeRoleWithWebIdentity`.
- Accesso sicuro alle API con MFA. In questo argomento viene descritto come utilizzare i ruoli per richiedere l'autenticazione a più fattori (MFA) per proteggere le operazioni API sensibili nel tuo account.

Per ulteriori informazioni su politiche e autorizzazioni, AWS consulta i seguenti argomenti:

- Gestione degli accessi per AWS risorse
- Logica di valutazione delle policy.
- Gestione delle autorizzazioni di accesso alle risorse di Amazon S3 in Guida per l'utente di Amazon Simple Storage Service.
- Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Confronta le AWS STS credenziali

La tabella seguente confronta le funzionalità delle operazioni API AWS STS che restituiscono credenziali di sicurezza temporanee. Per informazioni sui diversi metodi che si possono utilizzare per richiedere le credenziali di sicurezza temporanee assumendo un ruolo, consultare Metodi per assumere un ruolo. Per ulteriori informazioni sulle diverse operazioni AWS STS API che consentono di passare i tag di sessione, consulta. Passa i tag della sessione in AWS STS

> **Note**
>
> Puoi inviare chiamate AWS STS API a un endpoint globale o a uno degli endpoint regionali. Se si seleziona un endpoint vicino, è possibile ridurre la latenza e migliorare le prestazioni delle chiamate API. Se non è più possibile comunicare con l'endpoint originale è anche possibile scegliere di indirizzare le chiamate verso un endpoint regionale alternativo. Se utilizzi uno dei vari AWS SDKs, utilizza il metodo SDK per specificare una regione prima di effettuare la chiamata API. Se costruiscono manualmente richieste API HTTP, è necessario indirizzare la richiesta all'endpoint corretto. Per ulteriori informazioni, consulta la sezione AWS STS relativa alle regioni e agli endpoint e la sezione Manage (Gestione) AWS STS in un Regione AWS.

AWS STS API	Chi può chiamare	Ciclo di vita delle credenziali (minimo | massimo | predefinito)	Supporto MFA[1]	Supporto di policy di sessione[2]	Restrizioni per le credenziali provvisorie risultanti
AssumeRe	Utente IAM o ruolo IAM con credenziali di sicurezza temporanee esistenti	15 min | Impostazione durata massima della sessione[3] | 1 ora	Sì	Sì	Non è possibile chiamare `GetFederationToken` o `GetSessionToken` .
AssumeReWithSAM	Qualsiasi intermediario utente deve passare	15 min | Impostazione	No	Sì	Non è possibile chiamare `GetFederationToken` o `GetSessionToken` .

AWS STS API	Chi può chiamare	Ciclo di vita delle credenziali (minimo | massimo | predefinito)	Supporto MFA[1]	Supporto di policy di sessione[2]	Restrizioni per le credenziali provvisorie risultanti
	una risposta di autenticazione SAML che indica l'autenticazione da un provider di identità noto	durata massima della sessione[3] | 1 ora			
AssumeRoleWithWebIdentity	Qualsiasi utente; il chiamante deve passare un token JWT conforme a OIDC che indica l'autenticazione da un provider di identità noto	15 min | Impostazione durata massima della sessione[3] | 1 ora	No	Sì	Non è possibile chiamare `GetFederationToken` o `GetSessionToken` .

AWS STS API	Chi può chiamare	Ciclo di vita delle credenziali (minimo | massimo | predefinito)	Supporto MFA[1]	Supporto di policy di sessione[2]	Restrizioni per le credenziali provvisorie risultanti
GetFederationToken	Utente IAM o Utente root dell'account AWS	Utente IAM: 15 m | 36 ore | 12 ore Utente root: 15 m | 1 ora | 1 ora	No	Sì	Impossibile chiamare le operazioni IAM utilizzando l' AWS API AWS CLI o. Questa limitazione non si applica alle sessioni della console. Impossibile chiamare AWS STS le operazioni ad eccezione di `GetCallerIdentity` .[4] L'accesso SSO alla console è consentito.[5]
GetSessionToken	Utente IAM o Utente root dell'account AWS	Utente IAM: 15 m | 36 ore | 12 ore Utente root: 15 m | 1 ora | 1 ora	Sì	No	Impossibile chiamare le operazioni API IAM a meno che le informazioni di MFA siano incluse con la richiesta. Impossibile chiamare le operazioni AWS STS API tranne `AssumeRole` o`GetCallerIdentity` . L'accesso SSO alla console non è consentito.[6]

[1] Supporto MFA. Puoi includere informazioni su un dispositivo di autenticazione a più fattori (MFA) quando chiami AssumeRole le operazioni GetSessionToken e API. In questo modo le credenziali di sicurezza provvisorie risultanti dalla chiamata API possono essere utilizzate solo dagli utenti autenticati con un dispositivo MFA. Per ulteriori informazioni, consulta Accesso sicuro alle API con MFA.

[2] Supporto per la policy di sessione. I criteri di sessione sono criteri che vengono passati come parametro quando si crea a livello di codice una sessione temporanea per un ruolo o AWS STS una sessione utente federata. Questa policy limita le autorizzazioni dalla policy basata su identità del ruolo o dell'utente che sono assegnate alla sessione. Le autorizzazioni della sessione risultanti sono l'intersezione delle policy basate sull'identità dell'entità e delle policy di sessione. Le policy di sessione non possono essere utilizzate per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata sull'identità del ruolo che viene assunto. Per ulteriori informazioni sulle autorizzazioni della sessione del ruolo, consulta Policy di sessione.

[3] Impostazione della durata massima della sessione. Utilizzare il parametro `DurationSeconds` per specificare la durata della sessione del ruolo da 900 secondi (15 minuti) fino all'impostazione di durata massima della sessione per il ruolo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Aggiornamento della durata massima della sessione per un ruolo.

3. GetCallerIdentity Non sono necessarie autorizzazioni per eseguire questa operazione. Se un amministratore aggiunge una policy al tuo utente o ruolo IAM che nega esplicitamente l'accesso all'operazione `sts:GetCallerIdentity`, puoi comunque eseguire questa operazione. Le autorizzazioni non sono necessarie perché le stesse informazioni vengono restituite quando a un utente o ruolo IAM viene negato l'accesso. Per visualizzare un esempio di risposta, consulta Non sono autorizzato a eseguire: iam: DeleteVirtualMFADevice.

[5] Accesso Single Sign-On (SSO) alla console. Per supportare l'SSO, AWS consente di chiamare un endpoint di federazione (`https://signin.aws.amazon.com/federation`) e passare credenziali di sicurezza temporanee. L'endpoint restituisce un token che è possibile utilizzare per creare un URL che effettua l'accesso di un utente direttamente nella console senza richiedere una password. Per ulteriori informazioni, consulta Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS e Come abilitare l'accesso tra più account alla console di AWS gestione nel blog sulla sicurezza. AWS

[6] Dopo aver recuperato le credenziali provvisorie, non è possibile accedere alla Console di gestione AWS inoltrando le credenziali all'endpoint Single Sign-On della federazione. Per ulteriori informazioni, consulta Abilita l'accesso personalizzato del broker di identità al AWS console.

Token di connessione al servizio

Alcuni AWS servizi richiedono l'autorizzazione per ottenere un token AWS STS service bearer prima di poter accedere alle loro risorse a livello di programmazione. Questi servizi supportano un protocollo che richiede l'utilizzo di un token di connessione invece di utilizzare un AWS Signature versione 4 per le richieste API tradizionale. Quando esegui AWS CLI o esegui operazioni AWS API che richiedono token al portatore, il AWS servizio richiede un token al portatore per tuo conto. Il servizio fornisce il token, che è possibile utilizzare per eseguire le operazioni successive in tale servizio.

AWS STS i service bearer token includono informazioni relative all'autenticazione principale originale che potrebbero influire sulle autorizzazioni dell'utente. Queste informazioni possono includere tag del principale, tag di sessione e policy di sessione. L'ID chiave di accesso del token inizia con il prefisso `ABIA`. Ciò consente di identificare le operazioni eseguite utilizzando i token di connessione al servizio nei log CloudTrail.

> **Important**
>
> Il token di connessione può essere utilizzato solo per le chiamate al servizio che lo genera e nella regione in cui è stato generato. Non è possibile utilizzare il token di connessione per eseguire operazioni in altri servizi o regioni.

Un esempio di servizio che supporta i bearer token è. AWS CodeArtifact Prima di poter interagire AWS CodeArtifact utilizzando un gestore di pacchetti come NPM, Maven o PIP, è necessario chiamare l'operazione. `aws codeartifact get-authorization-token` Questa operazione restituisce un token portatore che è possibile utilizzare per eseguire operazioni. AWS CodeArtifact In alternativa, è possibile utilizzare il comando `aws codeartifact login` che completa la stessa operazione e quindi configura automaticamente il client.

Se esegui un'azione in un AWS servizio che genera un token bearer per te, devi disporre delle seguenti autorizzazioni nella tua politica IAM:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "AllowServiceBearerToken",
            "Effect": "Allow",
            "Action": "sts:GetServiceBearerToken",
            "Resource": "*"
        }
    ]
}
```

Per un esempio di token portatore di servizi, consulta Utilizzo di policy basate sulle identità per AWS CodeArtifact nella Guida per l'utente di AWS CodeArtifact.

Richiedere credenziali di sicurezza temporanee

Per richiedere credenziali di sicurezza temporanee, puoi utilizzare le operazioni Servizio di token di sicurezza AWS (AWS STS) nell' AWS API. Queste includono operazioni per creare e fornire agli utenti affidabili credenziali di sicurezza temporanee in grado di controllare l'accesso alle tue AWS risorse. Per ulteriori informazioni su AWS STS, vedereCredenziali di sicurezza temporanee in IAM. Per informazioni sui diversi metodi che si possono utilizzare per richiedere credenziali di sicurezza temporanee assumendo un ruolo, consultare Metodi per assumere un ruolo.

Per chiamare le operazioni API, è possibile utilizzare uno degli SDK AWS. Gli SDK sono disponibili per un'ampia gamma di linguaggi di programmazione e ambienti, tra cui Java, .NET, Python, Ruby, Android e iOS. I kit SDK si occupano di attività quali la firma crittografica delle richieste, la ripetizione delle richieste se necessario e la gestione delle risposte agli errori. Puoi anche utilizzare l'API AWS STS Query, descritta nell'Servizio di token di sicurezza AWS API Reference. Infine, due strumenti a riga di comando supportano i AWS STS comandi: the AWS Command Line Interface, e the AWS Tools for Windows PowerShell.

Le operazioni AWS STS API creano una nuova sessione con credenziali di sicurezza temporanee che includono una coppia di chiavi di accesso e un token di sessione. La coppia di chiavi di accesso è composta da un ID chiave di accesso e da una chiave segreta. Gli utenti (o un'applicazione che l'utente esegue) possono utilizzare queste credenziali per accedere alle risorse.

È possibile creare una sessione di ruolo e passare le policy di sessione e i tag di sessione a livello di codice utilizzando AWS STS le operazioni API. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. Per ulteriori informazioni sulle policy di sessione, consulta Policy di sessione. Per ulteriori informazioni sui tag di sessione, consultare Passa i tag della sessione in AWS STS.

> **Note**
>
> La dimensione del token di sessione restituito dalle operazioni AWS STS API non è fissa. È consigliabile di non effettuare alcuna supposizione sulle dimensioni massime. La dimensione tipica dei token è inferiore a 4.096 byte, ma essa può variare.

Utilizzo AWS STS con AWS Regioni

È possibile inviare chiamate AWS STS API a un endpoint globale o a uno degli endpoint regionali. Se si seleziona un endpoint vicino, è possibile ridurre la latenza e migliorare le prestazioni delle chiamate API. Se non è più possibile comunicare con l'endpoint originale è anche possibile scegliere di indirizzare le chiamate verso un endpoint regionale alternativo.

Se stai utilizzando uno dei vari AWS SDK, utilizza quel metodo SDK per specificare una regione prima di effettuare la chiamata API. Se costruiscono manualmente richieste API HTTP, è necessario indirizzare la richiesta all'endpoint corretto. Per ulteriori informazioni, consulta la sezione AWS STS relativa alle regioni e agli endpoint e la sezione Manage (Gestione) AWS STS in un Regione AWS.

Di seguito sono riportate le operazioni API che puoi utilizzare per acquisire credenziali temporanee da utilizzare nel tuo AWS ambiente e nelle tue applicazioni.

Richiesta di credenziali per la delega tra account e federazione tramite un gestore di identità personalizzato

L'operazione `AssumeRole` API è utile per consentire agli utenti IAM esistenti di accedere a AWS risorse a cui non hanno già accesso. Ad esempio, l'utente potrebbe aver bisogno di accedere alle risorse in un altro Account AWS. Inoltre, è utile come strumento per ottenere temporaneamente l'accesso privilegiato, ad esempio per fornire l'autenticazione a più fattori (MFA). È necessario chiamare questa API utilizzando le credenziali attive. Per sapere chi può chiamare questa operazione, vedere Confronta le AWS STS credenziali. Per ulteriori informazioni, consulta Creazione di un ruolo per fornire le autorizzazioni a un utente IAM e Accesso sicuro alle API con MFA.

Per richiedere credenziali di sicurezza temporanee per la delega tra account e federazione tramite un gestore di identità personalizzato

1. Effettua l'autenticazione con le tue credenziali AWS di sicurezza. Questa chiamata deve essere effettuata utilizzando le credenziali di sicurezza AWS .
2. Chiama `AssumeRole`l'operazione.

L'esempio seguente mostra una richiesta di esempio e la risposta utilizzando `AssumeRole`. Questa richiesta di esempio assume il ruolo `demo` per la durata specificata, inclusi policy di sessione, tag di sessione, ID esterno e identità di origine. La sessione risultante è denominata `John-session`.

Example Richiesta di esempio

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=AssumeRole
&RoleSessionName=John-session
&RoleArn=arn:aws:iam::123456789012:role/demo
&Policy=%7B%22Version%22%3A%222012-10-17       %22%2C%22Statement%22%3A%5B%7B%22Sid
%22%3A%20%22Stmt1%22%2C%22Effect%22%3A%20%22Allow%22%2C%22Action%22%3A%20%22s3%3A*
%22%2C%22Resource%22%3A%20%22*%22%7D%5D%7D
&DurationSeconds=1800
&Tags.member.1.Key=Project
&Tags.member.1.Value=Pegasus
&Tags.member.2.Key=Cost-Center
&Tags.member.2.Value=12345
&ExternalId=123ABC
&SourceIdentity=DevUser123
&AUTHPARAMS
```

Il valore della policy mostrato nell'esempio precedente è la URL-encoded versione della seguente policy:

JSON

```
{"Version":"2012-10-17",       "Statement":
[{"Sid":"Stmt1","Effect":"Allow","Action":"s3:*","Resource":"*"}]}
```

Il parametro `AUTHPARAMS` nell'esempio è un segnaposto per la propria firma. Una firma è l'informazione di autenticazione che devi includere nelle richieste API AWS HTTP. Per creare le richieste API ti consigliamo di utilizzare gli SDK AWS poiché, tra gli altri vantaggi, gli SDK gestiscono la firma delle richieste per tuo conto. Se devi creare e firmare le richieste API manualmente, vedi Signing AWS Requests By Signature Version 4 nella sezione Riferimenti generali di Amazon Web Services per scoprire come firmare una richiesta.

Oltre alle credenziali di sicurezza temporanee, la risposta include l'Amazon Resource Name (ARN) per l'utente federato e il periodo di scadenza delle credenziali.

Example Esempio di risposta

```
<AssumeRoleResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
<AssumeRoleResult>
<SourceIdentity>DevUser123</SourceIdentity>
<Credentials>
  <SessionToken>
   AQoDYXdzEPT//////////wEXAMPLEtc764bNrC9SAPBSM22wDOk4x4HIZ8j4FZTwdQW
   LWsKWHGBuFqwAeMicRXmxfpSPfIeoIYRqTflfKD8YUuwthAx7mSEI/qkPpKPi/kMcGd
   QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VSXDvp75YU
   9HFvlRd8Tx6q6fE8YQcHNVXAkiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
   +scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3iSIlTJabIQwj2ICCR/oLxBA==
  </SessionToken>
  <SecretAccessKey>
   wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY
  </SecretAccessKey>
  <Expiration>2019-07-15T23:28:33.359Z</Expiration>
  <AccessKeyId>AKIAIOSFODNN7EXAMPLE</AccessKeyId>
</Credentials>
<AssumedRoleUser>
  <Arn>arn:aws:sts::123456789012:assumed-role/demo/John</Arn>
  <AssumedRoleId>ARO123EXAMPLE123:John</AssumedRoleId>
</AssumedRoleUser>
<PackedPolicySize>34</PackedPolicySize>
<SessionTokenSize>1428</SessionTokenSize>
<SessionTokenUtilization>34</SessionTokenUtilization>
</AssumeRoleResult>
<ResponseMetadata>
<RequestId>c6104cbe-af31-11e0-8154-cbc7ccf896c7</RequestId>
</ResponseMetadata>
</AssumeRoleResponse>
```

> **Note**
>
> Le policy di sessione e i tag di sessione si aggiungono alla dimensione del token di sessione. Se il token supera il limite di dimensione, la richiesta ha esito negativo, anche se il testo in chiaro rientra negli altri limiti. L'elemento di `SessionTokenUtilization` risposta mostra la

percentuale del limite di dimensione utilizzata dal token. L'elemento di `PackedPolicySize` risposta è obsoleto. Usare invece `SessionTokenUtilization`.

Richiesta di credenziali tramite un provider OIDC

L'operazione API `AssumeRoleWithWebIdentity` restituisce un set di credenziali di sicurezza AWS temporanee in cambio di un JSON Web Token (JWT). Ciò include i provider di identità pubblici, come Login with Amazon, Facebook, Google e i provider che emettono JWT compatibili con il discovery di OpenID Connect (OIDC), come action o Azure Devops. GitHub Per ulteriori informazioni, consulta Federazione OIDC.

> **Note**
>
> Le richieste `AssumeRoleWithWebIdentity` non sono firmate e non richiedono credenziali AWS .

Richiesta di credenziali tramite un provider OIDC

1. Chiama `AssumeRoleWithWebIdentity`l'operazione.

   Quando chiami`AssumeRoleWithWebIdentity`, AWS convalida il token presentato verificando la firma digitale utilizzando le chiavi pubbliche rese disponibili tramite il set di chiavi web JSON (JWKS) del tuo IdP. Se il token è valido e tutte le condizioni stabilite nella policy di fiducia dei ruoli IAM sono soddisfatte, AWS restituisce le seguenti informazioni:

   - Un set di credenziali di sicurezza temporanee. Consistono in un ID chiave di accesso, in una Secret Access Key e in un token di sessione.
   - l'ID del ruolo e l'ARN del ruolo assunto.
   - Un valore `SubjectFromWebIdentityToken` che contiene l'ID utente univoco.
2. L'applicazione può quindi utilizzare le credenziali di sicurezza temporanee restituite nella risposta per effettuare chiamate AWS API. Si tratta della stessa procedura utilizzata per effettuare una chiamata AWS API con credenziali di sicurezza a lungo termine. La differenza è che è necessario includere il token di sessione, che consente di AWS verificare che le credenziali di sicurezza temporanee siano valide.

L'applicazione dovrebbe memorizzare nella cache le credenziali restituite da AWS STS e aggiornarle secondo necessità. Se l'applicazione è creata utilizzando un AWS SDK, l'SDK dispone di provider di credenziali in grado di gestire le chiamate `AssumeRoleWithWebIdentity` e l'aggiornamento AWS delle credenziali prima che scadano. Per ulteriori informazioni, consulta Provider di credenziali standardizzati di SDK AWS e strumenti nella Guida di riferimento agli SDK AWS e agli strumenti.

Richiesta di credenziali tramite un gestore di identità SAML 2.0

L'operazione API `AssumeRoleWithSAML` restituisce un set di credenziali di sicurezza temporanee per i principali federati SAML che sono autenticati dal sistema di identità esistente dell'organizzazione. Gli utenti devono utilizzare inoltre SAML 2.0 (Security Assertion Markup Language) per inoltrare le informazioni di autenticazione e autorizzazione ad AWS. Questa operazione di API è utile in organizzazioni che hanno integrato i propri sistemi di identità (ad esempio Windows Active Directory o OpenLDAP) con software che è in grado di produrre asserzioni SAML. Tale integrazione fornisce informazioni sulle autorizzazioni e identità dell'utente (ad esempio Active Directory Federation Services o Shibboleth). Per ulteriori informazioni, consulta Federazione SAML 2.0.

1. Chiama `AssumeRoleWithSAML`l'operazione.

   Si tratta di una chiamata non firmata, il che significa che non è necessario autenticare le credenziali AWS di sicurezza prima di effettuare la richiesta.

   > **Note**
   >
   > Una chiamata a `AssumeRoleWithSAML` non è firmata (crittografata). Pertanto, è opportuno includere le policy di sessione facoltative solo se la richiesta viene trasmessa attraverso un intermediario affidabile. In questo caso, qualcuno potrebbe modificare la policy per rimuovere le limitazioni.

2. Quando chiami`AssumeRoleWithSAML`, AWS verifica l'autenticità dell'asserzione SAML. Supponendo che il provider di identità convalidi l'asserzione, AWS ti restituisce le seguenti informazioni:

   - Un set di credenziali di sicurezza temporanee. Consistono in un ID chiave di accesso, in una Secret Access Key e in un token di sessione.
   - l'ID del ruolo e l'ARN del ruolo assunto.

- Un valore `Audience` che contiene il valore dell'attributo `Recipient` dell'elemento `SubjectConfirmationData` dell'asserzione SAML.
- Un valore `Issuer` che contiene il valore dell'attributo `Issuer` dell'elemento dell'asserzione SAML.
- Un `NameQualifier` elemento che contiene un valore hash creato a partire dal `Issuer` valore, dall' Account AWS ID e dal nome descrittivo del provider SAML. In combinazione con l'elemento `Subject`, permette di identificare in modo univoco il principale federato SAML.
- Un elemento `Subject` che contiene il valore dell'elemento `NameID` nell'elemento `Subject` dell'asserzione SAML.
- Un elemento `SubjectType` che indica il formato dell'elemento `Subject`. Il valore può essere `persistent`, `transient` o il pieno `Format` URI dagli elementi `Subject` e `NameID` utilizzati nell'asserzione SAML. Per ulteriori informazioni sull'attributo `Format` dell'elemento `NameID`, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

3. Utilizza le credenziali di sicurezza temporanee restituite nella risposta per effettuare AWS chiamate API. Si tratta della stessa procedura utilizzata per effettuare una chiamata AWS API con credenziali di sicurezza a lungo termine. La differenza è che è necessario includere il token della sessione, che consente ad AWS di verificare che le credenziali di sicurezza provvisorie siano valide.

L'app deve memorizzare le credenziali. Come impostazione predefinita, le credenziali scadono dopo un'ora. Se non stai utilizzando l'AmazonSTSCredentialsProviderazione nell' AWS SDK, spetta a te e alla tua app chiamare `AssumeRoleWithSAML` nuovamente. Chiamare questa operazione per ottenere un nuovo set di credenziali di sicurezza provvisorie prima che i vecchi scadere.

Richiesta di credenziali tramite un gestore di identità personalizzato

L'operazione `GetFederationToken` API restituisce un set di credenziali di sicurezza temporanee per gli utenti principali AWS STS federati. Questa API differisce da `AssumeRole` per il fatto che il periodo di scadenza predefinito è notevolmente più lungo (12 ore invece di un'ora). Inoltre, è possibile utilizzare il parametro `DurationSeconds` per specificare una durata per le credenziali di sicurezza temporanee perché rimanga valido. Le credenziali risultanti sono valide per la durata specificata, da 900 secondi (15 minuti) a un massimo di 129.600 secondi (36 ore). Un periodo di scadenza più lungo può aiutare a ridurre il numero di chiamate AWS perché non è necessario ottenere nuove credenziali con la stessa frequenza.

1. Effettua l'autenticazione con le credenziali di AWS sicurezza del tuo utente IAM specifico. Questa chiamata deve essere effettuata utilizzando credenziali di AWS sicurezza valide.
2. Chiama `GetFederationToken`l'operazione.

La chiamata `GetFederationToken` restituisce le credenziali di sicurezza temporanee che consistono nel token di sessione, nella chiave di accesso, nella chiave segreta e nella scadenza. È possibile utilizzare `GetFederationToken` se si desidera gestire le autorizzazioni nella propria organizzazione (ad esempio, utilizzando l'applicazione proxy per assegnare le autorizzazioni).

L'esempio seguente mostra una richiesta di esempio e la risposta che utilizza `GetFederationToken`. Questa richiesta di esempio consente di federare l'utente chiamante per la durata specificata tramite l'ARN della policy di sessione e i tag di sessione. La sessione risultante è denominata `Jane-session`.

Example Richiesta di esempio

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=GetFederationToken
&Name=Jane-session
&PolicyArns.member.1.arn==arn%3Aaws%3Aiam%3A%3A123456789012%3Apolicy%2FRole1policy
&DurationSeconds=1800
&Tags.member.1.Key=Project
&Tags.member.1.Value=Pegasus
&Tags.member.2.Key=Cost-Center
&Tags.member.2.Value=12345
&AUTHPARAMS
```

L'ARN della policy mostrato nell'esempio precedente include il seguente ARN: URL-encoded

`arn:aws:iam::123456789012:policy/Role1policy`

Inoltre, si noti che il parametro `&AUTHPARAMS` nell'esempio è inteso come segnaposto per le informazioni di autenticazione. Questa è la firma, che è necessario includere nelle richieste API AWS HTTP. Per creare le richieste API ti consigliamo di utilizzare gli SDK AWS poiché, tra gli altri vantaggi, gli SDK gestiscono la firma delle richieste per tuo conto. Se devi creare e firmare le richieste API manualmente, vedi Signing AWS Requests By Signature Version 4 nella sezione Riferimenti generali di Amazon Web Services per scoprire come firmare una richiesta.

Oltre alle credenziali di sicurezza temporanee, la risposta include l'Amazon Resource Name (ARN) per l'utente federato e il periodo di scadenza delle credenziali.

Example Esempio di risposta

```
<GetFederationTokenResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
<GetFederationTokenResult>
<Credentials>
  <SessionToken>
   AQoDYXdzEPT//////////wEXAMPLEtc764bNrC9SAPBSM22wDOk4x4HIZ8j4FZTwdQW
   LWsKWHGBuFqwAeMicRXmxfpSPfIeoIYRqTflfKD8YUuwthAx7mSEI/qkPpKPi/kMcGd
   QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VSXDvp75YU
   9HFvlRd8Tx6q6fE8YQcHNVXAkiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
   +scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3iSIlTJabIQwj2ICCEXAMPLE==
  </SessionToken>
  <SecretAccessKey>
  wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY
  </SecretAccessKey>
  <Expiration>2019-04-15T23:28:33.359Z</Expiration>
  <AccessKeyId>AKIAIOSFODNN7EXAMPLE;</AccessKeyId>
</Credentials>
<FederatedUser>
  <Arn>arn:aws:sts::123456789012:federated-user/Jean</Arn>
  <FederatedUserId>123456789012:Jean</FederatedUserId>
</FederatedUser>
<PackedPolicySize>26</PackedPolicySize>
<SessionTokenSize>1105</SessionTokenSize>
<SessionTokenUtilization>26</SessionTokenUtilization>
</GetFederationTokenResult>
<ResponseMetadata>
<RequestId>c6104cbe-af31-11e0-8154-cbc7ccf896c7</RequestId>
</ResponseMetadata>
</GetFederationTokenResponse>
```

> **Note**
>
> Le policy di sessione e i tag di sessione si aggiungono alla dimensione del token di sessione. Se il token supera il limite di dimensione, la richiesta ha esito negativo, anche se il testo in chiaro rientra negli altri limiti. L'elemento di `SessionTokenUtilization` risposta mostra la percentuale del limite di dimensione utilizzata dal token. L'elemento di `PackedPolicySize` risposta è obsoleto. Usare invece `SessionTokenUtilization`.

AWS consiglia di concedere autorizzazioni a livello di risorsa (ad esempio, allegando una policy basata sulle risorse a un bucket Amazon S3), puoi omettere il parametro. `Policy` Tuttavia, se non includi una policy per l'utente principale AWS STS federato, le credenziali di sicurezza temporanee non concederanno alcuna autorizzazione. In questo caso, è necessario utilizzare le policy delle risorse per concedere all'utente federato l'accesso alle risorse AWS .

Ad esempio, supponiamo che il tuo Account AWS numero sia 111122223333 e che tu abbia un bucket Amazon S3 a cui vuoi consentire l'accesso a Susan. Le credenziali di sicurezza temporanee di Susan non includono una policy per il bucket. In questo caso, è necessario accertarsi che il bucket disponga di una policy con un ARN corrisponda all'ARN di Susan, ad esempio `arn:aws:sts::111122223333:federated-user/Susan`.

Richiesta di credenziali per gli utenti in ambienti non attendibili

L'operazione API `GetSessionToken` restituisce un set di credenziali di sicurezza temporanee a un utente IAM esistente. Ciò è utile per fornire una maggiore sicurezza, ad esempio per consentire le AWS richieste solo quando l'MFA è abilitato per l'utente IAM. Poiché le credenziali sono temporanee, forniscono maggiore sicurezza quando hai un utente IAM che accede alle risorse tramite un ambiente meno sicuro. Esempi di ambienti meno protetti includono dispositivi mobili o browser web.

1. Effettua l'autenticazione con le credenziali di AWS sicurezza del tuo utente IAM specifico. Questa chiamata deve essere effettuata utilizzando credenziali di AWS sicurezza valide.
2. Chiama `GetSessionToken`l'operazione.
3. `GetSessionToken` restituisce le credenziali di sicurezza temporanee, ossia un token di sessione, l'ID chiave di accesso e una chiave di accesso segreta.

Per impostazione predefinita, le credenziali di sicurezza temporanee per un utente IAM sono valide per un massimo di 12 ore. Tuttavia, è possibile richiedere una durata di soli 15 minuti o fino a 36 ore utilizzando il parametro `DurationSeconds`. Per motivi di sicurezza, la durata di un token per un Utente root dell'account AWS è limitata a un'ora.

L'esempio seguente mostra una richiesta di esempio e la risposta utilizzando `GetSessionToken`. La risposta include inoltre il periodo di scadenza delle credenziali di sicurezza temporanee.

Example Richiesta di esempio

```
https://sts.amazonaws.com/
?Version=2011-06-15
```

```
&Action=GetSessionToken
&DurationSeconds=1800
&AUTHPARAMS
```

Il parametro AUTHPARAMS nell'esempio è un segnaposto per la propria firma. Una firma è l'informazione di autenticazione che devi includere nelle richieste API AWS HTTP. Per creare le richieste API ti consigliamo di utilizzare gli SDK AWS poiché, tra gli altri vantaggi, gli SDK gestiscono la firma delle richieste per tuo conto. Se devi creare e firmare le richieste API manualmente, vedi Signing AWS Requests By Signature Version 4 nella sezione Riferimenti generali di Amazon Web Services per scoprire come firmare una richiesta.

Example Esempio di risposta

```
<GetSessionTokenResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
<GetSessionTokenResult>
<Credentials>
  <SessionToken>
   AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT+FvwqnKwRcOIfrRh3c/L
   To6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/IvU1dYUg2RVAJBanLiHb4IgRmpRV3z
   rkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/AXlzBBko7b15fjrBs2+cTQtp
   Z3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE
  </SessionToken>
  <SecretAccessKey>
  wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY
  </SecretAccessKey>
  <Expiration>2011-07-11T19:55:29.611Z</Expiration>
  <AccessKeyId>AKIAIOSFODNN7EXAMPLE</AccessKeyId>
</Credentials>
<SessionTokenSize>856</SessionTokenSize>
<SessionTokenUtilization>20</SessionTokenUtilization>
</GetSessionTokenResult>
<ResponseMetadata>
<RequestId>58c5dbae-abef-11e0-8cfe-09039844ac7d</RequestId>
</ResponseMetadata>
</GetSessionTokenResponse>
```

Facoltativamente, la `GetSessionToken` richiesta può includere `TokenCode` valori per `SerialNumber` la verifica dell' AWS autenticazione a più fattori (MFA). Se i valori forniti sono validi, AWS STS fornisce credenziali di sicurezza temporanee che includono lo stato dell'autenticazione MFA. Le credenziali di sicurezza temporanee possono quindi essere utilizzate per accedere alle operazioni MFA-protected API o ai AWS siti Web finché l'autenticazione MFA è valida.

L'esempio seguente mostra una richiesta `GetSessionToken`, che include un codice di verifica MFA e il numero di serie del dispositivo.

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=GetSessionToken
&DurationSeconds=7200
&SerialNumber=YourMFADeviceSerialNumber
&TokenCode=123456
&AUTHPARAMS
```

> **Note**
>
> La chiamata AWS STS può essere rivolta all'endpoint globale o a uno qualsiasi degli endpoint regionali su cui attivi il tuo. Account AWS Per ulteriori informazioni, consulta la sezione di AWS STS relativa alle regioni e agli endpoint.
> Il parametro `AUTHPARAMS` nell'esempio è un segnaposto per la propria firma. Una firma è l'informazione di autenticazione che devi includere nelle richieste API AWS HTTP. Per creare le richieste API ti consigliamo di utilizzare gli SDK AWS poiché, tra gli altri vantaggi, gli SDK gestiscono la firma delle richieste per tuo conto. Se devi creare e firmare le richieste API manualmente, vedi Signing AWS Requests By Signature Version 4 nella sezione Riferimenti generali di Amazon Web Services per scoprire come firmare una richiesta.

Usa credenziali temporanee con AWS risorse

Puoi utilizzare le credenziali di sicurezza temporanee per effettuare richieste programmatiche di AWS risorse utilizzando l' AWS API AWS CLI or (utilizzando gli SDK). AWS Le credenziali temporanee forniscono le stesse autorizzazioni delle credenziali di sicurezza a lungo termine, come ad esempio le credenziali degli utenti IAM. Tuttavia, ci sono alcune differenze:

- Quando si effettua una chiamata utilizzando credenziali di sicurezza temporanee, la chiamata deve includere un token di sessione, che viene restituito insieme a tali credenziali temporanee. AWS utilizza il token di sessione per convalidare le credenziali di sicurezza temporanee.
- Le credenziali temporanee scadono dopo un intervallo di tempo specificato. Dopo che le credenziali temporanee scadono, tutte le chiamate effettuate con tali credenziali verranno respinte, pertanto dovrai generare un nuovo set di credenziali temporanee. Le credenziali temporanee non possono essere prorogate o aggiornate oltre l'intervallo specificato in origine.

- Quando si utilizzano credenziali temporanee per effettuare una richiesta, l'entità potrebbe includere un set di tag. Questi tag provengono da tag di sessione e tag associati al ruolo assunto. Per ulteriori informazioni sui tag di sessione, consultare Passa i tag della sessione in AWS STS.

Se utilizzi gli AWS SDK, AWS Command Line Interface (AWS CLI) o gli Strumenti per Windows PowerShell, il modo per ottenere e utilizzare le credenziali di sicurezza temporanee varia a seconda del contesto. Se stai eseguendo codice o PowerShell comandi Tools for Windows all'interno di un'istanza EC2, puoi sfruttare i ruoli per Amazon EC2. AWS CLI Altrimenti, puoi richiamare un'API AWS STS per ottenere le credenziali provvisorie e utilizzarle in modo esplicito per effettuare chiamate ai servizi AWS .

> **Note**
>
> Puoi usare Servizio di token di sicurezza AWS (AWS STS) per creare e fornire agli utenti affidabili credenziali di sicurezza temporanee in grado di controllare l'accesso alle tue risorse. AWS Per ulteriori informazioni su AWS STS, vedereCredenziali di sicurezza temporanee in IAM. AWS STS è un servizio globale con un endpoint predefinito in`https://sts.amazonaws.com`. Questo endpoint si trova nella regione Stati Uniti orientali (Virginia settentrionale), sebbene le credenziali ottenute da questo e da altri endpoint siano valide a livello globale. Queste credenziali funzionano con servizi e risorse in qualsiasi regione. Puoi anche scegliere di effettuare chiamate AWS STS API agli endpoint in una qualsiasi delle regioni supportate. Ciò può ridurre la latenza effettuando le richieste da server in una regione geograficamente più vicina a te. Indipendentemente dalla regione dalla quale provengono, le credenziali funzionano a livello globale. Per ulteriori informazioni, consulta Manage (Gestione) AWS STS in un Regione AWS.

Indice

- Utilizzo delle credenziali temporanee nelle istanze Amazon EC2
- Utilizzo di credenziali di sicurezza temporanee con AWS SDK
- Utilizzo di credenziali di sicurezza temporanee con AWS CLI
- Utilizzo delle credenziali di sicurezza temporanee con le operazioni API
- Ulteriori informazioni

Utilizzo delle credenziali temporanee nelle istanze Amazon EC2

Se desideri eseguire AWS CLI comandi o codice all'interno di un'istanza EC2, il modo consigliato per ottenere le credenziali è utilizzare i ruoli per Amazon EC2. È possibile creare un ruolo IAM che specifichi le autorizzazioni che si desidera concedere alle applicazioni che vengono eseguite sulle istanze EC2. Quando si avvia l'istanza, si associa il ruolo all'istanza.

Le applicazioni e AWS CLI i PowerShell comandi Tools for Windows eseguiti sull'istanza possono quindi ottenere credenziali di sicurezza temporanee automatiche dai metadati dell'istanza. Non è necessario ottenere esplicitamente le credenziali di sicurezza temporanee. Gli AWS SDK e gli strumenti per Windows ottengono PowerShell automaticamente le credenziali dall'EC2 Instance Metadata Service (IMDS) e le utilizzano. AWS CLI Le credenziali temporanee hanno le autorizzazioni che si definiscono per il ruolo associato all'istanza.

Per maggiori informazioni ed esempi, consultare quanto segue:

- Utilizzo dei ruoli IAM per concedere l'accesso alle AWS risorse su Amazon Elastic Compute Cloud — AWS SDK per Java
- Concessione dell'accesso tramite un ruolo IAM — AWS SDK per .NET
- Creazione di un ruolo: AWS SDK per Ruby

Utilizzo di credenziali di sicurezza temporanee con AWS SDK

Per utilizzare le credenziali di sicurezza temporanee nel codice, richiami a livello di codice un' AWS STS API like `AssumeRole` ed estrai le credenziali e il token di sessione risultanti. Questi valori vengono quindi utilizzati come credenziali per le chiamate successive a. AWS L'esempio seguente mostra uno pseudocodice su come utilizzare le credenziali di sicurezza temporanee se utilizzi un SDK: AWS

```
assumeRoleResult = AssumeRole(role-arn);
tempCredentials = new SessionAWSCredentials(
   assumeRoleResult.AccessKeyId,
   assumeRoleResult.SecretAccessKey,
   assumeRoleResult.SessionToken);
s3Request = CreateAmazonS3Client(tempCredentials);
```

Per un esempio scritto in Python (usando AWS SDK per Python (Boto)), consultare Passa a un ruolo IAM (AWS API). In questo esempio viene illustrato come richiamare `AssumeRole` per ottenere le

credenziali di sicurezza temporanee e quindi utilizzare tali credenziali per effettuare una chiamata ad Amazon S3.

Per dettagli su come richiamare `AssumeRole`, `GetFederationToken` e altre operazioni API, consulta la Documentazione di riferimento delle API Servizio di token di sicurezza AWS. Per informazioni su come ottenere le credenziali di sicurezza provvisorie e il token di sessione dal risultato, consulta la documentazione dell'SDK in uso. Puoi trovare la documentazione per tutti gli AWS SDK nella pagina principale della AWS documentazione, nella sezione SDK e toolkit.

È necessario accertarsi che sia possibile ottenere un nuovo set di credenziali prima della scadenza. In alcuni SDK, è possibile utilizzare un provider che gestisca il proprio processo di aggiornamento delle credenziali; controllare la documentazione del kit SDK che si sta utilizzando.

Utilizzo di credenziali di sicurezza temporanee con AWS CLI

È possibile utilizzare le credenziali di sicurezza temporanee con AWS CLI Questo può essere utile per testare le policy.

Tramite la AWS CLI, puoi richiamare un'API AWS STS come `AssumeRole` o `GetFederationToken` e acquisire l'output risultante. L'esempio seguente mostra una chiamata a `AssumeRole` che invia l'output a un file. Nell'esempio, si presume che il `profile` parametro sia un profilo nel file di AWS CLI configurazione. Si presume inoltre di fare riferimento alle credenziali di un utente IAM che disponga delle autorizzazioni per assumere il ruolo.

```
aws sts assume-role --role-arn arn:aws:iam::123456789012:role/role-name --role-session-
name "RoleSession1" --profile IAM-user-name > assume-role-output.txt
```

Quando il comando viene completato, è possibile estrarre l'ID della chiave di accesso, la chiave di accesso segreta e il token di sessione da qualunque posto sia stato instradato. È possibile farlo manualmente o utilizzando uno script. È possibile assegnare questi valori alle variabili di ambiente.

Quando si eseguono AWS CLI i comandi, AWS CLI cerca le credenziali in un ordine specifico, prima nelle variabili di ambiente e poi nel file di configurazione. Pertanto, dopo aver inserito le credenziali temporanee nelle variabili di ambiente, AWS CLI utilizza tali credenziali per impostazione predefinita. (Se si specifica un `profile` parametro nel comando, vengono AWS CLI ignorate le variabili di ambiente. Invece, AWS CLI cerca nel file di configurazione, che consente di sovrascrivere le credenziali nelle variabili di ambiente, se necessario.)

L'esempio seguente mostra come è possibile impostare le variabili di ambiente per le credenziali di sicurezza temporanee e quindi chiamare un comando. AWS CLI Poiché nel AWS CLI comando non

è incluso alcun `profile` parametro, AWS CLI cerca prima le credenziali nelle variabili di ambiente e quindi utilizza le credenziali temporanee.

Linux

```
$ export AWS_ACCESS_KEY_ID=ASIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of session token>
$ aws ec2 describe-instances --region us-west-1
```

Windows

```
C:\> SET AWS_ACCESS_KEY_ID=ASIAIOSFODNN7EXAMPLE
C:\> SET AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> SET AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of token>
C:\> aws ec2 describe-instances --region us-west-1
```

Utilizzo delle credenziali di sicurezza temporanee con le operazioni API

Se stai effettuando richieste API HTTPS dirette a AWS, puoi firmare tali richieste con le credenziali di sicurezza temporanee che ottieni da (). Servizio di token di sicurezza AWS AWS STS A tale scopo, utilizzi l'ID della chiave di accesso e la chiave di accesso segreta che ricevi da AWS STS. Utilizzare l'ID della chiave di accesso e la chiave di accesso segreta nello stesso modo in cui si utilizzano le credenziali a lungo termine per firmare una richiesta. Aggiungete inoltre alla vostra richiesta API il token di sessione da cui ricevete AWS STS. Aggiungere il token della sessione a un'intestazione HTTP o a un parametro della stringa di query denominato `X-Amz-Security-Token`. Aggiungi il token di sessione all'intestazione HTTP o il parametro della stringa di query, ma non entrambi. Per ulteriori informazioni sulla firma delle richieste API HTTPS, consulta Signing AWS API Requests nel Riferimenti generali di AWS.

Ulteriori informazioni

Per ulteriori informazioni sull'utilizzo AWS STS con altri AWS servizi, consulta i seguenti link:

- Amazon S3. Consulta Esecuzione di richieste mediante le credenziali temporanee per gli utenti IAM o Esecuzione di richieste mediante le credenziali temporanee per gli utenti federati nella Guida per l'utente di Amazon Simple Storage Service.
- Amazon SNS Consulta Utilizzo di policy basate su identità con Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

- Amazon SQS Consulta Identity and Access Management in Amazon SQS nella Guida per gli sviluppatori di Amazon Simple Queue Service
- Amazon SimpleDB Consulta Utilizzo di credenziali di sicurezza temporanee nella Guida per gli sviluppatori di Amazon SimpleDB.

Autorizzazioni per le credenziali di sicurezza temporanee

Puoi usare Servizio di token di sicurezza AWS (AWS STS) per creare e fornire a utenti affidabili credenziali di sicurezza temporanee in grado di controllare l'accesso alle tue AWS risorse. Per ulteriori informazioni su AWS STS, vedereCredenziali di sicurezza temporanee in IAM. Le credenziali di sicurezza temporanee emesse da AWS STS sono valide fino al periodo di scadenza e non possono essere revocate. Tuttavia, le autorizzazioni assegnate a tali credenziali vengono valutate ogni volta che viene effettuata una richiesta che utilizza le credenziali stesse, pertanto è possibile ottenere l'effetto di revoca delle credenziali modificando i relativi diritti di accesso dopo che sono state emesse.

Negli argomenti seguenti si presuppone che tu abbia una conoscenza pratica delle AWS autorizzazioni e delle politiche. Per ulteriori informazioni su questi argomenti, consultare Gestione degli accessi per AWS risorse.

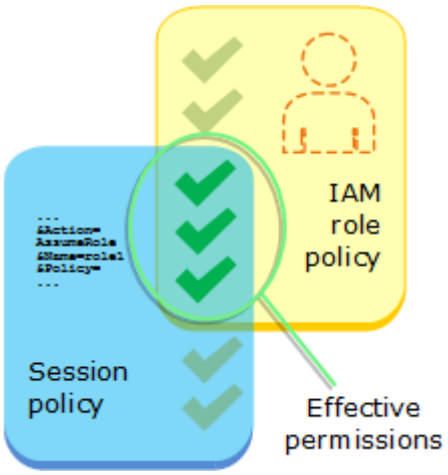
Argomenti

- Autorizzazioni per AssumeRole AssumeRoleWithSAML, e AssumeRoleWithWebIdentity
- Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti
- Autorizzazioni per GetFederationToken
- Autorizzazioni per GetSessionToken
- Disabilitazione delle autorizzazioni per le credenziali di sicurezza temporanee
- Concessione delle autorizzazioni per creare credenziali di sicurezza temporanee
- Concessione delle autorizzazioni per l'utilizzo di sessioni di console rafforzate con l'identità

Autorizzazioni per AssumeRole AssumeRoleWithSAML, e AssumeRoleWithWebIdentity

La policy di autorizzazione del ruolo assunto determina le autorizzazioni per le credenziali di sicurezza temporanee restituite da `AssumeRole`, `AssumeRoleWithSAML` e `AssumeRoleWithWebIdentity`. Puoi definire queste autorizzazioni quando crei o aggiorni il ruolo.

Facoltativamente, puoi trasferire le policy di sessione inline o gestite come parametri delle operazioni API `AssumeRole`, `AssumeRoleWithSAML` o `AssumeRoleWithWebIdentity`. Le policy di sessione limitano le autorizzazioni per la sessione con credenziali temporanee del ruolo. Le autorizzazioni della sessione risultante sono l'intersezione della policy basata sull'identità del ruolo e delle policy di sessione. È possibile utilizzare le credenziali temporanee del ruolo nelle successive chiamate AWS API per accedere alle risorse dell'account proprietario del ruolo. Non puoi utilizzare policy di sessione per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata su identità del ruolo che viene assunto. Per ulteriori informazioni su come AWS determina le autorizzazioni valide di un ruolo, consulta Logica di valutazione delle policy.



Le politiche allegate alle credenziali a cui è stata effettuata la chiamata originale non `AssumeRole` vengono valutate AWS quando si prende la decisione di autorizzazione «consentire» o «negare». L'utente rinuncia temporaneamente alle autorizzazioni originali a favore delle autorizzazioni assegnate dal ruolo assunto. Nel caso delle operazioni `AssumeRoleWithSAML` e dell'`AssumeRoleWithWebIdentity`API, non ci sono policy da valutare perché il chiamante dell'API non è un'identità. AWS

Esempio: assegnazione di autorizzazioni utilizzando AssumeRole

È possibile usare l'operazione API `AssumeRole` con diversi tipi di policy. Di seguito sono illustrati alcuni esempi.

Policy di autorizzazione di un ruolo

In questo esempio chiami l'operazione API `AssumeRole` senza specificare la policy di sessione nel parametro `Policy` facoltativo. Le autorizzazioni assegnate alle credenziali temporanee sono determinate dalla policy di autorizzazione del ruolo assunto. La policy di autorizzazioni di esempio seguente concede al ruolo l'autorizzazione per elencare tutti gli oggetti contenuti in un bucket S3

denominato `productionapp`. Consente inoltre al ruolo di ottenere, inserire ed eliminare gli oggetti all'interno del bucket.

Example Policy di autorizzazione di un ruolo di esempio

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::productionapp"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::productionapp/*"
    }
  ]
}
```

Policy di sessione passata come parametro

Immaginiamo di voler consentire a un utente di assumere lo stesso ruolo dell'esempio precedente. In questo caso però il ruolo della sessione deve avere l'autorizzazione solo per ottenere e mettere oggetti nel bucket S3 `productionapp`. Non desideri permettere all'utente di eliminare gli oggetti. Un metodo per raggiungere questo scopo consiste nel creare un nuovo ruolo e specificare le autorizzazioni desiderate nella policy di autorizzazione di tale ruolo. Un altro metodo per raggiungere lo scopo consiste nel chiamare l'API `AssumeRole` e includere una policy di sessione nel parametro `Policy` facoltativo come parte dell'operazione API. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. Le policy di sessione non possono essere utilizzate per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata sull'identità del ruolo che viene assunto. Per ulteriori informazioni sulle autorizzazioni della sessione del ruolo, consulta Policy di sessione.

Dopo aver recuperato le credenziali temporanee della nuova sessione, puoi passarle all'utente che deve disporre di tali autorizzazioni.

Immagina, ad esempio, che la policy seguente venga passata come parametro della chiamata API. L'utente che utilizza la sessione dispone di autorizzazioni per eseguire solo le seguenti azioni:

- Elencare tutti gli oggetti nel bucket `productionapp`.
- Ottenere e inserire gli oggetti nel bucket `productionapp`.

Nella policy di sessione seguente, l'autorizzazione `s3:DeleteObject` viene esclusa e alla sessione assunta non viene concessa l'autorizzazione `s3:DeleteObject`. La policy imposta il numero massimo di autorizzazioni per la sessione del ruolo, in modo che sostituisca qualsiasi policy di autorizzazione esistente su quel ruolo.

Example Esempio di policy di sessione passata con la chiamata API **`AssumeRole`**

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::productionapp"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::productionapp/*"
    }
  ]
}
```

Resource-based politica

Alcune AWS risorse supportano politiche basate sulle risorse e queste politiche forniscono un altro meccanismo per definire le autorizzazioni che influiscono sulle credenziali di sicurezza temporanee. Solo alcune risorse, come i bucket Amazon S3, gli argomenti Amazon SNS e le code Amazon SQS, supportano le policy basate sulle risorse. L'esempio seguente fornisce ulteriori informazioni sugli esempi precedenti, utilizzando un bucket S3, denominato `productionapp`. La policy seguente è collegata al bucket.

Quando colleghi la seguente policy basata su risorse al bucket `productionapp`, a tutti gli utenti viene negata l'autorizzazione per eliminare gli oggetti dal bucket. (Consulta l'elemento `Principal` nella policy). Ciò include tutti gli utenti che assumono il ruolo, anche se la policy di autorizzazione del ruolo concede l'autorizzazione `DeleteObject`. Un'istruzione `Deny` esplicita ha sempre la precedenza su un'istruzione `Allow`.

Example Esempio di policy di bucket

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Principal": {"AWS": "*"},
    "Effect": "Deny",
    "Action": "s3:DeleteObject",
    "Resource": "arn:aws:s3:::productionapp/*"
  }
}
```

Per ulteriori informazioni su come vengono combinati e valutati più tipi di policy, vedere. AWSLogica di valutazione delle policy

Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti

Un ruolo IAM è un oggetto in IAM a cui sono assegnate delle autorizzazioni. Quando assumi quel ruolo utilizzando un'identità IAM o un'identità esterna AWS, ricevi una sessione con le autorizzazioni assegnate al ruolo.

Quando esegui delle azioni in AWS, le informazioni sulla tua sessione possono essere registrate e AWS CloudTrail monitorate dall'amministratore dell'account. Gli amministratori possono configurare i

ruoli in modo da richiedere alle identità di inviare una stringa personalizzata che identifichi la persona o l'applicazione che esegue operazioni in AWS. Queste informazioni di identità vengono archiviate come identità di origine in AWS CloudTrail. Quando l'amministratore esamina l'attività in CloudTrail, può visualizzare le informazioni sull'identità di origine per determinare chi o cosa ha eseguito le azioni con le sessioni di ruolo presunte.

Una volta impostata, l'identità di origine è presente nelle richieste relative a qualsiasi AWS azione intrapresa durante la sessione di ruolo. Il valore impostato persiste quando un ruolo viene utilizzato per assumere un altro ruolo tramite l' AWS API AWS CLI or, nota come concatenamento di ruoli. Il valore impostato non può essere modificato durante la sessione del ruolo.

Gli amministratori possono configurare autorizzazioni granulari in base alla presenza o al valore dell'identità di origine per controllare ulteriormente AWS le azioni intraprese con ruoli condivisi. È possibile decidere se l'attributo dell'identità di origine può essere utilizzato, se è obbligatorio e quale valore può essere utilizzato.

Il modo in cui si utilizza l'identità di origine differisce dal nome della sessione di ruolo e dai tag di sessione in modo importante. Il valore dell'identità di origine non può essere modificato dopo l'impostazione e persiste per eventuali operazioni aggiuntive eseguite con la sessione del ruolo. Ecco come utilizzare i tag di sessione e il nome della sessione del ruolo:

- Tag di sessione: puoi passare i tag di sessione quando assumi un ruolo o federi un utente. I tag di sessione sono presenti quando si assume un ruolo. È possibile definire policy che utilizzano le chiavi condizionali sui tag per concedere autorizzazioni ai principali sulla base dei relativi tag. È quindi possibile utilizzare CloudTrail per visualizzare le richieste effettuate per assumere ruoli o federare utenti. Per ulteriori informazioni sui tag di sessione, consulta Passa i tag della sessione in AWS STS.
- Nome sessione del ruolo: puoi utilizzare la chiave di condizione `sts:RoleSessionName` in una policy di attendibilità del ruolo per richiedere che gli utenti forniscano un nome di sessione specifico quando assumono un ruolo. Il nome della sessione del ruolo può essere utilizzato per distinguere le sessioni del ruolo quando un ruolo viene utilizzato da principali diversi. Per saperne di più sul nome della sessione di ruolo, vedere sts: RoleSessionName.

Si consiglia di utilizzare l'identità di origine quando si desidera controllare l'identità che assume un ruolo. L'identità di origine è utile anche per estrarre CloudTrail i log per determinare chi ha utilizzato il ruolo per eseguire azioni.

Argomenti

- Configurazione per l'uso dell'identità di origine
- Cose da sapere sull'identità di origine
- Autorizzazioni necessarie per impostare l'identità di origine
- Specifica di un'identità di origine quando si assume un ruolo
- Usare l'identità di origine con AssumeRole
- Utilizzo dell'identità di origine con AssumeRoleWithSAML
- Utilizzo dell'identità di origine con AssumeRoleWithWebIdentity
- Controllo dell'accesso tramite le informazioni sull'identità di origine
- Visualizzazione dell'identità della fonte in CloudTrail

Configurazione per l'uso dell'identità di origine

Il modo in cui si imposta l'utilizzo dell'identità di origine dipende dal metodo utilizzato quando si assumono i ruoli. Ad esempio, gli utenti IAM potrebbero assumere ruoli direttamente utilizzando l'operazione `AssumeRole`. Se disponi di identità aziendali, note anche come identità della forza lavoro, potrebbero accedere alle tue AWS risorse utilizzando. `AssumeRoleWithSAML` Se gli utenti finali accedono alle tue applicazioni per dispositivi mobile o Web, potrebbero farlo utilizzando `AssumeRoleWithWebIdentity`. Di seguito è riportata una panoramica di alto livello del flusso di lavoro che consente di comprendere come impostare l'utilizzo delle informazioni sull'identità di origine nell'ambiente esistente.

1. Configurazione di utenti e ruoli di test: in un ambiente di preproduzione, configura utenti e ruoli di prova e configura le relative policy per consentire l'impostazione di un'identità di origine.

   Se utilizzi un provider di identità (IdP) per le identità federate, configura l'IdP per inviare un attributo utente a scelta per l'identità di origine nell'asserzione o nel token.
2. Assunzione del ruolo: verifica l'assunzione di ruoli e l'invio di un'identità di origine con gli utenti e i ruoli impostati per il test.
3. Revisione CloudTrail: esamina le informazioni sull'identità di origine per i ruoli di test nei registri. CloudTrail
4. Formazione degli utenti: dopo aver eseguito il test nell'ambiente di preproduzione, assicurati che gli utenti sappiano come inviare le informazioni sull'identità di origine, se necessario. Imposta una scadenza per il momento in cui sarà richiesto agli utenti di fornire un'identità di origine nell'ambiente di produzione.

5. Configurazione delle policy di produzione: configura le policy per l'ambiente di produzione e quindi aggiungile agli utenti e ai ruoli di produzione.
6. Monitora l'attività: monitora l'attività del tuo ruolo di produzione utilizzando CloudTrail i log.

Cose da sapere sull'identità di origine

Quando si utilizza un'identità di origine, tieni presente quanto segue.

- Le policy di attendibilità per tutti i ruoli connessi a un provider di identità (IdP) devono disporre dell'autorizzazione `sts:SetSourceIdentity`. Per i ruoli che non dispongono di questa autorizzazione nella policy di attendibilità, l'operazione `AssumeRole*` avrà esito negativo. Se non desideri aggiornare la policy di attendibilità del ruolo per ogni ruolo, puoi utilizzare un'istanza IdP separata per passare l'identità di origine. Quindi, aggiungere l'autorizzazione `sts:SetSourceIdentity` solo ai ruoli connessi all'IdP separato.
- Quando un'identità imposta un'identità di origine, la chiave `sts:SourceIdentity` è presente nella richiesta. Per le azioni successive intraprese durante la sessione di ruolo, la chiave `aws:SourceIdentity` è presente nella richiesta. AWS non controlla il valore dell'identità di origine nelle chiavi `sts:SourceIdentity` o `aws:SourceIdentity`. Se decidi di richiedere un'identità di origine, è necessario scegliere un attributo che sia fornito dagli utenti o dall'IdP. Per motivi di sicurezza, è necessario assicurarsi di poter controllare il modo in cui tali valori vengono forniti.
- Il valore dell'identità di origine deve avere una lunghezza compresa tra 2 e 256 caratteri, può contenere solo caratteri alfanumerici, trattini bassi e i seguenti caratteri:., + = @ - (trattino). Non è possibile utilizzare un valore che inizia con il testo **`aws:`**. Questo prefisso è riservato all'uso interno. AWS
- Le informazioni sull'identità di origine non vengono acquisite CloudTrail quando un AWS servizio o un ruolo collegato al servizio esegue un'azione per conto di un'identità federata o della forza lavoro.

> **Important**
>
> Non è possibile passare a un ruolo Console di gestione AWS che richiede l'impostazione di un'identità di origine quando si assume il ruolo. Per assumere tale ruolo, puoi utilizzare l' AWS API AWS CLI or per chiamare l'`AssumeRole`operazione e specificare il parametro di identità di origine.

Autorizzazioni necessarie per impostare l'identità di origine

Oltre a quella sull'operazione che corrisponde all'operazione API, è necessario disporre nella policy dell'autorizzazione per le seguenti operazioni:

```
sts:SetSourceIdentity
```

- Per specificare un'identità di origine, i principali (utenti e ruoli IAM) devono disporre delle autorizzazioni per`sts:SetSourceIdentity`. In qualità di amministratore, puoi configurarlo nella policy di attendibilità del ruolo e nella policy di autorizzazione del principale.
- Quando si assume un ruolo con un altro ruolo, secondo la funzione denominata concatenamento dei ruoli, le autorizzazioni per `sts:SetSourceIdentity` sono necessarie sia nella policy di autorizzazione del principale che assume il ruolo sia nella policy di attendibilità del ruolo del ruolo di destinazione. In caso contrario, l'operazione di assunzione del ruolo avrà esito negativo.
- Quando si utilizza l'identità di origine, le policy di attendibilità dei ruoli per tutti i ruoli connessi a un provider di identità (IdP) devono disporre dell'autorizzazione `sts:SetSourceIdentity`. L'operazione `AssumeRole*` avrà esito negativo per qualsiasi ruolo connesso a un IdP senza questa autorizzazione. Se non desideri aggiornare la policy di attendibilità del ruolo per ogni ruolo, puoi utilizzare un'istanza IdP separata per inviare l'identità di origine e aggiungere l'autorizzazione `sts:SetSourceIdentity` solo ai ruoli connessi all'IdP separato.
- Per impostare un'identità di origine oltre i limiti dell'account, è necessario includere l'autorizzazione `sts:SetSourceIdentity` in due posti. Deve trovarsi nella policy di autorizzazione del principale nell'account di origine e nella policy di attendibilità del ruolo del ruolo nell'account di destinazione. Questa operazione potrebbe essere necessaria, ad esempio, quando un ruolo viene utilizzato per assumere un ruolo in un altro account con il concatenamento dei ruoli.

Come amministratore dell'account, immagina di voler consentire all'utente IAM `DevUser` nel tuo account per assumere il `Developer_Role` nello stesso account. Tuttavia, si desidera consentire questa operazione solo se l'utente ha impostato l'identità di origine sul proprio nome utente IAM. La seguente policy può essere collegata a un utente IAM.

Example Esempio di politica basata sull'identità allegata a DevUser

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AssumeRole",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/Developer_Role"
    },
    {
      "Sid": "SetAwsUserNameAsSourceIdentity",
      "Effect": "Allow",
      "Action": "sts:SetSourceIdentity",
      "Resource": "arn:aws:iam::123456789012:role/Developer_Role",
      "Condition": {
        "StringLike": {
          "sts:SourceIdentity": "${aws:username}"
        }
      }
    }
  ]
}
```

Per applicare i valori di identità di origine accettabili, è possibile configurare la policy di attendibilità del ruolo riportata di seguito. La policy fornisce all'utente IAM le autorizzazioni `DevUser`per assumere il ruolo e impostare un'identità di origine. La chiave di condizione `sts:SourceIdentity` definisce il valore di identità di origine accettabile.

Example Esempio di policy di attendibilità dei ruoli dell'identità di origine

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDevUserAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/DevUser"
      },
      "Action": [
```

```
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringEquals": {
          "sts:SourceIdentity": "DevUser"
        }
      }
    }
  ]
}
```

Utilizzando le credenziali dell'utente IAM`DevUser`, l'utente tenta di presupporre l'`DeveloperRole`utilizzo della seguente richiesta. AWS CLI

Example Esempio di richiesta AssumeRole CLI

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/Developer_Role \
--role-session-name Dev-project \
--source-identity DevUser \
```

Quando AWS valuta la richiesta, il contesto della richiesta contiene il `sts:SourceIdentity` valore di. `DevUser`

Specifica di un'identità di origine quando si assume un ruolo

È possibile specificare un'identità di origine quando si utilizza una delle operazioni AWS STS `AssumeRole*` API per ottenere credenziali di sicurezza temporanee per un ruolo. L'operazione API che usi è diversa a seconda del caso d'uso.

Ad esempio, se utilizzi i ruoli IAM per consentire agli utenti IAM di accedere a AWS risorse a cui normalmente non hanno accesso, potresti utilizzare l'`AssumeRole`operazione. Se utilizzi la federazione delle identità aziendali per gestire gli utenti della forza lavoro, puoi utilizzare l'operazione `AssumeRoleWithSAML`. Se utilizzi la federazione OIDC per consentire agli utenti finali di accedere alle applicazioni mobili o Web, puoi utilizzare l'operazione `AssumeRoleWithWebIdentity`. Nelle sezioni seguenti viene illustrato come utilizzare l'identità di origine per ogni operazione. Per ulteriori informazioni sugli scenari comuni per le credenziali temporanee, consulta Scenari comuni per le credenziali temporanee.

Usare l'identità di origine con AssumeRole

L'`AssumeRole`operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere alle AWS risorse. È possibile utilizzare le credenziali dell'utente o del ruolo IAM per chiamare `AssumeRole`. Per passare l'identità di origine assumendo un ruolo, utilizzate l'`--source-identity` AWS CLI opzione o il parametro `SourceIdentity` AWS API. L'esempio seguente illustra come specificare l'identità di origine utilizzando la AWS CLI.

Example Esempio di richiesta AssumeRole CLI

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/developer \
--role-session-name Audit \
--source-identity Admin \
```

Utilizzo dell'identità di origine con AssumeRoleWithSAML

Il principale che chiama l'`AssumeRoleWithSAML`operazione viene autenticato tramite SAML-based federazione. Questa operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere AWS alle risorse. Per ulteriori informazioni sull'utilizzo SAML-based della federazione per Console di gestione AWS l'accesso, vedereConsentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS. Per dettagli sul AWS CLI AWS nostro accesso alle API, consultaFederazione SAML 2.0. Per un tutorial sulla configurazione della federazione SAML per gli utenti di Active Directory, consulta AWS Federated Authentication with Active Directory Federation Services (ADFS) nel AWS Security Blog.

In qualità di amministratore, puoi consentire ai membri della tua directory aziendale di unirsi all'utilizzo dell'operazione. AWS AWS STS `AssumeRoleWithSAML` A tale scopo, è necessario completare le seguenti attività:

1. Configura un provider SAML nella tua organizzazione.
2. Creazione di un provider SAML in IAM.
3. Configura un ruolo e le relative autorizzazioni AWS per i tuoi presidi federati SAML.
4. Fine della configurazione del provider di identità SAML e creazione di asserzioni per la risposta di autenticazione SAML

Per impostare un attributo SAML per l'identità di origine, includi l'elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/SourceIdentity`.

Utilizza l'elemento `AttributeValue` per specificare il valore dell'identità di origine. Ad esempio, si supponga di voler passare i seguenti attributi di identità come identità di origine:

`SourceIdentity:DiegoRamirez`

Per passare questi attributi, includi i seguenti elementi nell'asserzione SAML.

Example Esempio di frammento di un'asserzione SAML

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/SourceIdentity">
<AttributeValue>DiegoRamirez</AttributeValue>
</Attribute>
```

Utilizzo dell'identità di origine con AssumeRoleWithWebIdentity

Il principale che richiama l'operazione `AssumeRoleWithWebIdentity` viene autenticato utilizzando la federazione compatibile con OpenID Connect (OIDC). Questa operazione restituisce un insieme di credenziali temporanee che è possibile utilizzare per accedere alle risorse di AWS . Per ulteriori informazioni sull'utilizzo della federazione OIDC per l'accesso alla Console di gestione AWS , consulta Federazione OIDC.

Per passare l'identità di origine da OpenID Connect (OIDC), è necessario includere l'identità di origine nel Token Web JSON (JWT). Includi l'identità di origine nel namespace `https://aws.amazon.com/` `source_identity` nel token quando si invia la richiesta `AssumeRoleWithWebIdentity`. Per ulteriori informazioni sui token e le registrazioni OIDC, consultare Utilizzo di token con pool di utenti nella Guida per gli sviluppatori Amazon Cognito .

Ad esempio, il seguente JWT decodificato è un token utilizzato per chiamare `AssumeRoleWithWebIdentity` con l'identità di origine `Admin`.

Example Esempio di token Web JSON decodificato

```
{
    "sub": "john",
    "aud": "ac_oic_client",
    "jti": "ZYUCeRMQVtqHypVPWAN3VB",
    "iss": "https://xyz.com",
    "iat": 1566583294,
    "exp": 1566583354,
    "auth_time": 1566583292,
```

```
    "https://aws.amazon.com/source_identity":"Admin"
}
```

Controllo dell'accesso tramite le informazioni sull'identità di origine

Quando viene inizialmente impostata un'identità di origine, la SourceIdentity chiave sts: è presente nella richiesta. Dopo aver impostato un'identità di origine, la SourceIdentity chiave aws: è presente in tutte le richieste successive effettuate durante la sessione di ruolo. In qualità di amministratore, puoi scrivere politiche che concedono l'autorizzazione condizionale per eseguire AWS azioni in base all'esistenza o al valore dell'attributo di identità di origine.

Immagina di voler richiedere ai tuoi sviluppatori di impostare un'identità di origine per assumere un ruolo fondamentale con il permesso di scrivere su una AWS risorsa critica per la produzione. Immagina inoltre di concedere AWS l'accesso alle identità della tua forza lavoro utilizzando. `AssumeRoleWithSAML` Desideri solo che gli sviluppatori senior Saanvi e Diego abbiano accesso al ruolo, in modo che possano creare le seguenti policy di attendibilità per il ruolo.

Example Esempio di policy di attendibilità dei ruoli dell'identità di origine (SAML)

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "SAMLProviderAssumeRoleWithSAML",
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:saml-provider/name-of-identity-
provider"
      },
      "Action": [
        "sts:AssumeRoleWithSAML"
      ],
      "Condition": {
        "StringEquals": {
          "SAML:aud": "https://region-code.signin.aws.amazon.com/saml"
        }
      }
    },
    {
```

```
      "Sid": "SetSourceIdentitySrEngs",
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:saml-provider/name-of-identity-
provider"
      },
      "Action": [
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringLike": {
          "sts:SourceIdentity": [
            "Saanvi",
            "Diego"
          ]
        }
      }
    }
  ]
}
```

La policy di attendibilità contiene una condizione per `sts:SourceIdentity` che richiede un'identità di origine impostata su Saanvi o Diego per assumere il ruolo critico.

In alternativa, se si utilizza un provider OIDC per la federazione e gli utenti sono autenticati con `AssumeRoleWithWebIdentity`, la tua policy di attendibilità del ruolo potrebbe avere un aspetto simile al seguente.

Example Esempio di policy di attendibilità dei ruoli dell'identità di origine (provider OIDC)

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/server.example.com"
      },
      "Action": [
```

```
        "sts:AssumeRoleWithWebIdentity",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringEquals": {
          "server.example.com:aud": "oidc-audience-id"
        },
        "StringLike": {
          "sts:SourceIdentity": [
            "Saanvi",
            "Diego"
          ]
        }
      }
    }
  ]
}
```

Concatenamento dei ruoli e requisiti tra account

Immagina di voler consentire agli utenti che hanno assunto il ruolo `CriticalRole` di assumere un ruolo `CriticalRole_2` in un altro account. Le credenziali della sessione del ruolo ottenute per assumere `CriticalRole` sono utilizzate per il concatenamento dei ruoli per un secondo ruolo, `CriticalRole_2`, in un account diverso. Il ruolo viene assunto oltre un limite di account. Pertanto, l'autorizzazione `sts:SetSourceIdentity` deve essere concessa in entrambe le policy di autorizzazione su `CriticalRole` e nella policy di attendibilità del ruolo su `CriticalRole_2`.

Example Esempio di politica delle autorizzazioni su CriticalRole

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AssumeRoleAndSetSourceIdentity",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
```

```
      "Resource": "arn:aws:iam::222222222222:role/CriticalRole_2"
    }
  ]
}
```

Per proteggere l'impostazione dell'identità di origine attraverso il limite dell'account, la seguente policy di attendibilità del ruolo considera attendibile solo il principale del ruolo per `CriticalRole` per impostare l'identità di origine.

Example Esempio di politica di fiducia dei ruoli su _2 CriticalRole

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:role/CriticalRole"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringLike": {
          "aws:SourceIdentity": ["Saanvi","Diego"]
        }
      }
    }
  ]
}
```

L'utente effettua la seguente chiamata utilizzando le credenziali della sessione di ruolo ottenute assumendo. CriticalRole L'identità di origine è stata impostata durante l'assunzione di CriticalRole, quindi non è necessario impostarla nuovamente in modo esplicito. Se l'utente prova a impostare un'identità di origine diversa dal set di valori quando è stato assunto `CriticalRole`, la richiesta di assumere il ruolo verrà rifiutata.

Example Esempio AssumeRole di richiesta CLI

```
aws sts assume-role \
--role-arn arn:aws:iam::222222222222:role/CriticalRole_2 \
--role-session-name Audit \
```

Quando il principale chiamante assume il ruolo, l'identità di origine nella richiesta persiste dalla prima sessione del ruolo assunto. Pertanto, entrambe le chiavi `aws:SourceIdentity` e `sts:SourceIdentity` sono presenti nel contesto della richiesta.

Visualizzazione dell'identità della fonte in CloudTrail

È possibile CloudTrail utilizzarlo per visualizzare le richieste fatte per assumere ruoli o federare utenti. È inoltre possibile visualizzare le richieste di ruoli o utenti per eseguire operazioni in AWS. Il file di CloudTrail registro include informazioni sull'identità di origine impostata per la sessione utente con ruolo presunto o federato. Per ulteriori informazioni, consulta Registrazione di IAM e AWS STS Chiamate API con AWS CloudTrail

Ad esempio, supponiamo che un utente effettui una AWS STS `AssumeRole` richiesta e imposti un'identità di origine. Puoi trovare le `sourceIdentity` informazioni nella `requestParameters` chiave del tuo CloudTrail registro.

Example Esempio di sezione RequestParameters in un AWS CloudTrail log

```
"eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSAccount",
        "principalId": "AIDAJ45Q7YFFAREXAMPLE",
        "accountId": "111122223333"
    },
    "eventTime": "2020-04-02T18:20:53Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.64",
    "userAgent": "aws-cli/1.16.96 Python/3.6.0 Windows/10 botocore/1.12.86",
    "requestParameters": {
        "roleArn": "arn:aws:iam::123456789012:role/DevRole",
        "roleSessionName": "Dev1",
        "sourceIdentity": "source-identity-value-set"
    }
```

Se l'utente utilizza la sessione di ruolo presunta per eseguire un'azione, le informazioni sull'identità di origine sono presenti nella `userIdentity` chiave del CloudTrail registro.

Example Esempio di chiave UserIdentity in un AWS CloudTrail log

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAJ45Q7YFFAREXAMPLE:Dev1",
    "arn": "arn:aws:sts::123456789012:assumed-role/DevRole/Dev1",
    "accountId": "123456789012",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAJ45Q7YFFAREXAMPLE",
        "arn": "arn:aws:iam::123456789012:role/DevRole",
        "accountId": "123456789012",
        "userName": "DevRole"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-02-21T23:46:28Z"
      },
      "sourceIdentity": "source-identity-value-present"
    }
  }
}
```

Per vedere esempi di eventi AWS STS API nei CloudTrail log, consultaEsempi di eventi dell'API IAM nel registro CloudTrail. Per maggiori dettagli sulle informazioni contenute nei file di CloudTrail registro, consulta CloudTrail Event Reference nella Guida per l'AWS CloudTrail utente.

Autorizzazioni per GetFederationToken

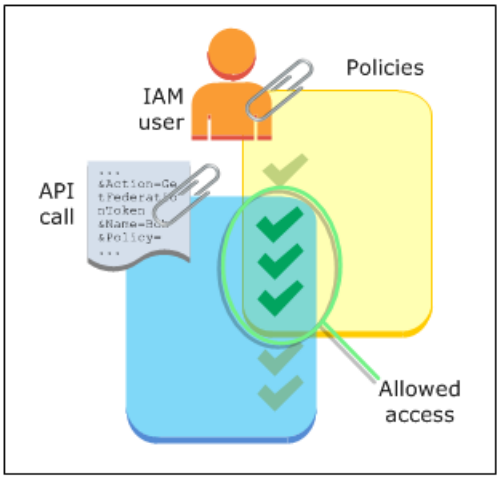
L'operazione `GetFederationToken` viene chiamata da un utente IAM e restituisce le credenziali temporanee per tale utente. Questa operazione consolida l'utente. Le autorizzazioni assegnate a una sessione utente AWS STS federata sono definite in una delle due posizioni seguenti:

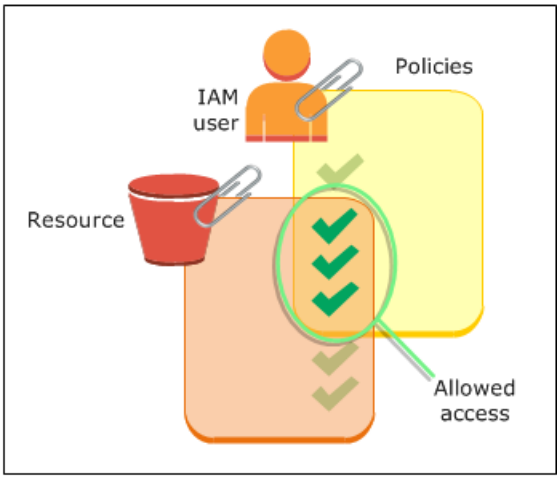
- Le policy di sessione passate come un parametro della chiamata API `GetFederationToken`. (Questo è più comune).
- Una politica basata sulle risorse che nomina esplicitamente la sessione utente AWS STS federata nell'elemento della politica. `Principal` (Questo è meno comune).

Le policy di sessione sono policy avanzate che vengono passate come parametri quando si crea una sessione temporanea a livello di programma. Quando si crea una sessione utente AWS STS federata e si passano i criteri di sessione, le autorizzazioni della sessione risultanti sono l'intersezione tra la politica basata sull'identità dell'utente e le politiche di sessione. Non puoi utilizzare la policy di sessione per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata su identità dell'utente che viene federato.

Nella maggior parte dei casi, se non si passa una policy con la chiamata API `GetFederationToken`, le credenziali di sicurezza temporanee risultanti non dispongono di autorizzazioni. Tuttavia, una policy basata sulle risorse è in grado di fornire ulteriori autorizzazioni per la sessione. Puoi accedere a una risorsa con una policy basata sulle risorse che specifica la sessione come l'entità principale consentita.

Le seguenti immagini mostrano una rappresentazione visiva di come le policy interagiscono per determinare le autorizzazioni per le credenziali di sicurezza provvisorie restituite da una chiamata a `GetFederationToken`.





Esempio: assegnazione di autorizzazioni tramite GetFederationToken

È possibile utilizzare l'operazione API `GetFederationToken` con diversi tipi di policy. Di seguito sono illustrati alcuni esempi.

Policy collegata all'utente IAM

In questo esempio, disponi di un'applicazione client basata sul browser che si avvale di due servizi Web di backend. Un servizio di backend è il tuo server di autenticazione che utilizza un sistema di identità per autenticare l'applicazione client. L'altro servizio di backend è un servizio AWS che fornisce alcune delle funzionalità dell'applicazione client. L'applicazione client viene autenticata mediante il tuo server, il quale crea o recupera la policy di autorizzazione appropriata. Il server chiama l'API `GetFederationToken` per ottenere le credenziali di sicurezza provvisorie e restituisce tali credenziali all'applicazione client. L'applicazione client può quindi effettuare richieste direttamente al AWS servizio con le credenziali di sicurezza temporanee. Questa architettura consente all'applicazione client di effettuare AWS richieste senza incorporare credenziali a lungo termine AWS .

Il tuo server di autenticazione chiama l'API `GetFederationToken` con le credenziali di sicurezza a lungo termine di un utente IAM denominato `token-app`. Tuttavia, le credenziali utente IAM a lungo termine rimangono nel server e non vengono mai distribuite al client. La seguente policy di esempio è allegata all'utente `token-app` IAM e definisce il set più ampio di autorizzazioni di cui avranno bisogno gli utenti AWS STS federati (client). Tieni presente che l'`sts:GetFederationToken`autorizzazione è necessaria affinché il servizio di autenticazione ottenga credenziali di sicurezza temporanee per gli utenti federati. AWS STS

Example Esempio di policy allegata all'app **`token`** utente IAM che chiama **`GetFederationToken`**

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:GetFederationToken",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "dynamodb:ListTables",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
```

```
      "Action": "sqs:ReceiveMessage",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "sns:ListSubscriptions",
      "Resource": "*"
    }
  ]
}
```

La policy precedente concede diverse autorizzazioni all'utente IAM. Tuttavia, questa politica da sola non concede alcuna autorizzazione all'utente federato AWS STS . Se questo utente IAM chiama `GetFederationToken` e non passa una policy come parametro della chiamata API, l'utente AWS STS federato risultante non dispone di autorizzazioni valide.

Policy di sessione passata come parametro

Il modo più comune per garantire che all'utente AWS STS federato venga assegnata l'autorizzazione appropriata consiste nel trasmettere le politiche di sessione nella `GetFederationToken` chiamata API. Sulla base dell'esempio precedente, immagina che `GetFederationToken` venga chiamato con le credenziali dell'utente IAM `token-app`. Quindi, immagina che la policy di sessione seguente venga passata come un parametro della chiamata API. L'utente federato AWS STS risultante dispone dell'autorizzazione per elencare i contenuti del bucket Amazon S3 denominato `productionapp`. L'utente non può eseguire le operazioni `GetObject`, `PutObject` e `DeleteObject` di Amazon S3 su elementi nel bucket `productionapp`.

All'utente federato vengono assegnate queste autorizzazioni perché le autorizzazioni sono l'intersezione delle policy utente IAM e delle policy di sessione che vengono passate.

L'utente AWS STS federato non poteva eseguire azioni in Amazon SNS, Amazon SQS, Amazon DynamoDB o in nessun bucket S3 tranne. `productionapp` Queste operazioni sono rifiutate anche se tali autorizzazioni sono concesse all'utente IAM associato alla chiamata `GetFederationToken`.

Example Esempio di policy di sessione passata come parametro della chiamata API **`GetFederationToken`**.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::productionapp"]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": ["arn:aws:s3:::productionapp/*"]
    }
  ]
}
```

Resource-based politiche

Alcune AWS risorse supportano politiche basate sulle risorse e queste politiche forniscono un altro meccanismo per concedere le autorizzazioni direttamente a un utente federato. AWS STS Solo alcuni servizi supportano politiche basate sulle risorse AWS . Ad esempio, Amazon S3 ha i bucket, Amazon SNS ha gli argomenti e Amazon SQS ha le code, tutti elementi ai quali è possibile collegare le policy. Per un elenco di tutti i servizi che supportano le politiche basate sulle risorse, consulta AWS servizi supportati da IAM e consulta la colonna "Resource-basedpolicy» delle tabelle. È possibile utilizzare le politiche basate sulle risorse per assegnare le autorizzazioni direttamente a un utente federato. AWS STS A tale scopo, specifica l'Amazon Resource Name (ARN) dell' AWS STS utente federato nell'elemento della policy basata `Principal` sulle risorse. Ciò viene illustrato nell'esempio seguente espandendo gli esempi precedenti e utilizzando un bucket S3 denominato `productionapp`.

La policy basata sulle risorse riportata di seguito è collegata al bucket. Questa policy sui bucket consente a un utente AWS STS federato di nome Carol di accedere al bucket. Quando la

policy di esempio descritta in precedenza viene allegata all'utente `token-app` IAM, l'utente AWS STS federato di nome Carol è autorizzato a eseguire le azioni e `s3:DeleteObject` le azioni sul `s3:GetObject` `s3:PutObject` bucket denominato. `productionapp` Questo vale anche quando nessuna policy di sessione viene passata come parametro della chiamata API `GetFederationToken`. Questo perché in questo caso all'utente AWS STS federato di nome Carol sono state concesse esplicitamente le autorizzazioni dalla seguente politica basata sulle risorse.

Ricorda che a un utente AWS STS federato vengono concesse le autorizzazioni solo quando tali autorizzazioni sono concesse esplicitamente sia all'utente IAM che all'utente federato. AWS STS Possono anche essere concesse (all'interno dell'account) da una policy basata sulle risorse che nomina esplicitamente l'utente AWS STS federato nell'`Principal`elemento della policy, come nell'esempio seguente.

Example Esempio di policy del bucket che consente l'accesso all'utente federato

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Principal": {
            "AWS": "arn:aws:sts::111122223333:federated-user/Carol"
        },
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:PutObject",
            "s3:DeleteObject"
        ],
        "Resource": [
            "arn:aws:s3:::productionapp/*"
        ]
    }
}
```

Per ulteriori informazioni su come vengono valutate le policy, consulta la sezione Logica di valutazione delle policy.

Autorizzazioni per GetSessionToken

La principale occasione per chiamare l'operazione API `GetSessionToken` o il comando CLI `get-session-token` è quando un utente deve essere autenticato tramite Multi-Factor Authentication (MFA). È possibile scrivere una policy che permette determinate operazioni solo quando tali operazioni vengono richieste da un utente autenticato con MFA. Per superare i controlli di autorizzazione MFA, un utente deve prima chiamare `GetSessionToken` e includere i parametri facoltativi `SerialNumber` e `TokenCode`. Se l'utente è stato autenticato con un dispositivo MFA, le credenziali restituite dall'operazione API `GetSessionToken` includono il contesto MFA. Tale contesto indica che l'utente viene autenticato con MFA ed è autorizzato per le operazioni API che richiedono l'autenticazione MFA.

Autorizzazioni richieste per GetSessionToken

Non è richiesta all'utente alcuna autorizzazione per ottenere un token di sessione. Lo scopo dell'operazione `GetSessionToken` è autenticare l'utente tramite MFA. Non è possibile utilizzare le policy per controllare le operazioni di autenticazione.

Per concedere le autorizzazioni necessarie per eseguire la maggior parte AWS delle operazioni, è necessario aggiungere l'azione con lo stesso nome a una policy. Ad esempio, per creare un utente, occorre utilizzare l'operazione API `CreateUser`, il comando della CLI `create-user` o la Console di gestione AWS. Per eseguire queste operazioni, è necessario disporre di una policy che consenta di accedere all'operazione `CreateUser`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateUser",
            "Resource": "*"
        }
    ]
}
```

È possibile includere l'operazione `GetSessionToken` nella policy, ma questo non incide sulla capacità di un utente di eseguire l'operazione `GetSessionToken`.

Autorizzazioni concesse da GetSessionToken

Se la chiamata a `GetSessionToken` viene eseguita con le credenziali di un utente IAM, le credenziali di sicurezza temporanee avranno le stesse autorizzazioni dell'utente IAM. Allo stesso modo, se `GetSessionToken` viene chiamata con Utente root dell'account AWS credenziali, le credenziali di sicurezza temporanee dispongono dei permessi di utente root.

> **Note**
>
> È consigliabile non chiamare `GetSessionToken` con credenziali dell'utente root Segui invece le best practice e crea utenti IAM con le autorizzazioni necessarie. Quindi usa questi utenti IAM per l'interazione quotidiana con. AWS

Le credenziali temporanee ottenute chiamando `GetSessionToken` hanno le funzionalità e le limitazioni seguenti:

- Puoi utilizzare le credenziali per accedere a passando le credenziali all' Console di gestione AWS endpoint Single Sign-On della federazione all'indirizzo. `https://signin.aws.amazon.com/federation` Per ulteriori informazioni, consulta Abilita l'accesso personalizzato del broker di identità al AWS console.
- Non puoi utilizzare le credenziali per richiamare le operazioni API IAM o AWS STS . È possibile utilizzarle per chiamare le operazioni API per altri servizi. AWS

Per un confronto tra questa operazione API e i relativi limiti e funzionalità con le altre operazioni API che creano credenziali di sicurezza temporanee, consulta Confronta le AWS STS credenziali

Per ulteriori informazioni sull'accesso API protetto da MFA con `GetSessionToken`, consulta Accesso sicuro alle API con MFA.

Disabilitazione delle autorizzazioni per le credenziali di sicurezza temporanee

Le credenziali di sicurezza provvisorie sono valide finché non scadono. Queste credenziali sono valide per la durata specificata, da 900 secondi (15 minuti) a un massimo di 129.600 secondi (36 ore). La durata predefinita della sessione è di 43.200 secondi (12 ore). Puoi revocare queste credenziali, ma devi anche modificare le autorizzazioni per il ruolo o l'utente IAM per bloccare l'uso di credenziali compromesse che consentirebbero attività dannose per l'account. Le autorizzazioni assegnate alle credenziali di sicurezza temporanee vengono valutate ogni volta che vengono

utilizzate per effettuare una richiesta. AWS Una volta rimosse tutte le autorizzazioni dalle credenziali, le AWS richieste che le utilizzano hanno esito negativo.

Potrebbero essere necessari alcuni minuti prima che gli aggiornamenti delle politicy siano applicati. Per le sessioni del ruolo IAM, puoi revocare le credenziali di sicurezza temporanee del ruolo per obbligare tutti gli utenti che assumono il ruolo a riautenticarsi e richiedere nuove credenziali. Per ulteriori informazioni, consulta Revoca delle credenziali di sicurezza provvisorie del ruolo.

Non è possibile modificare le autorizzazioni per un. Utente root dell'account AWS Analogamente, non puoi modificare le autorizzazioni relative alle credenziali di sicurezza temporanee create richiamando `GetFederationToken` o `GetSessionToken` mentre sei collegato come utente root. Per questo motivo, consigliamo di non effettuare la chiamata a `GetFederationToken` o `GetSessionToken` come utente root.

Per le procedure su come modificare le autorizzazioni per un ruolo IAM, consulta Modificare le autorizzazioni per un utente IAM.

Per le procedure su come modificare le autorizzazioni per un ruolo IAM, consulta Aggiornamento delle autorizzazioni per un ruolo.

> **Important**
>
> Non è possibile modificare i ruoli in IAM creati dai set di autorizzazioni del Centro identità IAM. È necessario revocare la sessione attiva del set di autorizzazioni per un utente nel Centro identità IAM. Per ulteriori informazioni, consulta Revocare le sessioni attive di un ruolo IAM create dai set di autorizzazioni nella Guida per l'utente del Centro identità IAM.

Argomenti

- Negare l'accesso a tutte le sessioni del ruolo IAM associate a un ruolo
- Negare l'accesso a una sessione specifica del ruolo IAM
- Negare l'accesso alle sessioni delle credenziali di sicurezza temporanee con chiavi di contesto della condizione
- Negare l'accesso a uno specifico principale con policy basate sulle risorse

Negare l'accesso a tutte le sessioni del ruolo IAM associate a un ruolo

Questa procedura nega le autorizzazioni a tutte le sessioni del ruolo IAM associate a un ruolo. Usa questo approccio quando hai il dubbio che sia stato effettuato un accesso sospetto da:

- Principali di un altro account utilizzando l'accesso multi-account
- Identità di utenti esterni con autorizzazioni per accedere alle AWS risorse del tuo account
- Utenti che sono stati autenticati in un'applicazione Web o mobile con un provider OIDC

Per modificare o rimuovere le autorizzazioni assegnate alle credenziali di sicurezza provvisorie ottenute richiamando `AssumeRole`, `AssumeRoleWithSAML` o `AssumeRoleWithWebIdentity`, `GetFederationToken` o `GetSessionToken`, puoi modificare o eliminare la policy basata sulle identità che definisce le autorizzazioni per il ruolo.

> **Important**
>
> Se esiste una policy basata sulle risorse che consente l'accesso principale, devi anche aggiungere un rifiuto esplicito per quella risorsa. Per informazioni dettagliate, vedi Negare l'accesso a uno specifico principale con policy basate sulle risorse.

Per negare l'accesso a tutte le sessioni del ruolo IAM associate a un ruolo

1. Accedi Console di gestione AWS e apri la console IAM.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Scegli il nome del ruolo da modificare. Puoi utilizzare la casella di ricerca per filtrare l'elenco.
4. Scegli la scheda Autorizzazioni.
5. Seleziona la policy pertinente da modificare. Prima di modificare una policy gestita dal cliente, consulta la scheda Entità collegate per evitare di interrompere l'accesso ad altre identità che potrebbero avere la stessa policy associata.
6. Scegli la scheda JSON e aggiorna la policy per negare tutte le risorse e le azioni.

> **Note**
>
> Queste autorizzazioni sono le stesse della policy AWS gestita AWSDenyAll. Puoi allegare questa policy AWS gestita a qualsiasi utente o ruolo IAM a cui desideri negare qualsiasi accesso.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DenyAll",
            "Effect": "Deny",
            "Action": [
                "*"
            ],
            "Resource": "*"
        }
    ]
}
```

7. Nella pagina Review (Esamina) controllare Summary (Riepilogo) e selezionare Save changes (Salva modifiche) per salvare.

Quando si modifica o si elimina la policy, le modifiche riguardano le autorizzazioni di tutte le credenziali di sicurezza provvisorie associate a tale ruolo, tra cui credenziali che sono state rilasciate prima di aver modificato la policy di autorizzazione del ruolo.

Dopo aver aggiornato la policy, è possibile revocare le credenziali di sicurezza temporanee del ruolo per revocare immediatamente tutte le autorizzazioni alle credenziali emesse per il ruolo.

Negare l'accesso a una sessione specifica del ruolo IAM

Quando aggiorni i ruoli IAM con una policy di negazione totale o elimini completamente il ruolo, tutti gli utenti che hanno accesso al ruolo vengono scollegati. Puoi negare l'accesso senza influire sulle autorizzazioni di tutte le altre sessioni associate al ruolo.

Ai `Principal` possono essere negate le autorizzazioni usando chiavi di contesto delle condizioni o policy basate sulle risorse.

> **Tip**
>
> Puoi trovare gli utenti ARNs federati utilizzando AWS CloudTrail i log. Per ulteriori informazioni, consulta Come identificare facilmente gli utenti federati utilizzando. AWS CloudTrail

Negare l'accesso alle sessioni delle credenziali di sicurezza temporanee con chiavi di contesto della condizione

Puoi usare le chiavi di contesto della condizione nelle policy basate sulle identità in situazioni in cui vuoi negare l'accesso a sessioni specifiche con credenziali di sicurezza temporanee senza compromettere le autorizzazioni dell'utente IAM o del ruolo che ha creato le credenziali. Per i ruoli IAM, dopo aver aggiornato la policy, puoi anche revocare le credenziali di sicurezza temporanee del ruolo per revocare immediatamente tutte le credenziali emesse.

Per ulteriori informazioni su queste chiavi di contesto della condizione, consulta AWS chiavi di contesto delle condizioni globali.

leggi: PrincipalArn

Puoi usare la chiave di contesto della condizione era: PrincipalArn in una policy basata sull'identità per negare l'accesso a uno specifico principale tramite il relativo nome della risorsa Amazon (ARN). A tale scopo, specificate l'ARN dell'utente, del ruolo AWS STS o della sessione utente federata IAM a cui sono associate le credenziali di sicurezza temporanee nell'elemento Condition di una policy.

Per negare l'accesso a uno specifico principale tramite il relativo ARN

1. Nel riquadro di navigazione della console IAM, scegli Utenti o Ruoli.
2. Scegli il nome dell'utente IAM o del ruolo da modificare. Puoi utilizzare la casella di ricerca per filtrare l'elenco.
3. Scegli la scheda Autorizzazioni.
4. Seleziona la policy pertinente da modificare. Prima di modificare una policy gestita dal cliente, consulta la scheda Entità collegate per evitare di interrompere l'accesso ad altre identità che potrebbero avere la stessa policy associata.

5. Scegli la scheda JSON e aggiungi un'istruzione di negazione per l'ARN principale, come mostrato nell'esempio seguente.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:PrincipalArn": [
            "arn:aws:iam::222222222222:role/ROLENAME",
            "arn:aws:iam::222222222222:user/USERNAME",
            "arn:aws:iam::222222222222:federated-user/USERNAME"
          ]
        }
      }
    }
  ]
}
```

6. Nella pagina Review (Esamina) controllare Summary (Riepilogo) e selezionare Save changes (Salva modifiche) per salvare.

Leggi: SourceIdentity

Puoi usare la chiave di contesto della condizione era: SourceIdentity in una policy basata sull'identità per negare l'accesso a un'identità di origine specifica associata a una sessione di ruolo IAM. Ciò vale a condizione che la sessione del ruolo sia stata emessa impostando il parametro di `SourceIdentity` richiesta quando il principale ha assunto un ruolo utilizzando qualsiasi AWS STS `assume-role` comando* CLI AWS STS `AssumeRole` o* operazioni API. A tale scopo, devi specificare l'identità di origine a cui sono associate le credenziali di sicurezza temporanee nell'elemento `Condition` di una policy.

A differenza della chiave di contesto sts:RoleSessionName, dopo aver impostato l'identità di origine, il valore non può essere modificato. La chiave `aws:SourceIdentity` è presente nel contesto della

richiesta di tutte le operazioni intraprese dal ruolo. L'identità di origine persiste nelle sessioni di ruolo successive quando si utilizzano le credenziali di sessione per assumere un altro ruolo. L'assunzione di un ruolo partendo da un altro si chiama concatenamento del ruolo.

La seguente policy mostra un esempio di come puoi negare l'accesso a sessioni con credenziali di sicurezza temporanee utilizzando la chiave di contesto della condizione.`aws:SourceIdentity`. Se specifichi l'identità di origine associata a una sessione del ruolo, verranno negate le sessioni di ruolo con l'identità di origine specificata senza influire sulle autorizzazioni del ruolo che ha creato le credenziali. Per questo esempio, l'identità di origine impostata dal principale al momento dell'emissione della sessione di ruolo è `nikki_wolf@example.com`. Qualsiasi richiesta effettuata da una sessione di ruolo con l'identità di origine `nikki_wolf@example.com` verrà rifiutata perché l'identità di origine è inclusa nella condizione della policy e la policy Effect è impostata su `Deny`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:SourceIdentity": [
            "nikki_wolf@example.com",
            "<source identity value>"
          ]
        }
      }
    }
  ]
}
```

aws:userid

Puoi usare la chiave di contesto della condizione aws:userid in una policy basata sulle identità per negare l'accesso a tutte o a specifiche sessioni con credenziali di sicurezza temporanee associate all'utente o al ruolo IAM. A tale scopo, è necessario specificare l'identificatore univoco (ID) dell'utente,

del ruolo o della sessione utente AWS STS federata IAM a cui sono associate le credenziali di sicurezza temporanee nell'`Condition`elemento di una policy.

La seguente policy mostra un esempio di come puoi negare l'accesso a sessioni con credenziali di sicurezza temporanee utilizzando la chiave di contesto della condizione.`aws:userid`.

- `AIDAXUSER1` rappresenta l'ID univoco per un utente IAM. Specificando l'ID univoco di un utente IAM come valore per la chiave di contesto `aws:userid`verrà negato l'accesso all'utente IAM. Ciò include tutte le sessioni di credenziali di sicurezza temporanee create chiamando l'API `GetSessionToken`.
- `AROAXROLE1:*` rappresenta l'ID univoco per tutte le sessioni associate al ruolo IAM. Specificando l'ID univoco di un ruolo IAM e un carattere jolly (*) nella parte caller-specified-role-session -name come valore per la chiave di contesto `aws:userid` verranno negate tutte le sessioni associate al ruolo.
- `AROAXROLE2:<caller-specified-role-session-name>` rappresenta l'ID univoco per una sessione del ruolo assunto. Nella parte caller-specified-role-session -name dell'ID univoco del ruolo assunto è possibile specificare un nome di sessione del ruolo o un carattere jolly se viene utilizzato l'operatore condition. StringLike Se specifichi il nome di sessione del ruolo, verrà negata la sessione di ruolo denominata senza influire sulle autorizzazioni del ruolo che ha creato le credenziali. Se specifichi un carattere jolly per il nome della sessione del ruolo, verranno negate tutte le sessioni associate al ruolo.

  > **Note**
  >
  > Il nome della sessione di ruolo specificato dal chiamante, che fa parte dell'identificatore univoco di una sessione del ruolo assunto, può cambiare durante il concatenamento dei ruoli. Il concatenamento dei ruoli si verifica quando un ruolo assume un altro ruolo. Il nome della sessione di ruolo viene impostato utilizzando il parametro `RoleSessionName` request quando il principale assume un ruolo utilizzando l'operazione API. AWS STS `AssumeRole`

- `account-id:<federated-user-caller-specified-name>`rappresenta l'ID univoco per una sessione utente AWS STS federata. Un utente IAM crea questa sessione chiamando l'API `GetFederationToken`. Se specifichi l'ID univoco per una sessione dell'utente federato AWS STS , la sessione federata denominata verrà negata senza influire sulle autorizzazioni dell'utente IAM che ha creato le credenziali.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:userId": [
            "AIDAXUSER1",
            "AROAXROLE1:*",
            "AROAXROLE2:<caller-specified-role-session-name>",
            "123456789012:<federated-user-caller-specified-name>"
          ]
        }
      }
    }
  ]
}
```

Per esempi specifici di valori chiave del principale, consulta Valori della chiave dell'entità principale. Per ulteriori informazioni sugli identificatori univoci IAM e su come ottenerli, consultaIdentificatori univoci.

Negare l'accesso a uno specifico principale con policy basate sulle risorse

Per limitare l'accesso a un principale specifico con una policy basata sulle risorse, puoi utilizzare le chiavi di contesto delle condizioni era: PrincipalArn o era: SourceIdentity nell'elemento `Condition`. Una policy basata sulle risorse è una policy di autorizzazione collegata a una risorsa che controlla chi può accedere alla risorsa e quali azioni è possibile eseguire su di essa.

Quando utilizzi la chiave di `aws:PrincipalARN` contesto, specifica l'ARN dell'utente, del ruolo o della sessione utente AWS STS federata IAM associata alle credenziali di sicurezza temporanee nell'elemento Condition di una policy. Il seguente esempio di policy mostra come utilizzare la chiave di contesto `aws:PrincipalARN` in una policy basata sulle risorse:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Principal": "*",
    "Effect": "Deny",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
    "Condition": {
      "ArnEquals": {
        "aws:PrincipalArn": [
          "arn:aws:iam::222222222222:role/ROLENAME",
          "arn:aws:iam::222222222222:user/USERNAME",
          "arn:aws:sts::222222222222:federated-user/USERNAME"
        ]
      }
    }
  }
}
```

Quando si utilizza la chiave di contesto `aws:SourceIdentity`, specifica il valore di identità di origine associato alle credenziali di sicurezza temporanee del ruolo nell'elemento `Condition` di una policy. Ciò vale a condizione che la sessione del ruolo sia stata emessa impostando il parametro di `SourceIdentity` richiesta quando il principale ha assunto un ruolo utilizzando qualsiasi AWS STS `assume-role` comando* CLI AWS STS `AssumeRole` o* operazioni API. Il seguente esempio di policy mostra come utilizzare la chiave di contesto `aws:SourceIdentity` in una policy basata sulle risorse:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Principal": "*",
    "Effect": "Deny",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
    "Condition": {
```

```
      "StringLike": {
        "aws:SourceIdentity": [
          "nikki_wolf@example.com",
          "<source identity value>"
        ]
      }
    }
  }
}
```

Se aggiorni solo la policy basata sull'identità per un principale, questo può comunque eseguire le azioni consentite nella policy basata sulle risorse, a meno che tali azioni siano negate esplicitamente nella policy basata sull'identità.

Per negare l'accesso a uno specifico principale con una policy basata sulle risorse

1. Fai riferimento a AWS servizi supportati da IAM per verificare se il servizio supporta policy basate sulle risorse.
2. Accedi Console di gestione AWS e apri la console per il servizio. Ogni servizio ha una posizione diversa nella console per allegare le policy.
3. Modifica la policy basata sulle risorse. Aggiungi un'istruzione di negazione della policy per specificare le informazioni identificative delle credenziali:

   a. Nell'elemento `Principal`, inserisci un carattere jolly (*). Il principale sarà limitato nell'elemento `Condition`.

   b. Nell'elemento `Effect`, inserisci "Nega".

   c. In `Action`, inserisci il namespace del servizio e il nome dell'azione da rifiutare. Per negare tutte le azioni, usa il carattere jolly (*). Ad esempio: `"s3:*"`.

   d. Nell'elemento `Resource`, inserisci l'ARN della risorsa di destinazione. Ad esempio: `"arn:aws:s3:::amzn-s3-demo-bucket"`.

   e. Nell'elemento `Condition`, specifica la chiave di contesto `aws:PrincipalARN` o `aws:SourceIdentity`.

      Se si utilizza la chiave di contesto `aws:PrincipalARN`, immetti l'ARN del principale a cui negare l'accesso.

      Se utilizzi la chiave di contesto `aws:SourceIdentity`, inserisci il valore dell'identità di origine impostato nella sessione del ruolo a cui negare l'accesso.

4. Salva il tuo lavoro.

Concessione delle autorizzazioni per creare credenziali di sicurezza temporanee

Per impostazione predefinita, gli utenti IAM non dispongono dell'autorizzazione per creare credenziali di sicurezza temporanee per ruoli e sessioni di utenti federati AWS STS . È necessario utilizzare una policy per fornire queste autorizzazioni agli utenti. Anche se è possibile concedere le autorizzazioni direttamente a un utente, ti consigliamo caldamente di assegnarle a un gruppo. In questo modo la gestione delle autorizzazioni risulta molto più semplice. Quando un utente non ha più bisogno di eseguire le operazioni associate alle autorizzazioni, non dovrai fare altro che rimuoverlo dal gruppo. Se un'altra persona si trova nella necessità di eseguire tale operazione, sarà sufficiente aggiungerla al gruppo per concederle le autorizzazioni.

Per concedere a un gruppo IAM l'autorizzazione per creare credenziali di sicurezza temporanee per ruoli o sessioni di utenti federati AWS STS , puoi collegare una policy che conceda uno o entrambi i seguenti privilegi:

- Affinché i principali federati OIDC e SAML possano accedere a un ruolo IAM, concedi l'accesso a. AWS STS `AssumeRole`
- Per gli utenti AWS STS federati che non necessitano di un ruolo, concedi l'accesso a. AWS STS `GetFederationToken`

Per informazioni sulle differenze fra le operazioni API `AssumeRole` e `GetFederationToken`, consultare Richiedere credenziali di sicurezza temporanee.

Per creare le credenziali di sicurezza temporanee, gli utenti IAM possono chiamare anche `GetSessionToken`. Non sono necessarie autorizzazioni perché un utente possa chiamare `GetSessionToken`. Lo scopo dell'operazione è autenticare l'utente tramite MFA. Non è possibile utilizzare le policy per controllare l'autenticazione. Ciò significa che non puoi impedire agli utenti IAM di chiamare `GetSessionToken` per creare le credenziali temporanee.

Example Esempio di policy che concede autorizzazioni per assumere un ruolo

La politica di esempio seguente concede l'autorizzazione a `AssumeRole` richiedere il `UpdateApp` ruolo in. Account AWS `123123123123` Quando si usa `AssumeRole`, l'utente (o l'applicazione) che crea le credenziali di sicurezza per conto di un utente federato non è in grado di delegare autorizzazioni che non siano già state specificate nella policy di autorizzazione del ruolo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::123123123123:role/UpdateAPP"
  }]
}
```

Example Esempio di policy che concede l'autorizzazione per creare credenziali di sicurezza temporanee per un utente federato.

La policy dell'esempio seguente concede l'autorizzazione per l'accesso a `GetFederationToken`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "sts:GetFederationToken",
    "Resource": "*"
  }]
}
```

Important

Quando si autorizzano utenti IAM a creare credenziali di sicurezza temporanee per gli utenti federati AWS STS con `GetFederationToken`, tieni a mente che tali utenti avranno la possibilità di delegare le proprie autorizzazioni. Per ulteriori informazioni sulla delega delle autorizzazioni tra utenti IAM e Account AWS, consulta. Esempi di policy per la delega dell'accesso Per informazioni sul controllo delle autorizzazioni nelle credenziali di sicurezza provvisorie, vedi Autorizzazioni per le credenziali di sicurezza temporanee.

Example Esempio di policy che concede a un utente autorizzazioni limitate per creare credenziali di sicurezza temporanee per utenti federati.

Quando si consente a un utente IAM di chiamare `GetFederationToken`, una best practice consiste nel limitare le autorizzazioni che l'utente IAM può delegare. Ad esempio, la seguente policy mostra come consentire a un utente IAM di creare credenziali di sicurezza temporanee solo per gli utenti AWS STS federati i cui nomi iniziano con Manager.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "sts:GetFederationToken",
    "Resource": ["arn:aws:sts::123456789012:federated-user/Manager*"]
  }]
}
```

Concessione delle autorizzazioni per l'utilizzo di sessioni di console rafforzate con l'identità

Le sessioni di console con identità migliorata consentono di includere Centro identità AWS IAM utenti e sessioni nelle sessioni della AWS console degli utenti IDs al momento dell'accesso. Ad esempio, Amazon Q Developer Pro utilizza sessioni di console rafforzate con l'identità per personalizzare l'esperienza del servizio. Per ulteriori informazioni sulle sessioni di console rafforzate con l'identità, consulta Enabling identity-enhanced console sessions nella Guida per l'utente di Centro identità AWS IAM . Per informazioni sulla configurazione di Amazon Q Developer, consulta Configurazione di Amazon Q Developer nella Guida per l'utente di Amazon Q Developer.

Affinché le sessioni di console rafforzate con l'identità siano disponibili per un utente, è necessario utilizzare una policy basata sull'identità per concedere al principale IAM l'autorizzazione `sts:SetContext` per la risorsa che rappresenta la propria sessione di console.

> **Important**
>
> Per impostazione predefinita, gli utenti non dispongono dell'autorizzazione per impostare il contesto per le sessioni di console rafforzate con l'identità. Per consentire ciò, è necessario

concedere al principale IAM l'autorizzazione `sts:SetContext` in una policy basata sull'identità, come illustrato nell'esempio di policy riportato di seguito.

L'esempio seguente di policy basata sull'identità concede l'`sts:SetContext`autorizzazione a un principale IAM, che consente al principale di impostare un contesto di sessione di console con identità migliorata per le proprie sessioni di console. AWS La risorsa policy,, `arn:aws:sts::account-id:self` rappresenta la sessione del chiamante. AWS Il segmento dell'ARN di `account-id` può essere sostituito con un carattere jolly * nei casi in cui la stessa policy di autorizzazione viene implementata su più account, ad esempio quando questa policy viene implementata utilizzando i set di autorizzazioni del Centro identità IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:SetContext",
            "Resource": "arn:aws:sts::111122223333:self"
        }
    ]
}
```

Manage (Gestione) AWS STS in un Regione AWS

Un endpoint regionale è l'URL del punto di ingresso all'interno di una particolare regione per un servizio AWS web. AWS consiglia di utilizzare gli endpoint Regional Servizio di token di sicurezza AWS (AWS STS) anziché l'endpoint globale per ridurre la latenza, integrare la ridondanza e aumentare la validità del token di sessione.

Sebbene l' AWS STS endpoint globale (legacy) sia altamente disponibile, `https://sts.amazonaws.com` è ospitato in una singola AWS regione, Stati Uniti orientali (Virginia settentrionale) e, come altri endpoint, non fornisce il failover automatico agli endpoint in altre regioni.

- Riduci la latenza: effettuando AWS STS chiamate verso un endpoint geograficamente più vicino ai tuoi servizi e alle tue applicazioni, puoi accedere AWS STS ai servizi con una latenza inferiore e tempi di risposta migliori.
- Progetta in ridondanza: puoi limitare gli effetti di un guasto all'interno di un carico di lavoro a un numero limitato di componenti con un ambito prevedibile di contenimento degli impatti. L'utilizzo AWS STS degli endpoint regionali consente di allineare l'ambito dei componenti con l'ambito dei token di sessione. Per ulteriori informazioni su questo pilastro dell'affidabilità, consulta Usare l'isolamento dei guasti per proteggere il carico di lavoro nel Framework. AWS Well-Architected
- Aumenta la validità del token di sessione: i token di sessione degli AWS STS endpoint regionali sono validi in tutti i casi. Regioni AWS I token di sessione provenienti dall'endpoint STS globale sono validi solo se sono abilitati per Regioni AWS impostazione predefinita. Se intendi abilitare una nuova regione per il tuo account, puoi utilizzare i token di sessione degli endpoint regionali. AWS STS Se scegli di utilizzare l'endpoint globale, devi modificare la compatibilità regionale dei token di AWS STS sessione per l'endpoint globale. In questo modo si garantisce che i token siano validi in tutti i casi. Regioni AWS

Per un elenco delle AWS STS regioni e dei relativi endpoint, consulta. AWS STS Regioni ed endpoint

> **Note**
>
> AWS ha apportato modifiche all'endpoint globale Servizio di token di sicurezza AWS (AWS STS`https://sts.amazonaws.com`) nelle regioni abilitate di default per migliorarne la resilienza e le prestazioni. AWS STS le richieste all'endpoint globale vengono gestite automaticamente insieme ai carichi di lavoro Regione AWS . Queste modifiche non verranno implementate nelle Regioni con attivazione facoltativa. Ti consigliamo di utilizzare gli endpoint AWS STS regionali appropriati. Per ulteriori informazioni, consulta AWS STS cambiamenti globali degli endpoint.

Argomenti

- Attivazione e disattivazione AWS STS in un Regione AWS
- Scrittura del codice da utilizzare AWS STS Regioni
- Gestione dei token di sessione emessi dall'endpoint globale

Attivazione e disattivazione AWS STS in un Regione AWS

Quando attivi gli AWS STS endpoint per una regione, AWS STS puoi rilasciare credenziali temporanee agli utenti e ai ruoli del tuo account che effettuano una AWS STS richiesta. Queste credenziali possono essere utilizzate in qualsiasi regione abilitata di default o manualmente.

Per le regioni abilitate per impostazione predefinita, è necessario attivare l' AWS STS endpoint regionale nell'account in cui vengono generate le credenziali temporanee. Al momento di effettuare la richiesta, non importa se un utente è autenticato sullo stesso account o su un altro account.

Quando si richiedono credenziali temporanee per un ruolo in un'altra regione Account AWS utilizzando una regione abilitata manualmente, l'account di destinazione (l'account contenente il ruolo) deve abilitare tale regione per le operazioni. AWS STS Ciò garantisce che le credenziali di sicurezza temporanee possano essere generate correttamente.

Immagina, ad esempio, che un utente nell'account A desideri inviare una richiesta API `sts:AssumeRole` all'endpoint regionale AWS STS `https://sts.ap-southeast-3.amazonaws.com`. La richiesta si riferisce a delle credenziali temporanee per il ruolo denominato `Developer` nell'account B. Poiché la richiesta è di creare le credenziali per un'entità nell'account B, l'account B deve avere la Regione `ap-southeast-3` abilitata. Gli utenti dell'account A (o di qualsiasi altro account) possono chiamare l'endpoint `ap-southeast-3` AWS STS per richiedere le credenziali per l'account B, che la regione sia attivata o meno nel loro account. Per saperne di più, vedi Abilitare o disabilitare Regioni AWS nel tuo account.

> **Note**
>
> Le regioni attive sono disponibili per tutti gli utenti che utilizzano credenziali provvisorie in tale account. Per controllare quali utenti o ruoli IAM possono accedere alla regione, utilizza la chiave di condizione `aws:RequestedRegion` nelle tue policy di autorizzazione.

Per attivare o disattivare AWS STS in una regione abilitata per impostazione predefinita (console)

1. Accedi come utente root o come utente con le autorizzazioni per eseguire attività di amministrazione di IAM.
2. Apri la console IAM e, nel pannello di navigazione, seleziona Impostazioni account.
3. Nella sezione Endpoint di Security Token Service (STS), trova la Regione che desideri configurare, quindi scegli Attiva o Inattiva nella colonna Stato STS.

4. Nella finestra di dialogo visualizzata, scegli Activate (Attiva) o Deactivate (Disattiva).

Per le regioni che devono essere abilitate, le attiviamo AWS STS automaticamente quando abiliti la regione. Dopo aver abilitato una regione, AWS STS è sempre attiva per la regione e non è possibile disattivarla. Per informazioni su come abilitare le regioni che sono disattivate per impostazione predefinita, consulta Specificare quali aree può utilizzare Regioni AWS il tuo account nella Guida di Gestione dell'account AWS riferimento.

Scrittura del codice da utilizzare AWS STS Regioni

Dopo aver attivato una regione, puoi indirizzare le chiamate AWS STS API verso quella regione. Il frammento di codice Java seguente mostra come configurare un oggetto `AWSSecurityTokenService` affinché effettui richieste alla regione Europa (Milano) (eu-south-1).

```
EndpointConfiguration regionEndpointConfig = new EndpointConfiguration("https://sts.eu-
south-1.amazonaws.com", "eu-south-1");
AWSSecurityTokenService stsRegionalClient =
 AWSSecurityTokenServiceClientBuilder.standard()
.withCredentials(credentials)
.withEndpointConfiguration(regionEndpointConfig)
.build();
```

AWS STS consiglia di effettuare chiamate verso un endpoint regionale. Per scoprire come abilitare manualmente una regione, consulta Specificare le Regioni AWS che il tuo account può utilizzare nella Guida di riferimento a Gestione dell'account AWS .

Nell'esempio, la prima riga crea un'istanza di un oggetto `EndpointConfiguration` chiamata `regionEndpointConfig`, passando l'URL dell'endpoint e la Regione AWS come parametri.

Per informazioni su come impostare gli endpoint AWS STS regionali utilizzando una variabile di ambiente per gli AWS SDK, consulta gli endpoint AWS STS regionalizzati nella Guida di riferimento per SDK and Tools. AWS

Per tutte le altre combinazioni di linguaggio e ambiente di programmazione, consulta la documentazione dell'SDK pertinente.

Gestione dei token di sessione emessi dall'endpoint globale

Per impostazione predefinita, la maggior Regioni AWS parte è abilitata per le operazioni. Servizi AWS Queste regioni vengono attivate automaticamente per essere utilizzate con AWS STS. Alcune

regioni, ad esempio Asia Pacifico (Hong Kong), devono essere abilitate manualmente. Per ulteriori informazioni sull'abilitazione e la disabilitazione di Regioni AWS, consulta Specificare le Regioni AWS che il tuo account può utilizzare nella Guida di riferimento a Gestione dell'account AWS .

Quando abiliti queste AWS regioni, vengono attivate automaticamente per essere utilizzate con AWS STS. Non è possibile attivare l' AWS STS endpoint per una regione disattivata. I token di sessione validi per tutti Regioni AWS includono più caratteri rispetto ai token validi nelle regioni abilitate per impostazione predefinita. La modifica di questa impostazione potrebbe influenzare i sistemi esistenti in cui vengono memorizzati temporaneamente i token.

Puoi modificare questa impostazione utilizzando l'API Console di gestione AWS, AWS CLI, o AWS .

Modificare le regioni compatibili con i token di sessione l'endpoint globale (console)

1. Accedi come utente root o come utente con le autorizzazioni per eseguire attività di amministrazione di IAM. Per modificare la compatibilità dei token di sessione, è necessario disporre di una policy che consente l'operazione `iam:SetSecurityTokenServicePreferences`.
2. Apri la console IAM. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
3. Nella sezione Security Token Service (STS) Token di sessione dagli endpoint STS. L'endpoint globale indica `Valid only in Regioni AWS enabled by default`. Scegliere Change (Cambia).
4. Nella finestra di dialogo Modifica la compatibilità della regione, seleziona Tutto Regioni AWS. Selezionare quindi Save changes (Salva modifiche).

> **Note**
>
> I token di sessione validi per tutti Regione AWS includono più caratteri rispetto ai token validi nelle regioni abilitate per impostazione predefinita. La modifica di questa impostazione potrebbe influenzare i sistemi esistenti in cui vengono memorizzati temporaneamente i token.

Per modificare la compatibilità regionale dei token di sessione per l'endpoint globale (AWS CLI)

Imposta la versione del token di sessione. I token della versione 1 sono validi solo in Regioni AWS quanto sono disponibili per impostazione predefinita. Questi token non funzionano nelle regioni

abilitate manualmente, ad esempio Asia Pacifico (Hong Kong). I token Versione 2 sono validi in tutte le regioni. Tuttavia, i token versione 2 sono composti da un numero maggiore di caratteri e ciò può influire sui sistemi in cui vengono memorizzati temporaneamente i token.

- `aws iam set-security-token-service-preferences`

Per modificare la compatibilità regionale dei token di sessione per l'endpoint globale (AWS API)

Imposta la versione del token di sessione. I token della versione 1 sono validi solo se disponibili per impostazione predefinita. Regioni AWS Questi token non funzionano nelle regioni abilitate manualmente, ad esempio Asia Pacifico (Hong Kong). I token Versione 2 sono validi in tutte le regioni. Tuttavia, i token versione 2 sono composti da un numero maggiore di caratteri e ciò può influire sui sistemi in cui vengono memorizzati temporaneamente i token.

- `SetSecurityTokenServicePreferences`

AWS STS Regioni ed endpoint

> **Note**
>
> AWS ha apportato modifiche all'endpoint globale Servizio di token di sicurezza AWS (AWS STS`https://sts.amazonaws.com`) nelle Regioni abilitate di default per migliorarne la resilienza e le prestazioni. AWS STS le richieste all'endpoint globale vengono servite automaticamente Regione AWS come i tuoi carichi di lavoro. Queste modifiche non verranno implementate nelle Regioni con attivazione facoltativa. Ti consigliamo di utilizzare gli endpoint AWS STS regionali appropriati. Per ulteriori informazioni, consulta AWS STS cambiamenti globali degli endpoint.

La seguente tabella elenca le regioni e i relativi endpoint. Indica quali sono attivate per default e quali possono essere attivate o disattivate.

Nome della Regione	Endpoint	Attivo per impostazione predefinita	Attivazione/disattivazione manuale
--Globale--	sts.amazonaws.com	✓ Sì	✗ No
Stati Uniti orientali (Ohio)	sts.us-east-2.amazonaws.com	✓ Sì	✓ Sì
Stati Uniti orientali (Virginia settentrionale)	sts.us-east-1.amazonaws.com	✓ Sì	✗ No
Stati Uniti occidentali (California settentrionale)	sts.us-west-1.amazonaws.com	✓ Sì	✓ Sì
Stati Uniti occidentali (Oregon)	sts.us-west-2.amazonaws.com	✓ Sì	✓ Sì
Africa (Città del Capo)	sts.af-south-1.amazonaws.com	✗ No[1]	✗ No
Asia Pacifico (Hong Kong)	sts.ap-east-1.amazonaws.com	✗ No[1]	✗ No
Asia Pacifico (Hyderabad)	sts.ap-south-2.amazonaws.com	✗ No[1]	✗ No
Asia Pacifico (Giacarta)	sts.ap-southeast-3.amazonaws.com	✗ No[1]	✗ No
Asia Pacifico (Malesia)	sts.ap-southeast-5.amazonaws.com	✗ No[1]	✗ No
Asia Pacifico (Melbourne)	sts.ap-southeast-4.amazonaws.com	✗ No[1]	✗ No

Nome della Regione	Endpoint	Attivo per impostazione predefinita	Attivazione/disattivazione manuale
Asia Pacifico (Mumbai)	sts.ap-south-1.amazonaws.com	✓ Sì	✓ Sì
Asia Pacifico (Osaka)	sts.ap-northeast-3.amazonaws.com	✓ Sì	✓ Sì
Asia Pacifico (Seul)	sts.ap-northeast-2.amazonaws.com	✓ Sì	✓ Sì
Asia Pacifico (Singapore)	sts.ap-southeast-1.amazonaws.com	✓ Sì	✓ Sì
Asia Pacifico (Sydney)	sts.ap-southeast-2.amazonaws.com	✓ Sì	✓ Sì
Asia Pacifico (Thailandia)	sts.ap-southeast-7.amazonaws.com	✗ No[1]	✗ No
Asia Pacifico (Tokyo)	sts.ap-northeast-1.amazonaws.com	✓ Sì	✓ Sì
Canada (Centrale)	sts.ca-central-1.amazonaws.com	✓ Sì	✓ Sì
Canada occidentale (Calgary)	sts.ca-west-1.amazonaws.com	✗ No[1]	✗ No
Cina (Pechino)	sts.cn-north-1.amazonaws.com.cn	✓ Sì[1]	✗ No
Cina (Ningxia)	sts.cn-northwest-1.amazonaws.com.cn	✓ Sì[1]	✓ Sì
Europa (Francoforte)	sts.eu-central-1.amazonaws.com	✓ Sì	✓ Sì
Europa (Irlanda)	sts.eu-west-1.amazonaws.com	✓ Sì	✓ Sì
Europa (Londra)	sts.eu-west-2.amazonaws.com	✓ Sì	✓ Sì

Nome della Regione	Endpoint	Attivo per impostazione predefinita	Attivazione/disattivazione manuale
Europa (Milano)	sts.eu-south-1.amazonaws.com	⊗ No[1]	⊗ No
Europa (Parigi)	sts.eu-west-3.amazonaws.com	⊙ Sì	⊙ Sì
Europa (Spagna)	sts.eu-south-2.amazonaws.com	⊗ No[1]	⊗ No
Europa (Stoccolma)	sts.eu-north-1.amazonaws.com	⊙ Sì	⊙ Sì
Europa (Zurigo)	sts.eu-central-2.amazonaws.com	⊗ No[1]	⊗ No
Israele (Tel Aviv)	sts.il-central-1.amazonaws.com	⊗ No[1]	⊗ No
Messico (centrale)	sts.mx-central-1.amazonaws.com	⊗ No[1]	⊗ No
Medio Oriente (Bahrein)	sts.me-south-1.amazonaws.com	⊗ No[1]	⊗ No
Medio Oriente (Emirati Arabi Uniti)	sts.me-central-1.amazonaws.com	⊗ No[1]	⊗ No
Sud America (San Paolo)	sts.sa-east-1.amazonaws.com	⊙ Sì	⊙ Sì

[1]È necessario abilitare la regione per utilizzarla. Ciò attiva automaticamente AWS STS. Non è possibile attivare o disattivare manualmente AWS STS in queste regioni.

[2]Per utilizzarlo AWS in Cina, sono necessari un account e credenziali specifici per la AWS Cina.

AWS STS cambiamenti globali degli endpoint

AWS ha apportato modifiche all'endpoint globale Servizio di token di sicurezza AWS (AWS STS`https://sts.amazonaws.com`) nelle Regioni abilitate di default per migliorarne la resilienza e le prestazioni. In precedenza, tutte le richieste all'endpoint AWS STS globale venivano servite da un unico dispositivo Regione AWS, gli Stati Uniti orientali (Virginia settentrionale). Ora, nelle regioni abilitate per impostazione predefinita, le richieste all'endpoint AWS STS globale vengono servite

automaticamente nella stessa regione da cui proviene la richiesta, anziché nella regione degli Stati Uniti orientali (Virginia settentrionale). Queste modifiche non verranno implementate nelle Regioni con attivazione facoltativa.

Con questa modifica, AWS STS elaborerà la richiesta in base alla regione di origine e al resolver DNS utilizzati. Le richieste all'endpoint AWS STS globale vengono servite nella stessa regione del carico di lavoro AWS distribuito se la richiesta DNS per l'endpoint AWS STS globale viene gestita dal server Amazon DNS nelle regioni abilitate per impostazione predefinita. Le richieste all'endpoint globale AWS STS continueranno a essere servite nella Regione Stati Uniti orientali (Virginia settentrionale) se la richiesta proviene da Regioni con attivazione facoltativa o se la richiesta è stata risolta utilizzando un resolver DNS diverso dal server Amazon DNS. Per ulteriori informazioni su Amazon DNS, consulta Server DNS Amazon nella Guida per l'utente di Amazon Virtual Private Cloud.

La tabella seguente mostra come le richieste verso l'endpoint AWS STS globale vengono instradate in base al provider DNS.

Resolver DNS	Le richieste all'endpoint AWS STS globale vengono instradate verso il locale? Regione AWS
Resolver DNS Amazon in un Amazon VPC in una Regione abilitata per impostazione predefinita	Sì
Resolver DNS Amazon in un Amazon VPC in una Regione con attivazione facoltativa	No, la richiesta verrà instradata nella Regione Stati Uniti orientali (Virginia settentrionale)
Resolver DNS fornito dal tuo ISP, da un provider DNS pubblico o da qualsiasi altro provider DNS	No, la richiesta verrà instradata nella Regione Stati Uniti orientali (Virginia settentrionale)

Per garantire un'interruzione minima dei processi esistenti, AWS ha implementato le seguenti misure:

- AWS CloudTrail i registri delle richieste effettuate all'endpoint AWS STS globale vengono inviati nella regione Stati Uniti orientali (Virginia settentrionale). CloudTrail i registri delle richieste servite dagli endpoint AWS STS regionali continueranno a essere registrati nella rispettiva regione. CloudTrail

- CloudTrail i registri per le operazioni eseguite dall'endpoint AWS STS globale e dagli endpoint regionali contengono campi aggiuntivi e indicano quale endpoint `endpointType` e `awsServingRegion` regione hanno fornito la richiesta. Per esempi di CloudTrail log, vedere. Esempio AWS STS Evento API che utilizza l'endpoint globale nel file di registro CloudTrail
- Le richieste effettuate all'endpoint AWS STS globale hanno un valore pari a `us-east-1` for the `aws:RequestedRegion` condition key, indipendentemente dalla regione che ha fornito la richiesta.
- Le richieste gestite dall'endpoint AWS STS globale non condividono una quota di richieste al secondo con gli endpoint regionali. AWS STS

Se disponi di carichi di lavoro in una regione opzionale e stai ancora utilizzando l'endpoint AWS STS globale, ti consigliamo di effettuare la migrazione agli endpoint AWS STS regionali per migliorare la resilienza e le prestazioni. Per ulteriori informazioni sulla configurazione degli endpoint regionali, consulta AWS STS Endpoint regionali nella and Tools Reference AWS STS Guide.AWS SDKs

AWS CloudTrail e endpoint regionali

Le chiamate agli endpoint regionali e globali vengono registrate sul campo `tlsDetails` in AWS CloudTrail. Le chiamate verso gli endpoint regionali, ad esempio`us-east-2.amazonaws.com`, vengono registrate nella regione CloudTrail appropriata. Le chiamate all'endpoint globale, `sts.amazonaws.com`, vengono registrate come chiamate a un servizio globale. Gli eventi per gli AWS STS endpoint globali vengono registrati su us-east-1.

> **Note**
>
> `tlsDetails` può essere visualizzato solo per i servizi che supportano questo campo. Vedi i dettagli sui servizi che supportano TLS nella Guida per l'utente CloudTrail AWS CloudTrail Per ulteriori informazioni, consulta Registrazione di IAM e AWS STS Chiamate API con AWS CloudTrail.

Abilita l'accesso personalizzato del broker di identità al AWS console

Puoi scrivere ed eseguire codice per creare un URL che permette agli utenti che accedono alla rete dell'organizzazione di accedere in modo sicuro alla Console di gestione AWS. L'URL include un token di accesso che ottieni AWS e che autentica l'utente. AWS La sessione console risultante potrebbe includere una distinta `AccessKeyId` a causa della federazione. Per tracciare l'utilizzo della chiave di

accesso per l'accesso federativo tramite CloudTrail eventi correlati, vedi Registrazione di IAM e AWS STS Chiamate API con AWS CloudTrail e accedi agli eventi. Console di gestione AWS

> **Note**
>
> Se la tua organizzazione usa un provider di identità (IdP) compatibile con SAML, puoi configurare l'accesso alla console senza la necessità di scrivere codice. Ciò è possibile con provider come Microsoft Active Directory Federation Services oppure Shibboleth open source. Per informazioni dettagliate, vedi Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS.

Per consentire agli utenti della tua organizzazione di accedere a Console di gestione AWS, puoi creare un broker di identità personalizzato che esegua i seguenti passaggi:

1. Verifica che l'utente sia autenticato dal sistema di identità locale.
2. Chiama le operazioni Servizio di token di sicurezza AWS (AWS STS) AssumeRole (consigliata) o GetFederationToken API per ottenere credenziali di sicurezza temporanee per l'utente. Per informazioni sui diversi metodi che si possono utilizzare per assumere un ruolo, consulta Metodi per assumere un ruolo. Per informazioni su come passare i tag di sessione facoltativi quando si ottengono le credenziali di sicurezza, consulta Passa i tag della sessione in AWS STS.
   - Se usi una delle operazioni API `AssumeRole*` per ottenere credenziali di sicurezza temporanee per un ruolo, puoi includere il parametro `DurationSeconds` nella chiamata. Questo parametro specifica la durata della sessione del ruolo, da 900 secondi (15 minuti) fino all'impostazione di durata massima della sessione per il ruolo. Quando si utilizza `DurationSeconds` in un'operazione `AssumeRole*`, è necessario chiamarlo come un utente IAM con credenziali a lungo termine. In caso contrario, la chiamata all'endpoint di federazione nella fase 3 ha esito negativo. Per informazioni su come visualizzare o modificare il valore massimo per un ruolo, consulta Aggiornamento della durata massima della sessione per un ruolo.
   - Se usi l'operazione API `GetFederationToken` per ottenere le credenziali, puoi includere il parametro `DurationSeconds` nella chiamata. Questo parametro specifica la durata della sessione del ruolo. Il valore può variare da 900 secondi (15 minuti) a 129.600 secondi (36 ore). Puoi effettuare questa chiamata API solo utilizzando le credenziali di AWS sicurezza a lungo termine di un utente IAM. Puoi anche effettuare queste chiamate utilizzando Utente root dell'account AWS le credenziali, ma non lo consigliamo. Se effettui la chiamata come utente root, la sessione di default dura un'ora. In alternativa, puoi specificare una sessione con durata compresa tra 900 secondi (15 minuti) e 3.600 secondi (un'ora).

3. Chiama l'endpoint AWS federativo e fornisci le credenziali di sicurezza temporanee per richiedere un token di accesso.
4. Crea un URL per la console che include il token:
   - Se usi una delle operazioni API `AssumeRole*` nell'URL, puoi includere il parametro HTTP `SessionDuration`. Questo parametro specifica la durata della sessione della console, da 900 secondi (15 minuti) a 43.200 secondi (12 ore).
   - Se usi l'operazione API `GetFederationToken` nell'URL, puoi includere il parametro `DurationSeconds`. Questo parametro specifica la durata della sessione della console federata. Il valore può variare da 900 secondi (15 minuti) a 129.600 secondi (36 ore).

   > **Note**
   >
   > - `SessionDuration` non può essere maggiore o uguale alla durata massima della sessione impostata per il ruolo che stai assumendo. Ad esempio, hai impostato la durata massima della sessione per il ruolo che desideri assumere su 5 ore. Il parametro `SessionDuration` può essere 16.524 secondi o 4 ore e 59 secondi.
   > - Non usare il parametro HTTP `SessionDuration` quando ottieni le credenziali temporanee con `GetFederationToken`. L'operazione avrà esito negativo.
   > - L'utilizzo delle credenziali perché un ruolo assuma un ruolo diverso viene chiamato concatenamento dei ruoli. Quando si utilizza il concatenamento dei ruoli, le nuove credenziali sono limitate a una durata massima di un'ora. Quando si utilizzano i ruoli per concedere autorizzazioni alle applicazioni eseguite su istanze EC2, tali applicazioni non sono soggette a questa limitazione.
   > - Non usare il parametro HTTP `SessionDuration` quando ottieni le credenziali temporanee tramite la concatenazione dei ruoli. L'operazione avrà esito negativo.
5. Fornisce l'URL all'utente o richiama l'URL per conto dell'utente.

L'URL fornito dall'endpoint di federazione è valido per 15 minuti dopo la creazione. Questo intervallo di tempo è diverso dalla durata (in secondi) della sessione delle credenziali di sicurezza temporanee associata all'URL. Queste credenziali sono valide per la durata specificata al momento della creazione, a partire dal momento in cui sono state create.

> **Important**
>
> L'URL consente l'accesso alle tue AWS risorse tramite Console di gestione AWS se hai abilitato le autorizzazioni nelle credenziali di sicurezza temporanee associate. Per questo motivo, devi trattare l'URL come segreto. Ti consigliamo di restituire l'URL attraverso un reindirizzamento sicuro, ad esempio usando un codice di stato della risposta HTTP 302 in una connessione SSL. Per ulteriori informazioni sul codice di stato della risposta HTTP 302, vedere RFC 2616, sezione 10.3.3.

Per completare queste attività, puoi utilizzare l'API di query HTTPS per AWS Identity and Access Management (IAM) e Servizio di token di sicurezza AWS (AWS STS). In alternativa, puoi utilizzare linguaggi di programmazione come Java, Ruby o C # con l'SDK AWS appropriato. Ognuno di questi metodi è descritto negli argomenti seguenti.

Argomenti

- Codice di esempio per l'uso delle operazioni API di query IAM
- Codice di esempio con Python
- Esempio di codice con Java
- Esempio ci creazione dell'URL (Ruby)

Codice di esempio per l'uso delle operazioni API di query IAM

Puoi creare un URL che concede ai ruoli e ai principali federati l'accesso diretto alla Console di gestione AWS. Questa attività utilizza IAM e AWS STS HTTPS Query API. Per ulteriori informazioni su come effettuare richieste di query, consulta Effettuare richieste di query.

> **Note**
>
> La procedura seguente contiene esempi di stringhe di testo. Per migliorare la leggibilità, sono state aggiunte interruzioni di riga in alcuni degli esempi più lunghi. Quando crei queste stringhe per l'uso, ometti le interruzioni di riga.

Per consentire ai ruoli e ai presidi federati di accedere alle tue risorse da Console di gestione AWS

1. Autentica l'utente nel sistema di identità e autorizzazione.

2. Ottieni credenziali di sicurezza temporanee per l'utente. Le credenziali temporanee sono costituite da un ID chiave di accesso, una chiave di accesso segreta e un token di sessione. Per ulteriori informazioni sulla creazione di credenziali temporanee, consulta Credenziali di sicurezza temporanee in IAM.

   Per ottenere credenziali temporanee, chiami l' AWS STS AssumeRoleAPI (consigliata) o l'API. GetFederationToken Per ulteriori informazioni sulle differenze tra queste operazioni API, consulta Understanding the API Options for Security Delegating Access to Your AWS Account nel AWS Security Blog.

   > **Important**
   >
   > Quando utilizzi l'GetFederationTokenAPI per creare credenziali di sicurezza temporanee, devi specificare le autorizzazioni che le credenziali concedono all'utente che assume il ruolo. Per le operazioni API che iniziano con `AssumeRole*`, è necessario usare un ruolo IAM per assegnare le autorizzazioni. Per le altre operazioni API, il meccanismo varia a seconda dell'API. Per ulteriori dettagli, consultare Autorizzazioni per le credenziali di sicurezza temporanee. Inoltre, se usi le operazioni API `AssumeRole*`, devi chiamarle come utente IAM con credenziali a lungo termine. In caso contrario, la chiamata all'endpoint di federazione nella fase 3 ha esito negativo.

3. Dopo aver ottenuto le credenziali di sicurezza temporanee, integrale in una stringa di sessione JSON per scambiarle con un token di accesso. Nell'esempio seguente viene illustrato come codificare le credenziali. Sostituisci il testo segnaposto con i valori appropriati delle credenziali ricevute nella fase precedente.

   ```
   {"sessionId":"*** temporary access key ID ***",
   "sessionKey":"*** temporary secret access key ***",
   "sessionToken":"*** session token ***"}
   ```

4. Effettuare la codifica tramite URL della stringa di sessione della fase precedente. Poiché le informazioni codificate sono informazioni sensibili, ti consigliamo di evitare l'uso di un servizio Web per la codifica. Usa invece una funzione o una caratteristica installata in locale nel kit di strumenti di sviluppo per codificare queste informazioni in modo sicuro. Puoi usare la funzione `urllib.quote_plus` in Python, la funzione `URLEncoder.encode` in Java o la funzione `CGI.escape` in Ruby. Consulta gli esempi più avanti in questo argomento.

5.

> **Note**
>
> AWS supporta le richieste POST qui.

Invia la tua richiesta all'endpoint della AWS federazione:

`https://region-code.signin.aws.amazon.com/federation`

Per un elenco dei *region-code* valori possibili, consulta la colonna Regione negli AWS Sign-In endpoints endpoints È possibile utilizzare facoltativamente l'endpoint di federazione predefinito AWS Sign-In :

`https://signin.aws.amazon.com/federation`

La richiesta deve includere i parametri `Action` e `Session` e, facoltativamente, se è stata utilizzata un'operazione API `AssumeRole*`, un parametro HTTP `SessionDuration` come illustrato nell'esempio seguente.

```
Action = getSigninToken
SessionDuration = time in seconds
Session = *** the URL encoded JSON string created in steps 3 & 4 ***
```

> **Note**
>
> Le seguenti istruzioni in questa fase funzionano solo con le richieste GET.

Il parametro HTTP `SessionDuration` specifica la durata della sessione della console. Si tratta di un valore diverso rispetto alla durata delle credenziali temporanee specificato usando il parametro `DurationSeconds`. Puoi specificare un valore massimo di `SessionDuration` pari a 43.200 (12 ore). Se il `SessionDuration` parametro è mancante, per impostazione predefinita la durata delle credenziali recuperate AWS STS nel passaggio 2 (che per impostazione predefinita è un'ora). Consultare la documentazione per l'API `AssumeRole` per informazioni dettagliate su come specificare una durata tramite il parametro `DurationSeconds`. La possibilità di creare una sessione della console più lunga di un'ora è intrinseca nell'operazione `getSigninToken` dell'endpoint di federazione.

> **Note**
>
> - `SessionDuration` non può essere maggiore o uguale alla durata massima della sessione impostata per il ruolo che stai assumendo. Ad esempio, hai impostato la durata massima della sessione per il ruolo che desideri assumere su 5 ore. Il parametro `SessionDuration` può essere 16.524 secondi o 4 ore e 59 secondi.
> - Non usare il parametro HTTP `SessionDuration` quando ottieni le credenziali temporanee con `GetFederationToken`. L'operazione avrà esito negativo.
> - L'utilizzo delle credenziali perché un ruolo assuma un ruolo diverso viene chiamato concatenamento dei ruoli. Quando si utilizza il concatenamento dei ruoli, le nuove credenziali sono limitate a una durata massima di un'ora. Quando si utilizzano i ruoli per concedere autorizzazioni alle applicazioni eseguite su istanze EC2, tali applicazioni non sono soggette a questa limitazione.
> - Non usare il parametro HTTP `SessionDuration` quando ottieni le credenziali temporanee tramite la concatenazione dei ruoli. L'operazione avrà esito negativo.

Quando abiliti le sessioni della console con una durata estesa, aumenti il rischio di esposizione delle credenziali. Per mitigare questo rischio, è possibile disabilitare immediatamente le sessioni della console attive per tutti i ruoli, scegliendo Revoca sessioni nella pagina Riepilogo ruolo nella console IAM. Per ulteriori informazioni, consulta Revocare le credenziali di sicurezza temporanee per i ruoli IAM.

Di seguito è riportato un esempio di richiesta. Per le righe è impostato il ritorno a capo per semplificare la lettura, ma devi inviare la richiesta come stringa su un'unica riga.

```
https://signin.aws.amazon.com/federation
?Action=getSigninToken
&SessionDuration=1800
&Session=%7B%22sessionId%22%3A+%22ASIAJUMHIZPTOKTBMK5A%22%2C+%22sessionKey%22
%3A+%22LSD7LWI%2FL%2FN%2BgYpan5QFz0XUpc8s7HYjRsgcsrsm%22%2C+%22sessionToken%2
2%3A+%22FQoDYXdzEBQaDLbj3VWv2u50NN%2F3yyLSASwYtWhPnGPMNmzZFfZsL0Qd3vtYHw5A5dW
AjOsrkdPkghomIe3mJip5%2F0djDBbo7SmO%2FENDEiCdpsQKodTpleKA8xQq0CwFg6a69xdEBQT8
FipATnLbKoyS4b%2FebhnsTUjZZQWp0wXXqFF7gSm%2FMe2tXe0jzsdP0O12obez9lijPSdF1k2b5
PfGhiuyAR9aD5%2BubM0pY86fKex1qsytjvyTbZ9nXe6DvxVDcnCOhOGETJ7XFkSFdH0v%2FYR25C
UAhJ3nXIkIbG7Ucv9cOEpCf%2Fg23ijRgILIBQ%3D%3D%22%7D
```

La risposta dell'endpoint di federazione è un documento JSON con un valore `SigninToken`. L'aspetto sarà simile all'esempio seguente.

```
{"SigninToken":"*** the SigninToken string ***"}
```

6.

> **Note**
>
> AWS supporta le richieste POST qui.

Infine, crea l'URL che gli utenti possono usare per accedere alla Console di gestione AWS. L'URL corrisponde all'URL dell'endpoint di federazione usato in Step 5, con l'aggiunta dei parametri seguenti:

```
?Action = login
&Issuer = *** the form-urlencoded URL for your internal sign-in page ***
&Destination = *** the form-urlencoded URL to the desired AWS console page ***
&SigninToken = *** the value of SigninToken received in the previous step ***
```

> **Note**
>
> Le seguenti istruzioni in questa fase funzionano solo con utilizzando l'API GET.

L'esempio seguente mostra l'aspetto dell'URL finale. L'URL è valido per 15 minuti dal momento della creazione. Le credenziali di sicurezza temporanee e la sessione della console incorporate nell'URL sono valide per la durata specificata nel parametro HTTP `SessionDuration` al momento della richiesta iniziale.

```
https://signin.aws.amazon.com/federation
?Action=login
&Issuer=https%3A%2F%2Fexample.com
&Destination=https%3A%2F%2Fconsole.aws.amazon.com%2F
&SigninToken=VCQgs5qZZt3Q6fn8Tr5EXAMPLEmLnwB7JjUc-SHwnUUWabcRdnWsi4DBn-dvC
CZ85wrD0nmldUcZEXAMPLE-vXYH4Q__mleuF_W2BE5HYexbe9y4Of-kje53SsjNNecATfjIzpW1
WibbnH6YcYRiBoffZBGExbEXAMPLE5aiKX4THWjQKC6gg6alHu6JFrnOJoK3dtP6I9a6hi6yPgm
iOkPZMmNGmhsvVxetKzr8mx3pxhHbMEXAMPLETv1pij0rok3IyCR2YVcIjqwfWv32HU2Xlj471u
3fU6uOfUComeKiqTGX974xzJOZbdmX_t_lLrhEXAMPLEDDIisSnyHGw2xaZZqudm4mo2uTDk9Pv
```

```
9l5K0ZCqIgEXAMPLEcA6tgLPykEWGUyH6BdSC6166n4M4JkXIQgac7_7821YqixsNxZ6rsrpzwf
nQoS14O7R0eJCCJ684EXAMPLEZRdBNnuLbUYpz2Iw3vIN0tQgOujwnwydPscM9F7foaEK3jwMkg
Apeb1-6L_OB12MZhuFxx55555EXAMPLEhyETEd4ZulKPdXHkgl6T9ZkIlHz2Uy1RUTUhhUxNtSQ
nWc5xkbBoEcXqpoSIeK7yhje9Vzhd61AEXAMPLElbWeouACEMG6-Vd3dAgFYd6i5FYoyFrZLWvm
0LSG7RyYKeYN5VIzUk3YWQpyjP0RiT5KUrsUi-NEXAMPLExMOMdoODBEgKQsk-iu2ozh6r8bxwC
RNhujg
```

Codice di esempio con Python

Gli esempi seguenti mostrano come usare Python a livello di programmazione per creare un URL che concede agli utenti l'accesso diretto alla Console di gestione AWS. Gli esempi sono due:

- Federate tramite richieste GET a AWS
- Federa tramite richieste POST a AWS

Entrambi gli esempi utilizzano l'AssumeRoleAPI AWS SDK per Python (Boto3) and per ottenere credenziali di sicurezza temporanee.

Non includere `SessionDuration` se le credenziali `AssumeRoleSession` provengono dalla concatenazione dei ruoli. Se si include `SessionDuration`, l'operazione avrà esito negativo.

Utilizzo delle richieste GET

```
import urllib, json, sys
import requests # 'pip install requests'
import boto3 # AWS SDK for Python (Boto3) 'pip install boto3'

# Step 1: Authenticate user in your own identity system.

# Step 2: Using the access keys for an IAM user in your Account AWS,
# call "AssumeRole" to get temporary access keys for the role or federated principal

# Note: Calls to AWS STS AssumeRole must be signed using the access key ID
# and secret access key of an IAM user or using existing temporary credentials.
# The credentials can be in Amazon EC2 instance metadata, in environment variables,
# or in a configuration file, and will be discovered automatically by the
# client('sts') function. For more information, see the Python SDK docs:
# http://boto3.readthedocs.io/en/latest/reference/services/sts.html
# http://boto3.readthedocs.io/en/latest/reference/services/
sts.html#STS.Client.assume_role
sts_connection = boto3.client('sts')
```

```
assumed_role_object = sts_connection.assume_role(
    RoleArn="arn:aws:iam::account-id:role/ROLE-NAME",
    RoleSessionName="AssumeRoleSession",
)

# Step 3: Format resulting temporary credentials into JSON
url_credentials = {}
url_credentials['sessionId'] =
 assumed_role_object.get('Credentials').get('AccessKeyId')
url_credentials['sessionKey'] =
 assumed_role_object.get('Credentials').get('SecretAccessKey')
url_credentials['sessionToken'] =
 assumed_role_object.get('Credentials').get('SessionToken')
json_string_with_temp_credentials = json.dumps(url_credentials)

# Step 4. Make request to AWS federation endpoint to get sign-in token. Construct the
 parameter string with
# the sign-in action request, a 12-hour session duration, and the JSON document with
 temporary credentials
# as parameters.
request_parameters = "?Action=getSigninToken"
request_parameters += "&SessionDuration=43200"
if sys.version_info[0] < 3:
    def quote_plus_function(s):
        return urllib.quote_plus(s)
else:
    def quote_plus_function(s):
        return urllib.parse.quote_plus(s)
request_parameters += "&Session=" +
 quote_plus_function(json_string_with_temp_credentials)
request_url = "https://signin.aws.amazon.com/federation" + request_parameters
r = requests.get(request_url)
# Returns a JSON document with a single element named SigninToken.
signin_token = json.loads(r.text)

# Step 5: Create URL where users can use the sign-in token to sign in to
# the console. This URL must be used within 15 minutes after the
# sign-in token was issued.
request_parameters = "?Action=login"
request_parameters += "&Issuer=Example.org"
request_parameters += "&Destination=" + quote_plus_function("https://
console.aws.amazon.com/")
request_parameters += "&SigninToken=" + signin_token["SigninToken"]
```

```
request_url = "https://signin.aws.amazon.com/federation" + request_parameters

# Send final URL to stdout
print (request_url)
```

Utilizzo delle richieste POST

```
import urllib, json, sys
import requests # 'pip install requests'
import boto3 # AWS SDK for Python (Boto3) 'pip install boto3'
import os
from selenium import webdriver # 'pip install selenium', 'brew install chromedriver'

# Step 1: Authenticate user in your own identity system.

# Step 2: Using the access keys for an IAM user in your A Account AWS,
# call "AssumeRole" to get temporary access keys for the role or federated principal

# Note: Calls to AWS STS AssumeRole must be signed using the access key ID
# and secret access key of an IAM user or using existing temporary credentials.
# The credentials can be in Amazon EC2 instance metadata, in environment variables,

# or in a configuration file, and will be discovered automatically by the
# client('sts') function. For more information, see the Python SDK docs:
# http://boto3.readthedocs.io/en/latest/reference/services/sts.html
# http://boto3.readthedocs.io/en/latest/reference/services/
sts.html#STS.Client.assume_role
if sys.version_info[0] < 3:
    def quote_plus_function(s):
        return urllib.quote_plus(s)
else:
    def quote_plus_function(s):
        return urllib.parse.quote_plus(s)

sts_connection = boto3.client('sts')

assumed_role_object = sts_connection.assume_role(
    RoleArn="arn:aws:iam::account-id:role/ROLE-NAME",
    RoleSessionName="AssumeRoleDemoSession",
)

# Step 3: Format resulting temporary credentials into JSON
url_credentials = {}
```

```
url_credentials['sessionId'] =
 assumed_role_object.get('Credentials').get('AccessKeyId')
url_credentials['sessionKey'] =
 assumed_role_object.get('Credentials').get('SecretAccessKey')
url_credentials['sessionToken'] =
 assumed_role_object.get('Credentials').get('SessionToken')
json_string_with_temp_credentials = json.dumps(url_credentials)

# Step 4. Make request to AWS federation endpoint to get sign-in token. Construct the
 parameter string with
# the sign-in action request, a 12-hour session duration, and the JSON document with
 temporary credentials
# as parameters.
request_parameters = {}
request_parameters['Action'] = 'getSigninToken'
request_parameters['SessionDuration'] = '43200'
request_parameters['Session'] = json_string_with_temp_credentials

request_url = "https://signin.aws.amazon.com/federation"
r = requests.post( request_url, data=request_parameters)

# Returns a JSON document with a single element named SigninToken.
signin_token = json.loads(r.text)

# Step 5: Create a POST request where users can use the sign-in token to sign in to
# the console. The POST request must be made within 15 minutes after the
# sign-in token was issued.
request_parameters = {}
request_parameters['Action'] = 'login'
request_parameters['Issuer']='Example.org'
request_parameters['Destination'] = 'https://console.aws.amazon.com/'
request_parameters['SigninToken'] =signin_token['SigninToken']

jsrequest = '''
var form = document.createElement('form');
form.method = 'POST';
form.action = '{request_url}';
request_parameters = {request_parameters}
for (var param in request_parameters) {{
    if (request_parameters.hasOwnProperty(param)) {{
        const hiddenField = document.createElement('input');
        hiddenField.type = 'hidden';
        hiddenField.name = param;
        hiddenField.value = request_parameters[param];
```

```
        form.appendChild(hiddenField);
    }}
}}
document.body.appendChild(form);
form.submit();
'''.format(request_url=request_url, request_parameters=request_parameters)

driver = webdriver.Chrome()
driver.execute_script(jsrequest)
input("Press Enter to close the browser window...")
```

Esempio di codice con Java

L'esempio seguente mostra come usare Java a livello di programmazione per creare un URL che concede agli utenti l'accesso diretto alla Console di gestione AWS. Nel frammento di codice seguente viene utilizzato AWS SDK for Java.

```
import java.net.URLEncoder;
import java.net.URL;
import java.net.URLConnection;
import java.io.BufferedReader;
import java.io.InputStreamReader;
// Available at http://www.json.org/java/index.html
import org.json.JSONObject;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClient;
import com.amazonaws.services.securitytoken.model.Credentials;
import com.amazonaws.services.securitytoken.model.GetFederationTokenRequest;
import com.amazonaws.services.securitytoken.model.GetFederationTokenResult;


/* Calls to AWS STS API operations must be signed using the access key ID
   and secret access key of an IAM user or using existing temporary
   credentials. The credentials should not be embedded in code. For
   this example, the code looks for the credentials in a
   standard configuration file.
*/
AWSCredentials credentials =
  new PropertiesCredentials(
         AwsConsoleApp.class.getResourceAsStream("AwsCredentials.properties"));

AWSSecurityTokenServiceClient stsClient =
```

```
  new AWSSecurityTokenServiceClient(credentials);

GetFederationTokenRequest getFederationTokenRequest =
  new GetFederationTokenRequest();
getFederationTokenRequest.setDurationSeconds(1800);
getFederationTokenRequest.setName("UserName");

// A sample policy for accessing Amazon Simple Notification Service (Amazon SNS) in the
 console.

String policy = "{\"Version\":\"2012-10-17\",       \"Statement\":[{\"Action\":\"sns:*
\"," +
  "\"Effect\":\"Allow\",\"Resource\":\"*\"}]}";

getFederationTokenRequest.setPolicy(policy);

GetFederationTokenResult federationTokenResult =
  stsClient.getFederationToken(getFederationTokenRequest);

Credentials federatedCredentials = federationTokenResult.getCredentials();

// The issuer parameter specifies your internal sign-in
// page, for example https://mysignin.internal.mycompany.com/.
// The console parameter specifies the URL to the destination console of the
// AWS Management Console. This example goes to Amazon SNS.
// The signin parameter is the URL to send the request to.

String issuerURL = "https://mysignin.internal.mycompany.com/";
String consoleURL = "https://console.aws.amazon.com/sns";
String signInURL = "https://signin.aws.amazon.com/federation";

// Create the sign-in token using temporary credentials,
// including the access key ID,  secret access key, and session token.
String sessionJson = String.format(
  "{\"%1$s\":\"%2$s\",\"%3$s\":\"%4$s\",\"%5$s\":\"%6$s\"}",
  "sessionId", federatedCredentials.getAccessKeyId(),
  "sessionKey", federatedCredentials.getSecretAccessKey(),
  "sessionToken", federatedCredentials.getSessionToken());

// Construct the sign-in request with the request sign-in token action, a
// 12-hour console session duration, and the JSON document with temporary
// credentials as parameters.

String getSigninTokenURL = signInURL +
```

```
                            "?Action=getSigninToken" +
                            "&DurationSeconds=43200" +
                            "&SessionType=json&Session=" +
                            URLEncoder.encode(sessionJson,"UTF-8");

URL url = new URL(getSigninTokenURL);

// Send the request to the AWS federation endpoint to get the sign-in token
URLConnection conn = url.openConnection ();

BufferedReader bufferReader = new BufferedReader(new
  InputStreamReader(conn.getInputStream()));
String returnContent = bufferReader.readLine();

String signinToken = new JSONObject(returnContent).getString("SigninToken");

String signinTokenParameter = "&SigninToken=" + URLEncoder.encode(signinToken,"UTF-8");

// The issuer parameter is optional, but recommended. Use it to direct users
// to your sign-in page when their session expires.

String issuerParameter = "&Issuer=" + URLEncoder.encode(issuerURL, "UTF-8");

// Finally, present the completed URL for the AWS console session to the user

String destinationParameter = "&Destination=" + URLEncoder.encode(consoleURL,"UTF-8");
String loginURL = signInURL + "?Action=login" +
                     signinTokenParameter + issuerParameter + destinationParameter;
```

Esempio ci creazione dell'URL (Ruby)

L'esempio seguente mostra come usare Ruby a livello di programmazione per creare un URL che concede agli utenti l'accesso diretto alla Console di gestione AWS. In questo frammento di codice viene utilizzato AWS SDK for Ruby.

```
require 'rubygems'
require 'json'
require 'open-uri'
require 'cgi'
require 'aws-sdk'

# Create a new STS instance
#
```

```
# Note: Calls to AWS STS API operations must be signed using an access key ID
# and secret access key. The credentials can be in EC2 instance metadata
# or in environment variables and will be automatically discovered by
# the default credentials provider in the AWS Ruby SDK.
sts = Aws::STS::Client.new()

# The following call creates a temporary session that returns
# temporary security credentials and a session token.
# The policy grants permissions to work
# in the AWS SNS console.

session = sts.get_federation_token({
  duration_seconds: 1800,
  name: "UserName",
  policy: "{\"Version\":\"2012-10-17\",       \"Statement\":{\"Effect\":\"Allow\",
\"Action\":\"sns:*\",\"Resource\":\"*\"}}",
})

# The issuer value is the URL where users are directed (such as
# to your internal sign-in page) when their session expires.
#
# The console value specifies the URL to the destination console.
# This example goes to the Amazon SNS console.
#
# The sign-in value is the URL of the AWS STS federation endpoint.
issuer_url = "https://mysignin.internal.mycompany.com/"
console_url = "https://console.aws.amazon.com/sns"
signin_url = "https://signin.aws.amazon.com/federation"

# Create a block of JSON that contains the temporary credentials
# (including the access key ID, secret access key, and session token).
session_json = {
  :sessionId => session.credentials[:access_key_id],
  :sessionKey => session.credentials[:secret_access_key],
  :sessionToken => session.credentials[:session_token]
}.to_json

# Call the federation endpoint, passing the parameters
# created earlier and the session information as a JSON block.
# The request returns a sign-in token that's valid for 15 minutes.
# Signing in to the console with the token creates a session
# that is valid for 12 hours.
get_signin_token_url = signin_url +
                       "?Action=getSigninToken" +
```

```
                        "&SessionType=json&Session=" +
                        CGI.escape(session_json)

returned_content = URI.parse(get_signin_token_url).read

# Extract the sign-in token from the information returned
# by the federation endpoint.
signin_token = JSON.parse(returned_content)['SigninToken']
signin_token_param = "&SigninToken=" + CGI.escape(signin_token)

# Create the URL to give to the user, which includes the
# sign-in token and the URL of the console to open.
# The "issuer" parameter is optional but recommended.
issuer_param = "&Issuer=" + CGI.escape(issuer_url)
destination_param = "&Destination=" + CGI.escape(console_url)
login_url = signin_url + "?Action=login" + signin_token_param +
  issuer_param + destination_param
```

Tag per AWS Identity and Access Management risorse

Un tag è un'etichetta di attributi personalizzata assegnata a una risorsa AWS . Ogni tag è costituito da due parti:

- Una chiave di tag (ad esempio, `CostCenter`, `Environment`, `Project` o `Purpose`).
- Un campo facoltativo noto come valore del tag (ad esempio, `111122223333`, `Production` o un nome di team). Non specificare il valore del tag equivale a utilizzare una stringa vuota.

Tutti questi sono noti come coppie chiave-valore. Per i limiti sul numero di tag che è possibile avere sulle risorse IAM, consulta IAM e AWS STS quote.

> **Note**
>
> Per dettagli sulla distinzione tra maiuscole e minuscole per le chiavi dei tag e i valori delle chiavi dei tag, consulta Case sensitivity.

I tag ti aiutano a identificare e organizzare AWS le tue risorse. Molti AWS servizi supportano i tag, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Ad esempio, è possibile assegnare lo stesso tag a un ruolo IAM che si assegna a un bucket

Amazon S3. Per ulteriori informazioni sulle strategie di tagging, consulta la Guida per l'utente delle AWS risorse di tagging.

Oltre a identificare, organizzare e monitorare le risorse IAM con i tag, puoi usare i tag nelle policy IAM per controllare chi può visualizzare e interagire con le risorse. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passa i tag della sessione in AWS STS.

Argomenti

- Scegli un AWS convenzione di denominazione dei tag
- Regole per l'etichettatura in IAM e AWS STS
- Aggiungere tag agli utenti IAM
- Aggiungere tag ai ruoli IAM
- Aggiungere tag alle policy gestite dal cliente
- Aggiungere tag ai provider di identità OpenID Connect (OIDC)
- Aggiungere tag ai provider di identità SAML per IAM
- Aggiunta di tag ai profili dell'istanza per i ruoli Amazon EC2
- Aggiungere tag ai certificati server
- Aggiungere tag ai dispositivi MFA virtuali
- Etichetta le istanze del gestore degli accessi agli account
- Passa i tag della sessione in AWS STS

Scegli un AWS convenzione di denominazione dei tag

Quando si inizia a collegare tag alle risorse IAM, scegli attentamente la convenzione di denominazione dei tag. Applica la stessa convenzione a tutti i AWS tag. Ciò è particolarmente importante se utilizzi i tag nelle policy per controllare l'accesso alle AWS risorse. Se utilizzi già tag in AWS, rivedi la convenzione di denominazione e modificala di conseguenza.

> **Note**
>
> Se il tuo account è membro di AWS Organizations, consulta le politiche sui tag nella guida per l' AWS Organizations utente per saperne di più sull'utilizzo dei tag in AWS Organizations.

Best practice per la denominazione dei tag

Di seguito sono riportate alcune best practice e convenzioni di denominazione per i tag.

Assicurati che i nomi dei tag vengano utilizzati in modo coerente. Ad esempio, i tag `CostCenter` e `costcenter` sono diversi, pertanto uno potrebbe essere configurato come tag di allocazione dei costi per l'analisi e il report finanziario mentre l'altro no. Analogamente, il `Name` tag appare nella AWS Console per molte risorse, ma non è così. `name` Per dettagli sulla distinzione tra maiuscole e minuscole per le chiavi dei tag e i valori delle chiavi dei tag, consulta Case sensitivity.

Alcuni tag sono predefiniti AWS o creati automaticamente da vari AWS servizi. Molti nomi AWS di tag definiti utilizzano solo lettere minuscole, con trattini che separano le parole del nome e prefissi per identificare il servizio di origine del tag. Ad esempio:

- `aws:ec2spot:fleet-request-id` identifica la richiesta di istanza Spot di Amazon EC2 che ha avviato l'istanza.
- `aws:cloudformation:stack-name`identifica lo stack che ha creato la risorsa. CloudFormation
- `elasticbeanstalk:environment-name` identifica l'applicazione che ha creato la risorsa.

Prendi in considerazione la possibilità di assegnare un nome ai tag utilizzando tutte lettere minuscole, con trattini che separano le parole e un prefisso che identifichi il nome dell'organizzazione o il nome abbreviato. Ad esempio, per una società fittizia denominata AnyCompany, potresti definire tag come:

- `anycompany:cost-center` per identificare il codice interno del centro di costo
- `anycompany:environment-type` per identificare se l'ambiente è in fase di sviluppo, test o produzione
- `anycompany:application-id` per identificare l'applicazione per cui è stata creata la risorsa

Il prefisso garantisce che i tag siano chiaramente identificati come definiti dall'organizzazione e non da AWS uno strumento di terze parti che potresti utilizzare. L'uso di tutte le lettere minuscole con i trattini per i separatori evita confusione su come scrivere il nome di un tag. Ad esempio, `anycompany:project-id` è più semplice da ricordare rispetto `ANYCOMPANY:ProjectID`, `anycompany:projectID` oppure `Anycompany:ProjectId`.

Regole per l'etichettatura in IAM e AWS STS

Un certo numero di convenzioni regola la creazione e l'applicazione di tag in IAM e AWS STS.

Denominazione di tag

Osserva le seguenti convenzioni quando formuli una convenzione di denominazione dei tag per le risorse IAM, le sessioni AWS STS assume-role e le sessioni utente federate: AWS STS

Requisiti per i caratteri: chiavi e valori di tag possono includere qualsiasi combinazione di lettere, numeri, spazi e simboli _ . : / = + - @.

Distinzione tra maiuscole e minuscole: la distinzione tra maiuscole e minuscole per le chiavi di tag varia a seconda del tipo di risorsa IAM taggata. I valori chiave dei tag per utenti e ruoli IAM non distingue tra maiuscole e minuscole, anche se questi caratteri vengono mantenuti. Questo significa che non è possibile avere le chiavi di tag **`Department`** e **`department`** separate. Se hai assegnato a un utente il tag **`Department=finance`** e aggiungi il tag **`department=hr`**, quest'ultimo sostituirà il primo tag. Non viene aggiunto un secondo tag.

Per gli altri tipi di risorse IAM, i valori della chiave di tag fanno distinzione tra maiuscole e minuscole. Questo significa che è possibile avere chiavi di tag **`Costcenter`** e **`costcenter`** separate. Ad esempio, se sono stati applicati i tag a una policy gestita dal cliente con il tag **`Costcenter = 1234`** e si aggiunge il tag **`costcenter = 5678`**, il criterio avrà entrambe le chiavi di tag **`Costcenter`** e **`costcenter`**.

Come best practice, si consiglia di evitare l'uso di tag simili con un'applicazione di maiuscole e minuscole non coerente. Consigliamo di definire una strategia per l'uso delle lettere maiuscole e minuscole nei tag e implementarla in modo coerente per tutti i tipi di risorse. Per saperne di più sulle migliori pratiche per il tagging, consulta Tagging Resources nel. AWS Riferimenti generali di AWS

Negli elenchi seguenti vengono illustrate le differenze nella distinzione tra maiuscole e minuscole per le chiavi di tag associate alle risorse IAM.

I valori delle chiavi tag non fanno distinzione tra maiuscole e minuscole:

- Ruoli IAM
- Utenti IAM

Le chiavi e i valori fanno distinzione tra maiuscole e minuscole.

- Policy gestite dal cliente
- Profili delle istanze
- Provider di identità OpenID Connect

- Provider di identità SAML
- Certificati server
- Dispositivi MFA virtuali

Inoltre, valgono le seguenti regole:

- Non è possibile creare una chiave o un valore di tag che inizi con il testo **`aws:`**. Questo prefisso del tag è riservato per AWS uso interno.
- È possibile creare un tag con un valore vuoto, ad esempio **`phoneNumber =`** . Non è possibile creare una chiave di tag vuota.
- Non è possibile specificare più valori in un singolo tag, ma è possibile creare una struttura multivalore personalizzata nel singolo valore. Ad esempio, supponiamo che l'utente Zhang lavori nel team di progettazione e nel team di QA. Se colleghi il tag **`team = Engineering`** e poi colleghi il tag **`team = QA`**, modifichi il valore del tag da **`Engineering`** a **`QA`**. Al contrario, è possibile includere più valori in un singolo tag con un separatore personalizzato. In questo esempio, è possibile collegare il tag **`team = Engineering:QA`** a Zhang.

> **Note**
>
> Per controllare l'accesso al team di progettazione in questo esempio utilizzando il tag **`team`**, è necessario creare una policy che consenta ogni configurazione che potrebbe includere **`Engineering`**, tra cui **`Engineering:QA`**. Per ulteriori informazioni sull'utilizzo dei tag nelle policy, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Applicazione e modifica di tag

Osserva le seguenti convenzioni quando applichi i tag alle risorse IAM:

- Puoi applicare tag alla maggior parte delle risorse IAM, ma non a gruppi, ruoli assunti, report di accesso o dispositivi MFA basati su hardware.
- Non è possibile utilizzare l'editor di tag per applicare i tag alle risorse IAM. L'editor di tag non supporta i tag IAM. Per ulteriori informazioni sull'utilizzo dell'editor di tag con altri servizi, consulta l'articolo relativo all'utilizzo dell'editor di tag nella Guida per l'utente della AWS Resource Groups .
- Per aggiungere i tag a una risorsa IAM, devi disporre di autorizzazioni specifiche. Per applicare o rimuovere i tag dalle risorse, è necessario disporre anche dell'autorizzazione per elencare i tag. Per

ulteriori informazioni, consulta l'elenco degli argomenti relativi a ciascuna risorsa IAM alla fine di questa pagina.

- Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.
- Puoi applicare lo stesso tag a più risorse IAM. Ad esempio, si supponga di avere un reparto denominato `AWS_Development` con 12 membri. È possibile avere 12 utenti e un ruolo con **`department`** come chiave del tag e **`awsDevelopment`** come valore (**`department = awsDevelopment`**). Puoi inoltre utilizzare lo stesso tag su risorse in altri servizi che supportano il tagging.
- Le entità IAM (utenti o ruoli) non possono avere più istanze della stessa chiave di tag. Ad esempio, se disponi di un utente con la coppia chiave-valore del tag **`costCenter = 1234`**, puoi collegare la coppia chiave-valore del tag **`costCenter = 5678`**. IAM aggiorna il valore del tag **`costCenter`** a **`5678`**.
- Per modificare un tag collegato a un'entità IAM (utente o ruolo), collega un tag con un nuovo valore per sovrascrivere il tag esistente. Ad esempio, supponiamo che tu disponga di un utente con la coppia chiave-valore del tag **`department = Engineering`**. Se devi spostare l'utente nel reparto QA, puoi collegare la coppia chiave-valore del tag **`department = QA`** all'utente. In questo modo il valore **`Engineering`** della chiave di tag **`department`** viene sostituito con il valore **`QA`**.

Aggiungere tag agli utenti IAM

Puoi utilizzare coppie chiave-valore di tag IAM per aggiungere attributi personalizzati a un utente IAM. Ad esempio, per aggiungere informazioni sulla posizione per un utente, puoi aggiungere la chiave di tag **`location`** e il valore di tag **`us_wa_seattle`**. In alternativa, puoi utilizzare tre coppie chiave-valore del tag per la posizione separate: **`loc-country = us`**, **`loc-state = wa`** e **`loc-city = seattle`**. È possibile usare i tag per controllare l'accesso di un utente alle risorse o per controllare quali tag possono essere collegati a un utente. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passa i tag della sessione in AWS STS.

Autorizzazioni necessarie per il tagging degli utenti IAM

Per consentire a un utente IAM di aggiungere tag ad altri utenti, è necessario configurare le autorizzazioni relative. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListUserTags`
- `iam:TagUser`
- `iam:UntagUser`

Come consentire a un utente IAM di aggiungere, elencare o rimuovere un tag per un utente specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'utente IAM che deve gestire i tag. Usa il tuo numero di account e sostituiscilo *<username>* con il nome dell'utente i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser",
        "iam:UntagUser"
    ],
    "Resource": "arn:aws:iam::<account-number>:user/<username>"
}
```

Come consentire a un utente IAM di gestire autonomamente i tag

Aggiungi l'istruzione seguente alla policy di autorizzazione per gli utenti per consentire loro di gestire i propri tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser",
        "iam:UntagUser"
    ],
    "Resource": "arn:aws:iam::user/${aws:username}"
}
```

Come consentire a un utente IAM di aggiungere un tag a un utente specifico

Aggiungi l'istruzione seguente alla policy delle autorizzazioni relativa all'utente IAM che potrà aggiungere ma non rimuovere i tag di un determinato utente.

> **Note**
>
> L'azione `iam:TagUser` richiede che tu includa anche l'azione `iam:ListUserTags`.

Per utilizzare questa politica, *<username>* sostituiscila con il nome dell'utente i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser"
    ],
    "Resource": "arn:aws:iam::<account-number>:user/<username>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita, ad esempio IAMFullAccess per fornire l'accesso completo a IAM.

Gestione dei tag degli utenti IAM (console)

Puoi gestire i tag per gli utenti IAM dalla Console di gestione AWS.

Per gestire i tag degli utenti (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegliere Users (Utenti) e selezionare il nome dell'utente da modificare.
3. Scegliere la scheda Tags (Tag) e completare una delle seguenti operazioni:
   - Scegli Add new tag (Aggiungi nuovo tag) se all'utente non sono ancora stati assegnati tag.
   - Scegli Manage tags (Gestisci tag) per gestire il set di tag esistente.

4. Aggiungere o rimuovere i tag per completare il set di tag. Selezionare quindi Save changes (Salva modifiche).

Gestione dei tag sugli utenti IAM (AWS CLI or AWS API)

Puoi elencare, collegare o rimuovere i tag per gli utenti IAM. Puoi utilizzare AWS CLI o l' AWS API per gestire i tag per gli utenti IAM.

Per elencare i tag attualmente associati a un utente IAM (AWS CLI or AWS API)

- AWS CLI: aws iam list-user-tags
- AWS API: ListUserTags

Per allegare tag a un utente IAM (AWS CLI or AWS API)

- AWS CLI: aws iam tag-user
- AWS API: TagUser

Per rimuovere i tag da un utente IAM (AWS CLI or AWS API)

- AWS CLI: aws iam untag-user
- AWS API: UntagUser

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Aggiungere tag ai ruoli IAM

Puoi utilizzare coppie chiave-valore di tag IAM per aggiungere attributi personalizzati a un ruolo IAM. Ad esempio, per aggiungere informazioni sulla posizione per un ruolo, puoi aggiungere la chiave tag **`location`** e il valore tag **`us_wa_seattle`**. In alternativa, puoi utilizzare tre coppie chiave-valore del tag per la posizione separate: **`loc-country = us`**, **`loc-state = wa`** e **`loc-city = seattle`**. Puoi utilizzare i tag per controllare l'accesso di un ruolo alle risorse o per controllare

quali tag possono essere collegati a un ruolo. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passa i tag della sessione in AWS STS.

Autorizzazioni necessarie per il tagging dei ruoli IAM

Per permettere a un ruolo IAM di applicare tag ad altre entità (utenti o ruoli), devi configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListRoleTags`
- `iam:TagRole`
- `iam:UntagRole`
- `iam:ListUserTags`
- `iam:TagUser`
- `iam:UntagUser`

Come consentire a un ruolo IAM di aggiungere, elencare o rimuovere un tag per un utente specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per il ruolo IAM che deve gestire i tag. Usa il tuo numero di account e sostituiscilo *<username>* con il nome dell'utente i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser",
        "iam:UntagUser"
    ],
    "Resource": "arn:aws:iam::<account-number>:user/<username>"
}
```

Come consentire a un ruolo IAM di aggiungere un tag a un utente specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione relativa al ruolo IAM che potrà aggiungere ma non rimuovere i tag di un utente specifico.

Per utilizzare questa politica, *<username>* sostituiscila con il nome dell'utente i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser"
    ],
    "Resource": "arn:aws:iam::<account-number>:user/<username>"
}
```

Come consentire a un ruolo IAM di aggiungere, elencare o rimuovere un tag per un ruolo specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per il ruolo IAM che deve gestire i tag. Sostituisci *<rolename>* con il nome del ruolo i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListRoleTags",
        "iam:TagRole",
        "iam:UntagRole"
    ],
    "Resource": "arn:aws:iam::<account-number>:role/<rolename>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita, ad esempio IAMFullAccess per fornire l'accesso completo a IAM.

Gestione dei tag sui ruoli IAM (console)

Puoi gestire i tag per i ruoli IAM dalla Console di gestione AWS.

Per gestire i tag sui ruoli (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegliere Roles (Ruoli) e selezionare il nome del ruolo che si desidera modificare.
3. Scegliere la scheda Tags (Tag) e completare una delle seguenti operazioni:
   - Scegli Add new tag (Aggiungi nuovo tag) se al ruolo non sono ancora stati assegnati tag.
   - Scegli Manage tags (Gestisci tag) per gestire il set di tag esistente.
4. Aggiungere o rimuovere i tag per completare il set di tag. Quindi, scegli Save changes (Salva modifiche).

Gestione dei tag sui ruoli IAM (AWS CLI or AWS API)

Puoi elencare, collegare o rimuovere i tag per i ruoli IAM. Puoi utilizzare AWS CLI o l' AWS API per gestire i tag per i ruoli IAM.

Per elencare i tag attualmente associati a un ruolo IAM (AWS CLI or AWS API)

- AWS CLI: aws iam list-role-tags
- AWS API: ListRoleTags

Per allegare tag a un ruolo IAM (AWS CLI or AWS API)

- AWS CLI: aws iam tag-role
- AWS API: TagRole

Per rimuovere i tag da un ruolo IAM (AWS CLI or AWS API)

- AWS CLI: aws iam untag-role
- AWS API: UntagRole

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Aggiungere tag alle policy gestite dal cliente

È possibile utilizzare le coppie chiave-valore del tag IAM per aggiungere attributi personalizzati alle policy gestite dal cliente. Ad esempio, per applicare un tag a una policy con le informazioni sul reparto, puoi aggiungere la chiave tag **`Department`** e il valore tag **`eng`**. In alternativa, è possibile contrassegnare i criteri per indicare che si riferiscono a un ambiente specifico, ad esempio **`Environment = lab`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati a una risorsa. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passa i tag della sessione in AWS STS.

Autorizzazioni necessarie per applicare i tag alle policy gestite dai clienti

È necessario configurare le autorizzazioni per consentire a un'entità IAM (utenti o ruoli) di applicare i tag alle policy gestite dal cliente. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListPolicyTags`
- `iam:TagPolicy`
- `iam:UntagPolicy`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per una policy gestita dal cliente

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituiscilo *`<policyname>`* con il nome della policy i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
```

```
        "iam:ListPolicyTags",
        "iam:TagPolicy",
        "iam:UntagPolicy"
    ],
    "Resource": "arn:aws:iam::<account-number>:policy/<policyname>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a una determinata policy gestita dal cliente

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per una policy specifica.

> **Note**
>
> L'azione `iam:TagPolicy` richiede che tu includa anche l'azione `iam:ListPolicyTags`.

Per utilizzare questa policy, *<policyname>* sostituiscila con il nome della policy i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListPolicyTags",
        "iam:TagPolicy"
    ],
    "Resource": "arn:aws:iam::<account-number>:policy/<policyname>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita, ad esempio IAMFullAccess per fornire l'accesso completo a IAM.

Gestione dei tag nelle policy gestiti dal cliente IAM (console)

Puoi gestire i tag delle policy gestite dal cliente IAM dalla Console di gestione AWS.

Per gestire i tag nelle policy gestite dal cliente (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegliere Policies (Policy) e selezionare il nome della policy gestita dal cliente che si desidera modificare.
3. Scegli la scheda Tag, quindi scegli Gestisci tag.
4. Aggiungere o rimuovere i tag per completare il set di tag. Selezionare quindi Save changes (Salva modifiche).

Gestione dei tag nelle policy gestite dai clienti IAM (AWS CLI or AWS API)

Puoi elencare, allegare o rimuovere i tag per le policy gestite dal cliente IAM. Puoi utilizzare AWS CLI o l' AWS API per gestire i tag per le policy gestite dai clienti IAM.

Per elencare i tag attualmente associati a una policy gestita dal cliente IAM (AWS CLI or AWS API)

- AWS CLI: aws iam list-policy-tags
- AWS API: ListPolicyTags

Per allegare tag a una policy gestita dal cliente IAM (AWS CLI or AWS API)

- AWS CLI: aws iam tag-policy
- AWS API: TagPolicy

Per rimuovere i tag da una policy gestita dal cliente IAM (AWS CLI or AWS API)

- AWS CLI: aws iam untag-policy
- AWS API: UntagPolicy

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Aggiungere tag ai provider di identità OpenID Connect (OIDC)

Puoi utilizzare le coppie chiave-valore dei tag IAM per aggiungere attributi personalizzati ai provider di identità OpenID Connect (OIDC) IAM. Ad esempio, per identificare un provider di identità OIDC, puoi aggiungere la chiave tag **`google`** e il valore tag **`oidc`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati a un oggetto. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Autorizzazioni necessarie per l'aggiunta di tag ai provider di identità OIDC IAM

Per permettere a un'entità IAM (utente o ruolo) di applicare tag ai provider di identità OIDC IAM, devi configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListOpenIDConnectProviderTags`
- `iam:TagOpenIDConnectProvider`
- `iam:UntagOpenIDConnectProvider`

Per consentire a un'entità IAM di aggiungere, elencare o rimuovere un tag per un provider di identità OIDC IAM

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituiscilo *<OIDCProviderName>* con il nome del provider OIDC i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListOpenIDConnectProviderTags",
        "iam:TagOpenIDConnectProvider",
        "iam:UntagOpenIDConnectProvider"
    ],
    "Resource": "arn:aws:iam::<account-number>:oidc-provider/<OIDCProviderName>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un provider di identità OIDC IAM specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per un determinato provider di identità.

> **Note**
>
> L'azione `iam:TagOpenIDConnectProvider` richiede che tu includa anche l'azione `iam:ListOpenIDConnectProviderTags`.

Per utilizzare questa politica, sostituiscila *<OIDCProviderName>* con il nome del provider OIDC i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListOpenIDConnectProviderTags",
        "iam:TagOpenIDConnectProvider"
    ],
    "Resource": "arn:aws:iam::<account-number>:oidc-provider/<OIDCProviderName>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita, ad esempio IAMFullAccess per fornire l'accesso completo a IAM.

Gestione dei tag nei provider di identità OIDC IAM (console)

Puoi gestire i tag per i provider di identità OIDC IAM dalla Console di gestione AWS.

Per gestire i tag dei provider di identità OIDC (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegli Identity providers (Provider di identità), quindi scegli il nome del provider di identità che desideri aggiornare.

3. Selezionare la scheda Tag, quindi nella sezione Tag scegli Gestisci tag e completa una delle seguenti azioni:
   - Sceglie Add tag (Aggiungi tag) se il provider di identità OIDC non dispone ancora di tag o per aggiungere un nuovo tag.
   - Modificare le chiavi e i valori dei tag esistenti.
   - Per rimuovere un tag, scegli Remove tag (Rimuovi tag).
4. Selezionare quindi Save changes (Salva modifiche).

Gestione dei tag sui provider di identità OIDC IAM (AWS CLI or AWS API)

Puoi elencare, allegare o rimuovere i tag per i provider di identità OIDC IAM. Puoi utilizzare AWS CLI o l' AWS API per gestire i tag per i provider di identità IAM OIDC.

Per elencare i tag attualmente associati a un provider di identità OIDC IAM (AWS CLI or AWS API)

- AWS CLI: aws iam list-open-id-connect-provider-tags
- AWS API: ListOpenIDConnectProviderTags

Per allegare tag a un provider di identità IAM OIDC (AWS CLI or AWS API)

- AWS CLI: aws iam tag-open-id-connect-provider
- AWS API: TagOpenIDConnectProvider

Per rimuovere i tag da un provider di identità IAM OIDC (AWS CLI or AWS API)

- AWS CLI: aws iam untag-open-id-connect-provider
- AWS API: UntagOpenIDConnectProvider

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Aggiungere tag ai provider di identità SAML per IAM

È possibile utilizzare coppie chiave-valore del tag IAM per aggiungere attributi personalizzati ai provider di identità SAML. Ad esempio, per identificare un provider, puoi aggiungere la chiave tag **`okta`** e il valore tag **`saml`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati a un oggetto. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Autorizzazioni necessarie per l'assegnazione di tag ai provider di identità SAML

È necessario configurare le autorizzazioni per consentire a un'entità IAM (utenti o ruoli) di etichettare gli Identity Provider basati su SAML 2.0 (). IdPs Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListSAMLProviderTags`
- `iam:TagSAMLProvider`
- `iam:UntagSAMLProvider`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per un provider di identità SAML

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituiscilo *<SAMLProviderName>* con il nome del provider SAML i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListSAMLProviderTags",
        "iam:TagSAMLProvider",
        "iam:UntagSAMLProvider"
    ],
    "Resource": "arn:aws:iam::<account-number>:saml-provider/<SAMLProviderName>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un provider di identità SAML specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per un determinato provider SAML.

> **Note**
>
> L'azione `iam:TagSAMLProvider` richiede che tu includa anche l'azione `iam:ListSAMLProviderTags`.

Per utilizzare questa policy, *<SAMLProviderName>* sostituiscila con il nome del provider SAML i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListSAMLProviderTags",
        "iam:TagSAMLProvider"
    ],
    "Resource": "arn:aws:iam::<account-number>:saml-provider/<SAMLProviderName>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita, ad esempio IAMFullAccess per fornire l'accesso completo a IAM.

Gestione dei tag nei provider di identità SAML IAM (console)

È possibile gestire i tag per i provider di identità SAML IAM dalla Console di gestione AWS.

Per gestire i tag dei provider di identità SAML (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegli Identity providers (Provider di identità), quindi scegli il nome del provider di identità SAML che desideri aggiornare.
3. Selezionare la scheda Tag, quindi nella sezione Tag scegli Gestisci tag e completa una delle seguenti azioni:

- Sceglie Add tag (Aggiungi tag) se il provider di identità SAML non dispone ancora di tag o per aggiungere un nuovo tag.
- Modificare le chiavi e i valori dei tag esistenti.
- Per rimuovere un tag, scegli Remove tag (Rimuovi tag).

4. Aggiungere o rimuovere i tag per completare il set di tag. Selezionare quindi Save changes (Salva modifiche).

Gestione dei tag sui provider di identità SAML IAM (AWS CLI or AWS API)

Puoi elencare, allegare o rimuovere i tag per i provider di identità SAML IAM. Puoi utilizzare AWS CLI o l' AWS API per gestire i tag per i provider di identità SAML IAM.

Per elencare i tag attualmente associati a un provider di identità SAML (AWS CLI or AWS API)

- AWS CLI: aws iam list-saml-provider-tags
- AWS API: ListSAMLProviderTags

Per allegare tag a un provider di identità SAML (AWS CLI or AWS API)

- AWS CLI: aws iam tag-saml-provider
- AWS API: tagSAMLProvider

Per rimuovere i tag da un provider di identità SAML (AWS CLI or AWS API)

- AWS CLI: aws iam untag-saml-provider
- AWS API: UntagSAMLProvider

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Aggiunta di tag ai profili dell'istanza per i ruoli Amazon EC2

Quando avvii un'istanza Amazon EC2, devi specificare un ruolo IAM da associare ad essa. Un profilo dell'istanza è un container per un ruolo IAM che puoi utilizzare per inoltrare informazioni sul ruolo a un'istanza Amazon EC2 quando questa viene avviata. Puoi etichettare i profili delle istanze quando usi l' AWS API AWS CLI or.

Puoi utilizzare coppie chiave-valore del tag IAM per aggiungere attributi personalizzati a un profilo dell'istanza. Ad esempio, per aggiungere informazioni di reparto a un profilo di istanza, è possibile aggiungere la chiave tag **`access-team`** e il valore tag **`eng`**. In questo modo i principali con tag corrispondenti possono accedere ai profili dell'istanza con lo stesso tag. È possibile utilizzare più coppie chiave-valore del tag per specificare un team e un progetto: **`access-team = eng`** e **`project = peg`**. È possibile usare i tag per controllare l'accesso di un utente alle risorse o per controllare quali tag possono essere collegati a un utente. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passa i tag della sessione in AWS STS.

Autorizzazioni necessarie per il tagging dei profili dell'istanza

Per permettere a un'entità IAM (utente o ruolo) di aggiungere tag ai profili dell'istanza, devi configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListInstanceProfileTags`
- `iam:TagInstanceProfile`
- `iam:UntagInstanceProfile`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per un profilo dell'istanza

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituiscilo *`<InstanceProfileName>`* con il nome del profilo dell'istanza i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListInstanceProfileTags",
        "iam:TagInstanceProfile",
        "iam:UntagInstanceProfile"
    ],
    "Resource": "arn:aws:iam::<account-number>:instance-profile/<InstanceProfileName>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un determinato profilo dell'istanza

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per un determinato profilo dell'istanza.

> **Note**
>
> L'azione `iam:TagInstanceProfile` richiede che tu includa anche l'azione `iam:ListInstanceProfileTags`.

Per utilizzare questa policy, *<InstanceProfileName>* sostituiscila con il nome del profilo dell'istanza i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListInstanceProfileTags",
        "iam:TagInstanceProfile"
    ],
    "Resource": "arn:aws:iam::<account-number>:instance-profile/<InstanceProfileName>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita, ad esempio IAMFullAccess per fornire l'accesso completo a IAM.

Gestione dei tag sui profili delle istanze (AWS CLI or AWS API)

È possibile elencare, allegare o rimuovere i tag dei profili dell'istanza. Puoi utilizzare AWS CLI o l' AWS API per gestire i tag per i profili di istanza.

Per elencare i tag attualmente associati a un profilo di istanza (AWS CLI or AWS API)

- AWS CLI: aws iam list-instance-profile-tags
- AWS API: ListInstanceProfileTags

Per allegare tag a un profilo di istanza (AWS CLI or AWS API)

- AWS CLI: aws iam tag-instance-profile
- AWS API: TagInstanceProfile

Per rimuovere i tag dal profilo di un'istanza (AWS CLI or AWS API)

- AWS CLI: aws iam untag-instance-profile
- AWS API: UntagInstanceProfile

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Aggiungere tag ai certificati server

Se utilizzi IAM per gestire i SSL/TLS certificati, puoi etichettare i certificati del server in IAM utilizzando l'API AWS CLI or AWS . Per i certificati in una regione supportata da AWS Certificate Manager (ACM), ti consigliamo di utilizzare ACM anziché IAM per effettuare il provisioning, gestire e distribuire i certificati del server. Nelle regioni non supportate, è necessario utilizzare IAM come gestore di certificati. Per informazioni sulle regioni supportate da ACM, consulta Endpoint e quote di AWS Certificate Manager nella Riferimenti generali di AWS.

Puoi utilizzare coppie chiave-valore del tag IAM per aggiungere attributi personalizzati a un certificato server. Ad esempio, per aggiungere informazioni sul proprietario o sull'amministratore di un certificato server, aggiungi la chiave tag **`owner`** e il valore tag **`net-eng`**. In alternativa, puoi specificare un

centro di costo aggiungendo la chiave tag **`CostCenter`** e il valore tag **`1234`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati alle risorse. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passa i tag della sessione in AWS STS.

Autorizzazioni necessarie per l'assegnazione di tag ai certificati server

Per permettere a un'entità IAM (utente o ruolo) di aggiungere i tag ai certificati server, devi configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListServerCertificateTags`
- `iam:TagServerCertificate`
- `iam:UntagServerCertificate`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per un certificato server

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituiscilo *`<CertificateName>`* con il nome del certificato del server i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListServerCertificateTags",
        "iam:TagServerCertificate",
        "iam:UntagServerCertificate"
    ],
    "Resource": "arn:aws:iam::<account-number>:server-certificate/<CertificateName>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un determinato certificato server

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per un certificato server specifico.

> **Note**
>
> L'azione `iam:TagServerCertificate` richiede che tu includa anche l'azione `iam:ListServerCertificateTags`.

Per utilizzare questa politica, *<CertificateName>* sostituiscila con il nome del certificato del server i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListServerCertificateTags",
        "iam:TagServerCertificate"
    ],
    "Resource": "arn:aws:iam::<account-number>:server-certificate/<CertificateName>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita, ad esempio IAMFullAccess per fornire l'accesso completo a IAM.

Gestione dei tag sui certificati del server (AWS CLI or AWS API)

È possibile elencare, allegare o rimuovere i certificati server. È possibile utilizzare AWS CLI o l' AWS API per gestire i tag per i certificati del server.

Per elencare i tag attualmente allegati a un certificato del server (AWS CLI or AWS API)

- AWS CLI: aws iam list-server-certificate-tags
- AWS API: ListServerCertificateTags

Per allegare tag a un certificato del server (AWS CLI or AWS API)

- AWS CLI: aws iam tag-server-certificate

- AWS API: TagServerCertificate

Per rimuovere i tag da un certificato del server (AWS CLI or AWS API)

- AWS CLI: aws iam untag-server-certificate
- AWS API: UntagServerCertificate

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Aggiungere tag ai dispositivi MFA virtuali

Puoi utilizzare coppie chiave-valore di tag IAM per aggiungere attributi personalizzati a un dispositivo MFA virtuale. Ad esempio, per aggiungere informazioni sul centro di costo per il dispositivo MFA virtuale di un utente, puoi aggiungere il tag con chiave **`CostCenter`** e il tag con valore **`1234`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati a un oggetto. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

È inoltre possibile utilizzare i tag AWS STS per aggiungere attributi personalizzati quando si assume un ruolo o si federa un utente. Per ulteriori informazioni, consulta Passa i tag della sessione in AWS STS.

Autorizzazioni necessarie per la gestione dei tag dei dispositivi MFA virtuali

Per consentire a un'entità IAM (utente o ruolo) di aggiungere i tag ai dispositivi MFA virtuali, è necessario configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListMFADeviceTags`
- `iam:TagMFADevice`
- `iam:UntagMFADevice`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per un dispositivo MFA virtuale

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituiscilo *<MFATokenID>* con il nome del dispositivo MFA virtuale i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListMFADeviceTags",
        "iam:TagMFADevice",
        "iam:UntagMFADevice"
    ],
    "Resource": "arn:aws:iam::<account-number>:mfa/<MFATokenID>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un determinato dispositivo MFA virtuale

Aggiungi l'istruzione seguente alla policy di autorizzazione dell'entità IAM che potrà aggiungere ma non rimuovere i tag per uno specifico dispositivo MFA virtuale.

> **Note**
>
> L'azione `iam:TagMFADevice` richiede che tu includa anche l'azione `iam:ListMFADeviceTags`.

Per utilizzare questa politica, *<MFATokenID>* sostituiscila con il nome del dispositivo MFA virtuale i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListMFADeviceTags",
        "iam:TagMFADevice"
    ],
    "Resource": "arn:aws:iam::<account-number>:mfa/<MFATokenID>"
```

```
}
```

In alternativa, puoi utilizzare una policy AWS gestita, ad esempio IAMFullAccess per fornire l'accesso completo a IAM.

Gestione dei tag sui dispositivi MFA virtuali (AWS CLI or AWS API)

È possibile elencare, allegare o rimuovere i tag di un dispositivo MFA virtuale. È possibile utilizzare AWS CLI o l' AWS API per gestire i tag per un dispositivo MFA virtuale.

Per elencare i tag attualmente collegati a un dispositivo MFA virtuale (AWS CLI or AWS API)

- AWS CLI: aws iam list-mfa-device-tags
- AWS API: ListMFADeviceTags

Per allegare tag a un dispositivo MFA virtuale (AWS CLI or AWS API)

- AWS CLI: aws iam tag-mfa-device
- AWS API: tagMFADevice

Per rimuovere i tag da un dispositivo MFA virtuale (AWS CLI or AWS API)

- AWS CLI: aws iam untag-mfa-device
- AWS API: UntagMFADevice

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Etichetta le istanze del gestore degli accessi agli account

È necessario configurare le autorizzazioni per consentire a un'entità IAM (ruoli o utenti) di etichettare le istanze di Account Access Manager. Puoi specificare una o tutte le seguenti azioni relative ai tag di accesso all'account in una policy IAM:

- `account-access:TagResource`

- `account-access:ListTagsForResource`
- `account-access:UntagResource`

Per consentire a un'entità IAM (ruolo o utente) di aggiungere, elencare o rimuovere un tag per un'istanza di account access manager

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il numero di account e gli ID di istanza del tuo IAM Identity Center e delle istanze dell'account access manager per sostituire i segnaposto. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creare policy IAM (console)".

```
{
    "Effect": "Allow",
    "Action": [
        "account-access:TagResource",
        "account-access:ListTagsForResource",
        "account-access:UntagResource"
    ],
    "Resource": "<account_access_manager_ARN>"
}
```

Per consentire a un'entità IAM (ruolo o utente) di aggiungere un tag a una specifica istanza di account access manager

Aggiungi la seguente dichiarazione alla politica delle autorizzazioni per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per una specifica istanza di account access manager.

> **Note**
>
> L'azione `account-access:TagResource` richiede che tu includa anche l'azione `account-access:ListTagsForResource`.

Usa il numero di account e gli ID di istanza del tuo IAM Identity Center e delle istanze dell'account access manager per sostituire i segnaposto. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creare policy IAM (console)".

```
{
    "Effect": "Allow",
    "Action": [
        "account-access:TagResource",
        "account-access:ListTagsForResource"
    ],
    "Resource": "<account_access_manager_ARN>"
}
```

Gestione dei tag sull'account access manager (console)

È possibile gestire i tag per un'istanza di account access manager da Console di gestione AWS.

Per gestire i tag su Account Access Manager (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console, scegli Account access manager, quindi scegli la scheda Impostazioni.
3. Nella sezione Tag, puoi vedere i tag attualmente configurati. Scegli Gestisci tag e quindi completa una delle seguenti azioni:
   - Scegli Aggiungi nuovo tag per aggiungere un nuovo tag.
   - Modificare le chiavi e i valori dei tag esistenti.
   - Scegli Rimuovi per rimuovere un tag.
4. Aggiungere o rimuovere i tag per completare il set di tag. Selezionare quindi Save changes (Salva modifiche).

Gestione dei tag su Account Access Manager (AWS CLI or AWS API)

Puoi elencare, allegare o rimuovere i tag per le istanze di Account Access Manager utilizzando l' AWS API AWS CLI or.

Per elencare i tag attualmente associati a un'istanza (AWS CLI o AWS API) di account access manager

- AWS CLI: `aws account-access list-tags-for-resource`
- AWS API: `account-access:ListTagsForResource`

Per allegare tag a un'istanza (AWS CLI o AWS API) di account access manager

- AWS CLI: `aws account-access tag-resource`
- AWS API: `TagResource`

Per rimuovere i tag da un'istanza (AWS CLI o AWS API) dell'account access manager

- AWS CLI: `aws account-access untag-resource`
- AWS API: `UntagResource`

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta the section called "Variabili e tag".

Passa i tag della sessione in AWS STS

I tag di sessione sono attributi di coppia chiave-valore che vengono passati quando si assume un ruolo IAM o si federa un utente in AWS STS. Puoi farlo effettuando una richiesta AWS CLI o AWS API tramite AWS STS o tramite il tuo provider di identità (IdP). Quando utilizzi AWS STS per richiedere credenziali di sicurezza temporanee, generi una sessione. Le sessioni scadono e dispongono di credenziali, ad esempio una coppia di chiavi di accesso e un token di sessione. Quando si utilizzano le credenziali di sessione per effettuare una richiesta successiva, il contesto della richiesta include la chiave di contesto `aws:PrincipalTag`. È possibile utilizzare la chiave `aws:PrincipalTag` nell'elemento `Condition` delle proprie policy per consentire o negare l'accesso in base a tali tag.

Quando si utilizzano credenziali temporanee per effettuare una richiesta, l'entità potrebbe includere un set di tag. Questi tag provengono dalle seguenti fonti:

1. Tag di sessione: i tag passati quando assumi il ruolo o federi l'utente utilizzando l'API AWS CLI or AWS . Per ulteriori informazioni su queste operazioni, consulta Operazioni di tagging di sessione.
2. Tag di sessione transitivi in ingresso: questi tag sono stati ereditati da una sessione precedente in un concatenamento di ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione più avanti in questo argomento.
3. Tag IAM: i tag associati al ruolo IAM assunto.

Argomenti

- Operazioni di tagging di sessione
- Cose da sapere sui tag di sessione
- Autorizzazioni necessarie per aggiungere tag di sessione
- Passaggio dei tag di sessione utilizzando AssumeRole
- Passaggio dei tag di sessione utilizzando AssumeRoleWithSAML
- Passaggio dei tag di sessione utilizzando AssumeRoleWithWebIdentity
- Passaggio dei tag di sessione utilizzando GetFederationToken
- Concatenamento dei ruoli con i tag di sessione
- Utilizzo dei tag di sessione per ABAC
- Visualizzazione dei tag di sessione in CloudTrail

Operazioni di tagging di sessione

Puoi passare i tag di sessione utilizzando le seguenti operazioni AWS CLI o AWS API in AWS STS. La funzione Console di gestione AWS Cambia ruolo non consente di passare i tag di sessione.

È inoltre possibile impostare i tag di sessione come transitivi. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Nella tabella seguente vengono confrontati i metodi per il passaggio dei tag di sessione.

Operation	Chi può assumere il ruolo	Metodo di passaggio dei tag	Metodo di impostazione dei tag transitivi
`assume-role` CLI oppure operazione API `AssumeRole`	Utente IAM o sessione	Parametro API `Tags` o opzione CLI `--tags`	Parametro API `TransitiveTagKeys` o opzione CLI `--transitive-tag-keys`

Operation	Chi può assumere il ruolo	Metodo di passaggio dei tag	Metodo di impostazione dei tag transitivi
assume-role-with-saml CLI oppure operazione API AssumeRoleWithSAML	Tutti gli utenti autenticati che utilizzano un provider di identità SAML	Attributo SAML `PrincipalTag`	Attributo SAML `TransitiveTagKeys`
assume-role-with-web-identity CLI oppure operazione API AssumeRoleWithWebIdentity	Tutti gli utenti autenticati che utilizzano un provider OIDC	Token OIDC `PrincipalTag`	Token OIDC `TransitiveTagKeys`
get-federation-token CLI oppure operazione API GetFederationToken	Utente IAM o utente root	Parametro API `Tags` o opzione CLI `--tags`	Non supportata

Le operazioni che supportano i tag di sessione potrebbero non concludersi correttamente se si verifica una delle seguenti condizioni:

- Sono passati oltre 50 tag di sessione.
- Il testo in chiaro delle chiavi dei tag di sessione supera i 128 caratteri.
- Il testo in chiaro dei valori dei tag di sessione supera i 256 caratteri.
- La dimensione totale del testo in chiaro delle policy di sessione supera i 2048 caratteri.
- Le politiche di sessione e i tag di sessione si aggiungono alla dimensione del token di sessione. Se il token supera il limite di dimensione, l'operazione non riesce. Il messaggio di errore riporta la dimensione del token di sessione e la dimensione massima consentita, entrambe in byte. Monitora `SessionTokenUtilization` la risposta dell' AWS STS API per tracciare il tuo utilizzo.

Cose da sapere sui tag di sessione

Prima di utilizzare i tag di sessione, esaminare i seguenti dettagli sulle sessioni e sui tag.

- Quando utilizzi i tag di sessione, le policy di attendibilità per tutti i ruoli connessi al provider di identità (IdP) che passa i tag devono disporre dell'autorizzazione `sts:TagSession`. Per i ruoli che non dispongono di questa autorizzazione nella policy di attendibilità, l'operazione `AssumeRole` avrà esito negativo.
- Quando si richiede una sessione, è possibile specificare i tag principali come tag di sessione. I tag si applicano alle richieste effettuate utilizzando le credenziali della sessione.
- I tag di sessione sono coppie chiave-valore. Ad esempio, per aggiungere informazioni di contatto a una sessione, è possibile aggiungere la chiave del tag di sessione `email` e il valore del tag `johndoe@example.com`.
- I tag di sessione devono seguire le regole per la denominazione dei tag in IAM e AWS STS. In questo argomento sono incluse informazioni sulla distinzione tra maiuscole e minuscole e sui prefissi riservati validi per i tag di sessione.
- I nuovi tag di sessione sovrascrivono quelli relativi ai ruoli assunti o agli utenti federati con la stessa chiave di tag, indipendentemente dall'utilizzo di lettere minuscole o maiuscole.
- Non è possibile passare i tag di sessione utilizzando. Console di gestione AWS
- I tag di sessione sono validi solo per la sessione corrente.
- I tag di sessione supportano il concatenamento dei ruoli. Per impostazione predefinita, AWS STS non trasmette i tag alle sessioni di ruolo successive. Tuttavia, è possibile impostare i tag di sessione come transitivi. I tag transitivi persistono durante il concatenamento dei ruoli e sostituiscono i valori `ResourceTag` corrispondenti dopo la valutazione della policy di attendibilità del ruolo. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.
- È possibile utilizzare i tag di sessione per controllare l'accesso alle risorse o per controllare quali tag possono essere passati in una sessione successiva. Per ulteriori informazioni, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.
- È possibile visualizzare i tag del principale per la sessione, inclusi i tag di sessione, nei log AWS CloudTrail . Per ulteriori informazioni, consulta Visualizzazione dei tag di sessione in CloudTrail.
- È necessario passare un singolo valore per ogni tag di sessione. AWS STS non supporta tag di sessione con più valori.
- È possibile passare un massimo di 50 tag di sessione. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

- Le policy di sessione e i tag di sessione si aggiungono alla dimensione del token di sessione. Se il token supera il limite di dimensione, l' AWS API AWS CLI o restituisce un errore. Il messaggio di errore riporta la dimensione del token di sessione e la dimensione massima consentita, entrambe in byte. L'elemento di `SessionTokenUtilization` risposta mostra la percentuale del limite di dimensione utilizzata dal token. L'elemento di `PackedPolicySize` risposta è obsoleto. Usare invece `SessionTokenUtilization`.

Autorizzazioni necessarie per aggiungere tag di sessione

Oltre a quella sull'operazione che corrisponde all'operazione API, è necessario disporre nella policy dell'autorizzazione per le seguenti operazioni:

```
sts:TagSession
```

> **Important**
>
> Quando si utilizzano i tag di sessione, i criteri di attendibilità dei ruoli per tutti i ruoli connessi a un provider di identità (IdP) devono disporre dell'autorizzazione `sts:TagSession`. L'operazione `AssumeRole` avrà esito negativo per qualsiasi ruolo connesso a un provider di identità che passa tag di sessione senza questa autorizzazione. Se non si desidera aggiornare la policy di attendibilità del ruolo per ogni ruolo, è possibile utilizzare un'istanza IdP separata per passare i tag di sessione. Quindi, aggiungi l'autorizzazione `sts:TagSession` solo ai ruoli connessi all'IdP separato.

È possibile utilizzare l'operazione `sts:TagSession` con le seguenti chiavi di condizione.

- `aws:PrincipalTag`: confronta il tag collegato al principale che effettua la richiesta con il tag specificato nella policy. Ad esempio, è possibile consentire a un'entità di passare i tag di sessione solo se l'entità che effettua la richiesta dispone dei tag specificati.
- `aws:RequestTag`: confronta la coppia chiave-valore del tag passata nella richiesta con la coppia del tag specificata nella policy. Ad esempio, è possibile consentire all'entità di passare tag di sessione specificati, ma solo con i valori specificati.
- `aws:ResourceTag`: confronta la coppia chiave-valore tag specificata nella policy con la coppia chiave-valore collegata alla risorsa. Ad esempio, puoi consentire al principale di passare i tag di sessione solo se il ruolo che assume include i tag specificati.

- `aws:TagKeys`: confronta le chiavi tag in una richiesta con quelle specificate nella policy. Ad esempio, è possibile consentire all'entità di passare solo i tag di sessione con le chiavi dei tag specificate. Questa chiave di condizione limita l'insieme massimo di tag di sessione che possono essere passati.
- `sts:TransitiveTagKeys`: confronta le chiavi dei tag di sessione transitivi nella richiesta con quelle specificate nella policy. Ad esempio, è possibile scrivere una policy per consentire a un'entità di impostare solo tag specifici come transitivi. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Ad esempio, la seguente policy di attendibilità del ruolo consente all'utente `test-session-tags` di assumere il ruolo a cui è collegata la policy. Quando l'utente assume il ruolo, deve utilizzare l' AWS API AWS CLI o per passare i tre tag di sessione richiesti e l'ID esterno richiesto. Inoltre, l'utente può scegliere di impostare i tag `Department` e `Project` come transitivi.

Example Esempio di policy di attendibilità dei ruoli per i tag di sessione

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIamUserAssumeRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"AWS": "arn:aws:iam::123456789012:user/test-session-
tags"},
            "Condition": {
              "StringLike": {
                    "aws:RequestTag/Project": "*",
                    "aws:RequestTag/CostCenter": "*",
                    "aws:RequestTag/Department": "*"
                },
                "StringEquals": {"sts:ExternalId": "Example987"}
            }
        },
        {
            "Sid": "AllowPassSessionTagsAndTransitive",
            "Effect": "Allow",
```

```
            "Action": "sts:TagSession",
            "Principal": {"AWS": "arn:aws:iam::123456789012:user/test-session-
tags"},
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/Project": "*",
                    "aws:RequestTag/CostCenter": "*"
                },
                "StringEquals": {
                    "aws:RequestTag/Department": [
                        "Engineering",
                        "Marketing"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "sts:TransitiveTagKeys": [
                        "Project",
                        "Department"
                    ]
                }
            }
        }
    ]
}
```

Che cosa fa questa policy?

- L'istruzione `AllowIamUserAssumeRole` consente all'utente `test-session-tags` di assumere il ruolo a cui è collegata la policy. Quando tale utente assume il ruolo, deve passare i tag di sessione richiesti e l'ID esterno.
  - Il primo blocco condizionale di questa istruzione richiede all'utente di passare i tag di sessione `Project`, `CostCenter` e `Department`. I valori dei tag non sono significativi in questa istruzione, quindi abbiamo usato caratteri jolly (*) per i valori dei tag. Questo blocco assicura che l'utente passi almeno questi tre tag di sessione. In caso contrario, l'operazione non va a buon fine. L'utente può passare tag aggiuntivi.
  - Il secondo blocco condizionale richiede all'utente di passare un ID esterno con il valore `Example987`.
- L'istruzione `AllowPassSessionTagsAndTransitive` autorizza l'operazione `sts:TagSession`. Questa operazione deve essere autorizzata prima che l'utente possa passare

i tag di sessione. Se la policy include la prima istruzione senza la seconda, l'utente non può assumere il ruolo.

- Il primo blocco condizionale di questa istruzione consente all'utente di passare qualsiasi valore per i tag di sessione `CostCenter` e `Project`. A tale scopo, è necessario utilizzare i caratteri jolly (*) per il valore del tag nella policy, il che richiede l'utilizzo dell'operatore di condizione `StringLike`.
- Il secondo blocco condizionale consente all'utente di passare solo i valori `Marketing` o `Engineering` per il tag di sessione `Department`.
- Il terzo blocco condizionale elenca l'insieme massimo di tag che possono essere impostati come transitivi. L'utente può scegliere di impostare un sottoinsieme o nessun tag come transitivo. Ma non può impostare tag aggiuntivi come transitivi. È possibile richiedere che imposti almeno uno dei tag come transitivo aggiungendo un altro blocco condizionale che include `"Null": {"sts:TransitiveTagKeys":"false"}`.

Passaggio dei tag di sessione utilizzando AssumeRole

L'`AssumeRole`operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere alle AWS risorse. È possibile utilizzare le credenziali dell'utente o del ruolo IAM per chiamare `AssumeRole`. Per passare i tag di sessione assumendo un ruolo, usa l'`--tags` AWS CLI opzione o il parametro `Tags` AWS API.

Per impostare i tag come transitivi, usa l'`--transitive-tag-keys` AWS CLI opzione o il parametro `TransitiveTagKeys` AWS API. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Nell'esempio seguente viene illustrata una richiesta di esempio che utilizza `AssumeRole`. In questo esempio, quando si assume il ruolo `my-role-example`, si crea una sessione denominata `my-session`. Aggiungere le coppie chiave-valore dei tag di sessione `Project` = `Automation`, `CostCenter` = `12345` e `Department` = `Engineering`. È inoltre possibile impostare i tag `Project` e `Department` come transitivi specificando le loro chiavi. È necessario passare un singolo valore per ogni tag di sessione. AWS STS non supporta tag di sessione con più valori.

Example Esempio AssumeRole di richiesta CLI

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/my-role-example \
--role-session-name my-session \
```

```
--tags Key=Project,Value=Automation Key=CostCenter,Value=12345
 Key=Department,Value=Engineering \
--transitive-tag-keys Project Department \
--external-id Example987
```

Passaggio dei tag di sessione utilizzando AssumeRoleWithSAML

L'`AssumeRoleWithSAML`operazione si autentica con SAML-based la federazione. Questa operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere alle risorse. AWS Per ulteriori informazioni sull'utilizzo SAML-based della federazione per Console di gestione AWS l'accesso, vedereConsentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS. Per dettagli sul AWS CLI AWS nostro accesso alle API, consultaFederazione SAML 2.0. Per un tutorial sulla configurazione della federazione SAML per gli utenti di Active Directory, consulta AWS Federated Authentication with Active Directory Federation Services (ADFS) nel Security Blog. AWS

In qualità di amministratore, puoi consentire ai membri della tua directory aziendale di unirsi all'utilizzo dell'operazione. AWS AWS STS `AssumeRoleWithSAML` A tale scopo, è necessario completare le seguenti attività:

1. Configurazione della rete come un provider SAML per AWS
2. Creazione di un provider SAML in IAM
3. Creare un ruolo per una federazione SAML 2.0 (console)
4. Termine della configurazione del provider di identità SAML e creazione di asserzioni per la risposta di autenticazione SAML

AWS include provider di identità con esperienza end-to-end certificata per i tag di sessione con le relative soluzioni di identità. Per informazioni su come utilizzare questi provider di identità per configurare i tag di sessione, consultare Integra fornitori di soluzioni SAML di terze parti con AWS.

Per passare gli attributi SAML come tag di sessione, includere l'elemento `Attribute` con l'attributo `Name` impostato a `https://aws.amazon.com/SAML/Attributes/PrincipalTag:`*`{TagKey}`*. Utilizzare l'elemento `AttributeValue` per specificare il valore del tag. Includere un elemento `Attribute` separato per ogni tag di sessione.

Ad esempio, si supponga di voler passare i seguenti attributi di identità come tag di sessione:

- `Project:Automation`

- `CostCenter:12345`
- `Department:Engineering`

Per passare questi attributi, includere i seguenti elementi nell'asserzione SAML.

Example Esempio di frammento di un'asserzione SAML

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:Project">
  <AttributeValue>Automation</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:CostCenter">
  <AttributeValue>12345</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:Department">
  <AttributeValue>Engineering</AttributeValue>
</Attribute>
```

Per impostare i tag sopra elencati come transitivi, include un altro elemento `Attribute` con l'attributo Name impostato su `https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys`. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Per impostare i tag `Project` e `Department` come transitivi, utilizzare il seguente attributo multi valore.

Example Esempio di frammento di un'asserzione SAML

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys">
  <AttributeValue>Project</AttributeValue>
  <AttributeValue>Department</AttributeValue>
</Attribute>
```

Passaggio dei tag di sessione utilizzando AssumeRoleWithWebIdentity

Utilizza la federazione compatibile con OpenID Connect (OIDC) per autenticare l'operazione `AssumeRoleWithWebIdentity`. Questa operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere alle AWS risorse. Per ulteriori informazioni sull'utilizzo della federazione delle identità Web per Console di gestione AWS l'accesso, vedereFederazione OIDC.

Per passare i tag di sessione da OpenID Connect (OIDC), è necessario includere i tag di sessione nel token Web JSON (JWT) quando si invia la richiesta `AssumeRoleWithWebIdentity`. Per ulteriori informazioni sui token e le registrazioni OIDC, consultare Utilizzo di token con pool di utenti nella Guida per gli sviluppatori Amazon Cognito .

AWS supporta due formati di attestazione per l'inclusione dei tag di sessione nel JWT:

- Formato di attestazione nidificato
- Formato di nidificato compresso

Formato di attestazione nidificato

Il formato di attestazione nidificato utilizza una struttura all'interno del namespace `https://aws.amazon.com/tags` nel JWT. In questo formato:

- I tag principali sono rappresentati come un oggetto nidificato sotto la chiave `principal_tags`.
- Ogni tag principale è il valore di una singola stringa.
- Le chiavi dei tag transitivi sono rappresentate in un array sotto la chiave `transitive_tag_keys`.
- Sia `principal_tags` che `transitive_tag_keys` sono nidificati nel namespace `https://aws.amazon.com/tags`.

Nell'esempio seguente viene mostrato un JWT decodificato utilizzando il formato di oggetti nidificati:

Example Esempio di token Web JSON decodificato utilizzando il formato di attestazione nidificato

```
{
    "sub": "johndoe",
    "aud": "ac_oic_client",
    "jti": "ZYUCeRMQVtqHypVPWAN3VB",
    "iss": "https://xyz.com",
    "iat": 1566583294,
    "exp": 1566583354,
    "auth_time": 1566583292,
    "https://aws.amazon.com/tags": {
        "principal_tags": {
            "Project": ["Automation"],
            "CostCenter": ["987654"],
            "Department": ["Engineering"]
        },
        "transitive_tag_keys": [
```

```
            "Project",
            "CostCenter"
        ]
    }
}
```

Formato di nidificato compresso

Il formato di attestazione compresso è compatibile con i provider di identità che non supportano oggetti nidificati nelle attestazioni JWT, come Microsoft Entra ID. In questo formato:

- I tag principali sono rappresentati come attestazioni separate con il prefisso `https://aws.amazon.com/tags/principal_tags/`.
- Ogni tag principale è il valore di una singola stringa.
- Le chiavi dei tag transitivi sono rappresentate in una singola attestazione come array di stringhe con il prefisso `https://aws.amazon.com/tags/transitive_tag_keys`.

Ora, vediamo come vengono rappresentate le stesse informazioni utilizzando il formato di attestazione compresso:

Example Esempio di token Web JSON decodificato utilizzando il formato di attestazione compresso

```
{
    "sub": "johndoe",
    "aud": "ac_oic_client",
    "jti": "ZYUCeRMQVtqHypVPWAN3VB",
    "iss": "https://xyz.com",
    "iat": 1566583294,
    "exp": 1566583354,
    "auth_time": 1566583292,
    "https://aws.amazon.com/tags/principal_tags/Project": "Automation",
    "https://aws.amazon.com/tags/principal_tags/CostCenter": "987654",
    "https://aws.amazon.com/tags/principal_tags/Department": "Engineering",
    "https://aws.amazon.com/tags/transitive_tag_keys": [
        "Project",
        "CostCenter"
    ]
}
```

Entrambi gli esempi JWT decodificati mostrano una chiamata a `AssumeRoleWithWebIdentity` con i tag di sessione `Project`, `CostCenter` e `Department`. Entrambi i token impostano i tag

`Project` e `CostCenter` come transitivi. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Il formato di attestazione compresso ottiene lo stesso risultato del formato di attestazione nidificato, ma utilizza una struttura compressa per i tag. Consente di includere tag di sessione in ambienti in cui gli oggetti JSON nidificati non sono supportati nelle attestazioni JWT. Quando utilizzi uno dei due formati, assicurati che il tuo provider di identità sia configurato per emettere token con le strutture di attestazione appropriate. AWS supporta entrambi i formati di attestazione, quindi puoi scegliere quello più adatto ai requisiti specifici del tuo provider di identità.

Passaggio dei tag di sessione utilizzando GetFederationToken

La `GetFederationToken` consente di federare l'utente. Questa operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere alle AWS risorse. Per aggiungere tag alla sessione utente federata, usa l'`--tags` AWS CLI opzione o il parametro `Tags` AWS API. Non è possibile impostare i tag di sessione come transitivi quando si utilizza `GetFederationToken`, perché non è possibile utilizzare le credenziali provvisorie per assumere un ruolo. In questo caso non è possibile utilizzare il concatenamento dei ruoli.

Di seguito è mostrata una risposta di esempio che utilizza `GetFederationToken`. In questo esempio, quando si richiede il token, si crea una sessione denominata `my-fed-user`. Aggiungere le coppie chiave-valore dei tag di sessione `Project` = `Automation` e `Department` = `Engineering`.

Example Esempio di richiesta GetFederationToken CLI

```
aws sts get-federation-token \
--name my-fed-user \
--tags key=Project,value=Automation key=Department,value=Engineering
```

Quando si utilizzano le credenziali temporanee restituite dall'operazione `GetFederationToken`, i tag del principale della sessione includono i tag dell'utente e i tag di sessione passati.

Concatenamento dei ruoli con i tag di sessione

È possibile assumere un ruolo e quindi utilizzare le credenziali temporanee per assumere un altro ruolo. È possibile continuare da una sessione all'altra. Questa operazione è chiamata concatenamento del ruolo. Quando si passano i tag di sessione mentre si assume un ruolo, è possibile impostare le chiavi come transitive. Ciò garantisce che tali tag di sessione siano passati alle sessioni successive in un concatenamento di ruoli. Non è possibile impostare tag di ruolo come transitivi. Per passare questi tag alle sessioni successive, specificarli come tag di sessione.

> **Note**
>
> I tag transitivi persistono durante il concatenamento dei ruoli e sostituiscono i valori `ResourceTag` corrispondenti dopo la valutazione della policy di attendibilità del ruolo.

L'esempio seguente mostra come AWS STS passare i tag di sessione, i tag transitivi e i tag di ruolo nelle sessioni successive di una catena di ruoli.

In questo esempio di scenario di concatenamento dei ruoli, si utilizza una chiave di accesso utente IAM AWS CLI per assumere un ruolo denominato `Role1` È quindi possibile utilizzare le credenziali di sessione risultanti per assumere un secondo ruolo denominato `Role2`. È quindi possibile utilizzare le credenziali della seconda sessione per assumere un terzo ruolo denominato `Role3`. Queste richieste sono eseguite come tre operazioni separate. Ogni ruolo è già taggato in IAM. E durante ogni richiesta, è possibile passare ulteriori tag di sessione.
Concatenamento di ruoli.

Quando si concatenano i ruoli, è possibile assicurarsi che i tag di una sessione precedente persistano nelle sessioni successive. Per fare ciò utilizzando il comando `assume-role` della CLI, è necessario passare il tag come tag di sessione e impostare il tag come transitivo. Si passa il tag `Star = 1` come tag di sessione. Il comando collega anche il tag `Heart = 1` al ruolo e lo applica come tag del principale quando si utilizza la sessione. Tuttavia, si desidera anche che il tag `Heart = 1` sia passato automaticamente alla seconda o terza sessione. Per farlo, è necessario includerlo manualmente come tag di sessione. I tag principali di sessione risultanti includono entrambi questi tag e li impostano come transitivi.
Assumere il primo ruolo in una catena di ruoli.

Si esegue questa richiesta utilizzando il seguente AWS CLI comando:

Example Esempio AssumeRole di richiesta CLI

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/Role1 \
--role-session-name Session1 \
--tags Key=Star,Value=1 Key=Heart,Value=1 \
--transitive-tag-keys Star Heart
```

È quindi possibile utilizzare le credenziali di tale sessione per assumere il `Role2`. Il comando collega il tag `Sun = 2` al secondo ruolo e lo applica come tag del principale quando utilizzi la sessione. I `Star`

tag `Heart` and ereditano i tag transitivi della sessione nella prima sessione. I tag del principale della seconda sessione risultanti sono `Heart` = `1`, `Star` = `1` e `Sun` = `2`. `Heart` e `Star` continueranno a essere transitivi. Il tag `Sun` collegato a `Role2` non è contrassegnato come transitivo perché non è un tag di sessione. Le sessioni future non ereditano questo tag.
Assumere il secondo ruolo in una catena di ruoli.

Eseguite questa seconda richiesta utilizzando il seguente AWS CLI comando:

Example Esempio AssumeRole di richiesta CLI

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/Role2 \
--role-session-name Session2
```

È quindi possibile utilizzare le credenziali della seconda sessione per assumere il `Role3`. I tag dell'entità per la terza sessione derivano da tutti i nuovi tag di sessione, i tag di sessione transitivi ereditati e i tag di ruolo. I tag `Heart` = `1` e `Star` = `1` nella seconda sessione sono stati ereditati dai tag di sessione transitivi nella prima sessione. Se provi a passare il tag di sessione `Sun` = `2`, l'operazione avrà esito negativo. Il tag di sessione `Star` = `1` ereditato sostituisce il tag `Star` = `3` del ruolo. Nella concatenazione dei ruoli, il valore di un tag transitivo sovrascrive il ruolo che corrisponde al valore `ResourceTag` dopo della valutazione della policy di attendibilità del ruolo. In questo esempio, se `Role3` utilizza `Star` come `ResourceTag` nella policy di attendibilità ruolo e imposta il valore `ResourceTag` sul valore del tag transitivo dalla sessione del ruolo chiamante. Il `Lightning` tag role si applica anche alla terza sessione e non è impostato come transitivo.
Assumendo il terzo ruolo in una catena di ruoli.

Si esegue la terza richiesta utilizzando il seguente AWS CLI comando:

Example Esempio AssumeRole di richiesta CLI

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/Role3 \
--role-session-name Session3
```

Utilizzo dei tag di sessione per ABAC

Attribute-based il controllo degli accessi (ABAC) utilizza una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi dei tag.

Se la tua azienda utilizza un OIDC o un provider di SAML-based identità (IdP) per gestire le identità degli utenti, puoi configurare la tua asserzione a cui passare i tag di sessione. AWS Ad esempio, con le identità utente aziendali, quando i dipendenti si federano, AWS applicano i loro attributi al committente AWS risultante. È quindi possibile utilizzare ABAC per consentire o negare le autorizzazioni sulla base di tali attributi. Per informazioni dettagliate, vedi Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.

Per ulteriori informazioni sull'utilizzo di IAM Identity Center con ABAC, consulta Attributi per il controllo degli accessi nella Guida per l'utente di Centro identità AWS IAM .

Visualizzazione dei tag di sessione in CloudTrail

È possibile utilizzare AWS CloudTrail per visualizzare le richieste utilizzate per assumere ruoli o federare utenti. Il file di registro di CloudTrail include informazioni sui tag dell'entità per la sessione dell'utente che ha assunto il ruolo o dell'utente federato. Per ulteriori informazioni, consulta Registrazione di IAM e AWS STS Chiamate API con AWS CloudTrail.

Ad esempio, supponiamo di effettuare una AWS STS `AssumeRoleWithSAML` richiesta, di passare i tag di sessione e di impostarli come transitivi. Nel file di registro di CloudTrail è possibile trovare le seguenti informazioni.

Example Registro di esempio AssumeRoleWithSAML CloudTrail

```
    "requestParameters": {
        "sAMLAssertionID": "_c0046cEXAMPLEb9d4b8eEXAMPLE2619aEXAMPLE",
        "roleSessionName": "MyRoleSessionName",
        "principalTags": {
            "CostCenter": "987654",
            "Project": "Unicorn"
        },
        "transitiveTagKeys": [
            "CostCenter",
            "Project"
        ],
        "durationSeconds": 3600,
        "roleArn": "arn:aws:iam::123456789012:role/SAMLTestRoleShibboleth",
        "principalArn": "arn:aws:iam::123456789012:saml-provider/Shibboleth"
    },
```

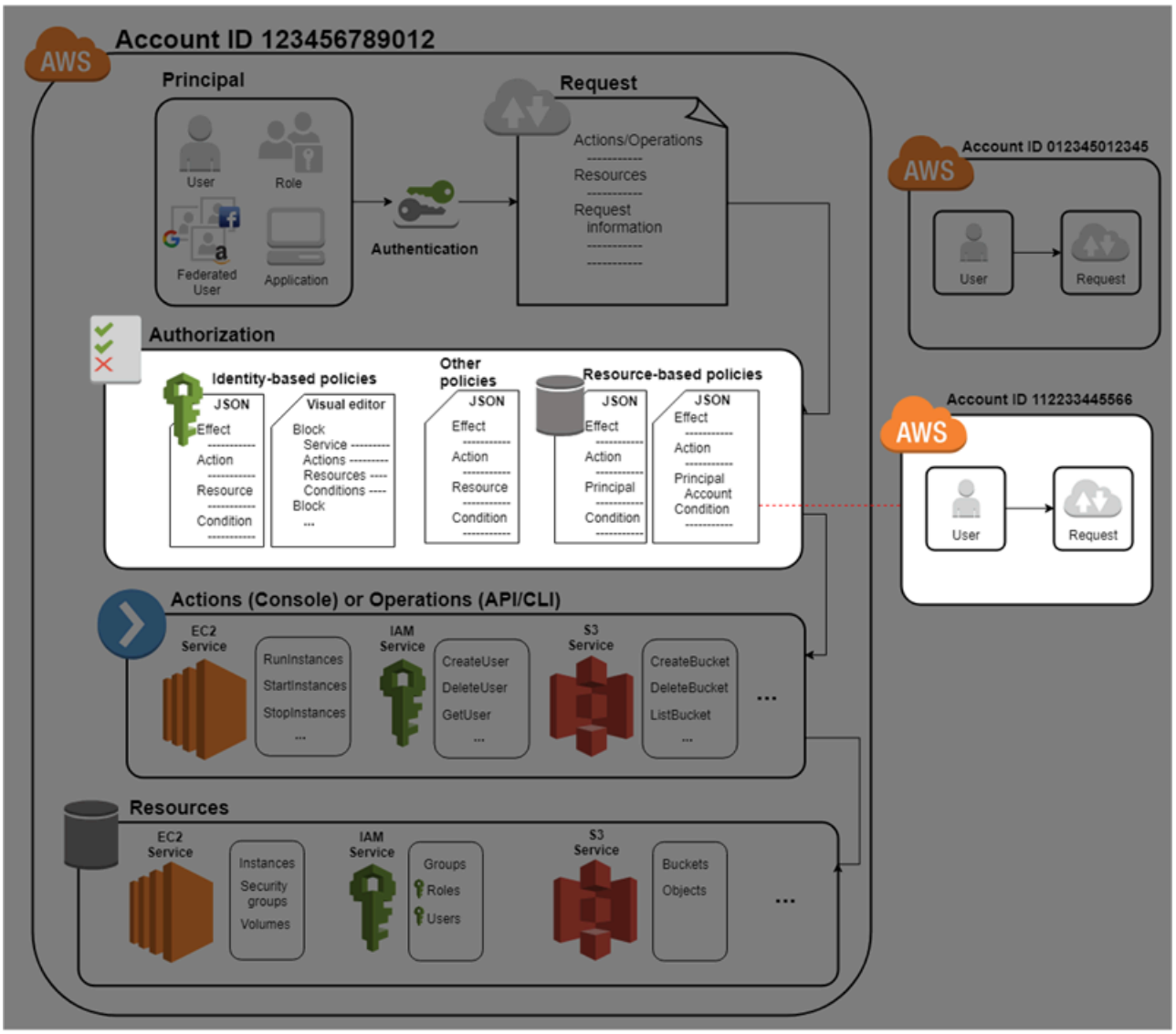
È possibile visualizzare i seguenti CloudTrail registri di esempio per visualizzare gli eventi che utilizzano i tag di sessione.

- Esempio AWS STS evento API di concatenamento dei ruoli nel file di registro CloudTrail
- Esempio SAML AWS STS Evento API nel CloudTrail file di registro
- Esempio: OIDC AWS STS Evento API nel file di registro CloudTrail

Gestione degli accessi per AWS risorse

AWS Identity and Access Management (IAM) è un servizio web che ti aiuta a controllare in modo sicuro l'accesso alle AWS risorse. Quando un principale effettua una richiesta di accesso AWS, il codice di AWS applicazione verifica se il principale è autenticato (ha effettuato l'accesso) e autorizzato (dispone delle autorizzazioni). Puoi gestire l'accesso AWS creando policy e associandole alle identità o alle risorse IAM. AWS Le policy sono documenti JSON in AWS cui, quando sono associate a un'identità o a una risorsa, ne definiscono le autorizzazioni. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Politiche e autorizzazioni in AWS Identity and Access Management.

Per ulteriori informazioni sul resto del processo di autenticazione e autorizzazione, consultare Funzionamento di IAM.



Durante l'autorizzazione, il codice di AWS applicazione utilizza i valori del contesto della richiesta per verificare le politiche corrispondenti e determinare se consentire o rifiutare la richiesta.

AWS controlla ogni policy che si applica al contesto della richiesta. Se una singola politica rifiuta la richiesta, AWS nega l'intera richiesta e interrompe la valutazione delle politiche. Questa azione si chiama rifiuto esplicito. Poiché le richieste vengono rifiutate per impostazione predefinita, IAM autorizza la richiesta solo se ogni parte di essa è autorizzata dalle policy applicabili. La logica di valutazione per una richiesta all'interno di un singolo account segue queste regole:

- Per impostazione predefinita, tutte le richieste vengono negate implicitamente (in alternativa, per impostazione predefinita, l' Utente root dell'account AWS ha accesso completo).
- Un'autorizzazione esplicita in una policy basata su identità o basata su risorse sostituisce questa impostazione predefinita.

- Se è presente un limite di autorizzazioni, un AWS Organizations SCP o una politica di sessione, potrebbe sovrascrivere l'autorizzazione con una negazione implicita.
- Un rifiuto esplicito in una policy sostituisce qualsiasi permesso.

Dopo che la richiesta è stata autenticata e autorizzata, la approva. AWS Se hai bisogno di effettuare una richiesta in un altro account, una policy nell'altro account deve consentire l'accesso alla risorsa. Inoltre, l'entità IAM utilizzata per effettuare la richiesta deve avere una policy basata su identità che consente la richiesta.

Puoi anche utilizzare Delegazione temporanea IAM per concedere ad Amazon and AWS Partners un accesso limitato e limitato nel tempo per configurare AWS i servizi nel tuo account.

Accesso alle risorse di gestione

Per ulteriori informazioni sulle autorizzazioni e sulla creazione di policy, consultare le seguenti risorse:

Le seguenti voci del AWS Security Blog descrivono i modi più comuni per scrivere policy per l'accesso a bucket e oggetti Amazon S3.

- Scrittura di policy IAM: come concedere l'accesso a un bucket Amazon S3
- Scrittura di policy IAM: concedere l'accesso alle User-Specific cartelle in un bucket Amazon S3
- Policy IAM, policy di bucket e ACL Oh My! (Controllo dell'accesso alle risorse S3)
- Introduzione alle autorizzazioni RDS Resource-Level
- Demistificazione delle autorizzazioni EC2 Resource-Level

Politiche e autorizzazioni in AWS Identity and Access Management

Gestisci l'accesso AWS creando policy e associandole alle identità (utenti, gruppi di utenti o ruoli) o AWS alle risorse IAM. Una policy è un oggetto AWS che, quando associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un responsabile IAM (utente o ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle policy sono archiviate AWS come documenti JSON. AWS supporta nove tipi di policy: policy basate sull'identità, policy basate sulle risorse, policy sugli endpoint VPC, limiti delle autorizzazioni, policy di controllo dei AWS Organizations servizi (SCP), policy di controllo AWS Organizations delle risorse (RCP), liste di controllo degli accessi (ACL), condivisioni di Resource Access Manager (RAM) e policy di sessione.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, indipendentemente dal metodo utilizzato per eseguirla. Ad esempio, se una policy consente l'GetUserazione, un utente con tale policy può ottenere informazioni sull'utente da, the o dall'API. Console di gestione AWS AWS CLI AWS Nella creazione di un utente IAM, è possibile scegliere di consentire l'accesso programmatico o alla console. Se è consentito l'accesso alla console, l'utente IAM può accedere alla console con le proprie credenziali di accesso. Se è consentito l'accesso programmatico, l'utente può utilizzare le chiavi di accesso per utilizzare la CLI o l'API.

Tipi di policy

I tipi di policy elencati di seguito in ordine da quello utilizzato più frequentemente a quello meno frequentemente sono disponibili per l'uso in AWS. Per ulteriori informazioni, consulta le sezioni seguenti per ogni tipo di policy.

- Identity-basedpolicy : associa policy gestite e in linea alle identità IAM (utenti, gruppi a cui appartengono gli utenti o ruoli). Identity-based le policy concedono le autorizzazioni a un'identità.
- Resource-basedpolitiche : allega politiche in linea alle risorse. Gli esempi più comuni di policy basate sulle risorse sono le politiche bucket di Amazon S3 e le politiche di fiducia dei ruoli IAM. Resource-based le policy concedono le autorizzazioni al principale specificato nella policy. Le entità principali possono essere nello stesso account della risorsa o in altri account.
- Policy degli endpoint VPC : collegatevi a un endpoint VPC per controllare quali principali possono utilizzare le policy degli endpoint VPC e a quali risorse è possibile accedere tramite di esse. Le policy degli endpoint VPC fungono da limite di accesso aggiuntivo con ambito di accesso al traffico che attraversa l'endpoint.
- Limiti delle autorizzazioni: utilizza una policy gestita come limite delle autorizzazioni per un'entità IAM (utente o ruolo). Questa policy definisce il numero massimo di autorizzazioni che la policy basata su identità può concedere a un'entità, ma non concede autorizzazioni. I limiti delle autorizzazioni non definiscono il numero massimo di autorizzazioni che una policy basata su risorse può concedere a un'entità.
- AWS Organizations SCP : utilizzate una policy di controllo del AWS Organizations servizio (SCP) per definire le autorizzazioni massime per gli utenti IAM e i ruoli IAM all'interno degli account della vostra organizzazione o unità organizzativa (OU). Le SCP limitano le autorizzazioni che le policy basate su identità o le policy basate su risorse concedono a utenti IAM o ruoli IAM all'interno dell'account. Le SCP non concedono autorizzazioni.
- AWS Organizations RCP : utilizzate una policy di controllo AWS Organizations delle risorse (RCP) per definire le autorizzazioni massime per le risorse all'interno degli account della vostra

organizzazione o unità organizzativa (OU). Le RCP limitano le autorizzazioni che le policy basate su identità e le policy basate sulle risorse possono concedere alle risorse degli account all'interno dell'organizzazione. Le RCP non concedono autorizzazioni.

- Liste di controllo accessi (ACL): utilizza le ACL per controllare quali entità principali in altri account possono accedere alla risorsa a cui è collegata l'ACL. Le ACL sono simili alle policy basate su risorse, anche se sono l'unico tipo di policy che non utilizza la struttura del documento di policy JSON. Le ACL sono policy di autorizzazione tra più account che concedono le autorizzazioni all'identità principale specificata. Le ACL non possono concedere autorizzazioni a entità nello stesso account.
- AWS Condivisioni di risorse RAM : utilizza AWS Resource Access Manager per condividere le risorse tra Account AWS le unità organizzative o un'intera organizzazione senza scrivere politiche basate sulle risorse individuali per ogni risorsa condivisa.
- Criteri di sessione : approva politiche di sessione avanzate quando usi l' AWS API AWS CLI o per assumere un ruolo o un utente federato. Le policy di sessione limitano le autorizzazioni che le policy basate su identità dell'utente o del ruolo concedono alla sessione. Le policy di sessione limitano le autorizzazioni per una sessione creata, ma non possono concedere autorizzazioni. Per ulteriori informazioni, consulta la sezione relativa alle policy di sessione.

Identity-based politiche

Identity-based le policy sono documenti di policy sulle autorizzazioni JSON che controllano quali azioni può eseguire un'identità (utenti, gruppi di utenti e ruoli), su quali risorse e in quali condizioni. Identity-based le politiche possono essere ulteriormente classificate:

- Policy gestite: politiche autonome basate sull'identità che puoi associare a più utenti, gruppi e ruoli nel tuo. Account AWS Sono disponibili due tipi di policy gestite.
  - AWS politiche gestite: politiche gestite create e gestite da. AWS
  - Policy gestite dal cliente: le policy gestite che sono create e gestite nel tuo Account AWS. Le politiche gestite dal cliente forniscono un controllo più preciso sulle politiche rispetto alle politiche AWS gestite.
- Policy inline: le policy che vengono aggiunte direttamente a un singolo utente, gruppo o ruolo. Le policy in linea sono utili per mantenere una stretta relazione uno a uno tra una policy e un'identità. Vengono eliminate quando elimini l'identità.

Per informazioni su come scegliere tra una policy gestita o una policy in linea, consulta Scegliere tra policy gestite e policy in linea.

Resource-based politiche

Resource-based le policy sono documenti di policy JSON che alleghi a una risorsa come un bucket Amazon S3. Queste politiche concedono al principale specificato il permesso di eseguire azioni specifiche su quella risorsa e definiscono a quali condizioni ciò si applica. Resource-basedle politiche sono politiche in linea. Non esistono policy basate su risorse gestite.

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. L'aggiunta di un'entità principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono separati Account AWS, è inoltre necessario utilizzare una politica basata sull'identità per concedere al principale l'accesso alla risorsa. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate sull'identità. Per istruzioni dettagliate sulla concessione dell'accesso tra servizi, consulta Tutorial IAM: Delega l'accesso attraverso AWS account che utilizzano ruoli IAM.

Il servizio IAM supporta solo un tipo di policy basata su risorse detta policy di attendibilità del ruolo, collegata a un ruolo IAM. Un ruolo IAM è sia un'identità che una risorsa che supporta le policy basate sulle risorse. Per questo motivo, è necessario collegare sia una policy di attendibilità sia una policy basata su identità a un ruolo IAM. Le policy di attendibilità definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono assumere il ruolo. Per capire in che modo i ruoli IAM si differenziano da altre policy basate su risorse, consulta Accesso alle risorse multi-account in IAM.

Per scoprire quali altri servizi supportano le policy basate su risorse, consulta AWS servizi supportati da IAM. Per ulteriori informazioni sulle policy basate su risorse, consulta la pagina Identity-based politiche e politiche basate sulle risorse. Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Policy di endpoint VPC

Una policy sugli endpoint VPC è una policy basata sulle risorse che si allega a un endpoint VPC per controllare quali principali possono utilizzare l'endpoint e a quali risorse è possibile accedere tramite esso. Le policy sugli endpoint VPC non sovrascrivono o sostituiscono le politiche basate sull'identità o le politiche basate sulle risorse collegate al servizio di destinazione: fungono da limite di accesso aggiuntivo con ambito di accesso al traffico che attraversa l'endpoint.

Le policy sugli endpoint VPC vengono utilizzate per garantire che solo le identità affidabili dell'organizzazione possano accedere alle risorse affidabili tramite gli endpoint VPC. AWS Organizations Se non alleghi una policy personalizzata per gli endpoint, AWS allega una policy predefinita che consente l'accesso completo. Per ulteriori informazioni sulle politiche degli endpoint, consulta Controllare l'accesso agli endpoint VPC utilizzando le politiche degli endpoint nella Guida. AWS PrivateLink

Limiti delle autorizzazioni IAM

Un limite delle autorizzazioni è una funzione avanzata in cui si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM. Quando si imposta un limite delle autorizzazioni per un'entità, l'entità può eseguire solo le operazioni consentite dalle sue policy basate su identità e dai suoi limiti delle autorizzazioni.

Se si specifica una sessione del ruolo o un utente nell'elemento principale di una policy basata sulle risorse, non è richiesta un'autorizzazione esplicita nel limite delle autorizzazioni. Tuttavia, se si specifica un ARN del ruolo nell'elemento principale di una policy basata sulle risorse, è richiesta un'autorizzazione esplicita nel limite delle autorizzazioni. In entrambi i casi, è effettiva una negazione esplicita nel limite dell'autorizzazione. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consultare Limiti delle autorizzazioni per le entità IAM.

AWS Organizations politiche di controllo dei servizi (SCP)

Se abiliti tutte le funzionalità in un'organizzazione, è possibile applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. Le SCP sono policy JSON che specificano il numero massimo di autorizzazioni per gli utenti IAM e i ruoli IAM all'interno di account di un'organizzazione o un'unità organizzativa (UO). L'SCP limita le autorizzazioni per i titolari negli account dei membri, inclusi ciascuno di essi. Utente root dell'account AWS Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione nelle altre policy.

Per ulteriori informazioni sugli SCP AWS Organizations e sugli SCP, consulta le politiche di controllo dei servizi (SCP) nella Guida per l'utente. AWS Organizations

AWS Organizations politiche di controllo delle risorse (RCP)

Se si abilitano tutte le funzionalità in un'organizzazione, è possibile utilizzare le policy di controllo delle risorse (RCP) per applicare centralmente i controlli degli accessi alle risorse su più Account AWS. Le policy di controllo delle risorse (RCP) sono policy JSON che possono essere utilizzate

per impostare il numero massimo di autorizzazioni disponibili per le risorse nei tuoi account senza aggiornare le policy IAM collegate a ciascuna risorsa di tua proprietà. L'RCP limita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Un rifiuto esplicito in qualsiasi RCP applicabile ha la precedenza su un'autorizzazione in altre policy che potrebbero essere associate a singole identità o risorse.

Per ulteriori informazioni sugli RCP AWS Organizations e su un elenco di quelli Servizi AWS che supportano gli RCP, consulta le politiche di controllo delle risorse (RCP) nella Guida per l'utente. AWS Organizations

Liste di controllo degli accessi (ACL)

Le liste di controllo accessi (ACL) sono policy di servizio che consentono di controllare quali principali in un altro account possono accedere a una risorsa. Le ACL non possono essere utilizzate per controllare l'accesso per un'entità principale all'interno dello stesso account. Le ACL sono simili alle policy basate su risorse, anche se sono l'unico tipo di policy che non utilizza il formato del documento di policy JSON. Amazon S3 e Amazon VPC sono AWS WAF esempi di servizi che supportano gli ACL. Per maggiori informazioni sulle ACL, consulta la pagina Access Control List (ACL) overview nella Guida per gli sviluppatori di Amazon Simple Storage Service.

AWS Resource Access Manager (AWS Condivisioni di risorse RAM

AWS Resource Access Manager (AWS RAM) consente di condividere le risorse create all'interno di un'altra Account AWS unità organizzativa o Account AWS con un'intera organizzazione AWS Organizations, senza dover scrivere politiche individuali basate sulle risorse per ogni risorsa condivisa. Sebbene sia possibile concedere l'accesso tra più account allegando una policy basata sulle risorse direttamente a una risorsa, la AWS RAM offre un'alternativa gestita e centralizzata che elimina la necessità di enumerare gli ID degli account nelle politiche o di mantenere documenti di policy identici tra gli account.

Con la RAM, gli account che consumano vedono le risorse condivise in modo nativo nelle loro console di servizio, i proprietari delle risorse mantengono la piena proprietà e visibilità su chi ha accesso e le autorizzazioni gestite definiscono le azioni massime che i consumatori possono eseguire, il tutto regolato da un'unica condivisione delle risorse anziché da politiche per risorsa. I casi d'uso più comuni includono la condivisione di sottoreti VPC, allegati AWS Transit Gateway, regole Route 53 Resolver e configurazioni di License Manager tra gli account. Per ulteriori informazioni, vedi Cos'è Resource Access Manager? AWSnella AWS RAM User Guide.

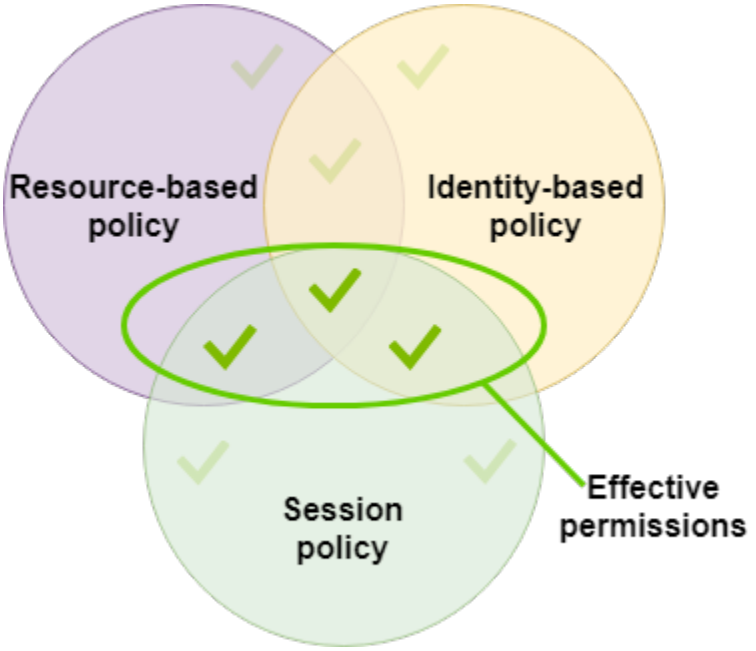
Policy di sessione

Le policy di sessione sono politiche avanzate che vengono passate come parametro quando si crea a livello di programmazione una sessione temporanea per un ruolo o un utente principale AWS STS federato. Le autorizzazioni per una sessione sono l'intersezione delle policy basate su identità per l'entità IAM (utente o ruolo) utilizzate per creare la sessione e delle policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione.

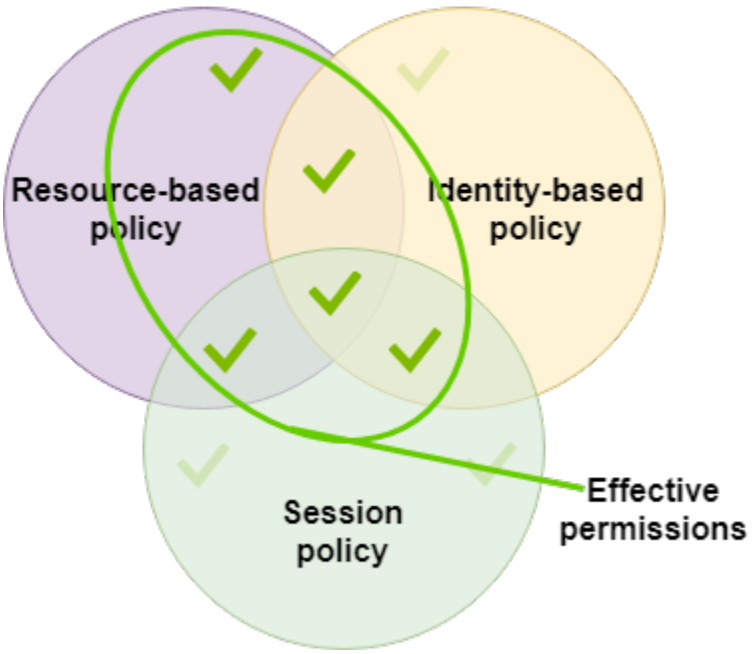
Puoi creare una sessione del ruolo e passare policy di sessione a livello di programmazione utilizzando le operazioni API `AssumeRole`, `AssumeRoleWithSAML` o `AssumeRoleWithWebIdentity`. Puoi passare un singolo documento della policy di sessione inline JSON utilizzando il parametro `Policy`. Puoi utilizzare il parametro `PolicyArns` per specificare fino a 10 policy di sessione gestite. Per ulteriori informazioni sulla creazione di una sessione del ruolo, consulta Autorizzazioni per le credenziali di sicurezza temporanee.

Quando si crea una sessione principale utente AWS STS federata, si utilizzano le chiavi di accesso dell'utente IAM per chiamare a livello di codice l'operazione API. `GetFederationToken` È inoltre necessario passare policy di sessione. Le autorizzazioni della sessione risultanti sono l'intersezione tra la policy basata su identità e la policy di sessione. Per ulteriori informazioni sulla creazione di una sessione per l'utente federato, consulta Richiesta di credenziali tramite un gestore di identità personalizzato.

Una policy basata sulle risorse è in grado di specificare l'ARN dell'utente o del ruolo come un principale. In questo caso, le autorizzazioni della policy basata sulle risorse vengono aggiunte alla policy basata su identità dell'utente o del ruolo prima che la sessione venga creata. La policy di sessione limita le autorizzazioni totali concesse dalla policy basata sulle risorse e dalla policy basata su identità. Le autorizzazioni della sessione risultante sono l'intersezione delle policy di sessione e della policy basata sulle risorse più l'intersezione delle policy di sessione e delle policy basate su identità.

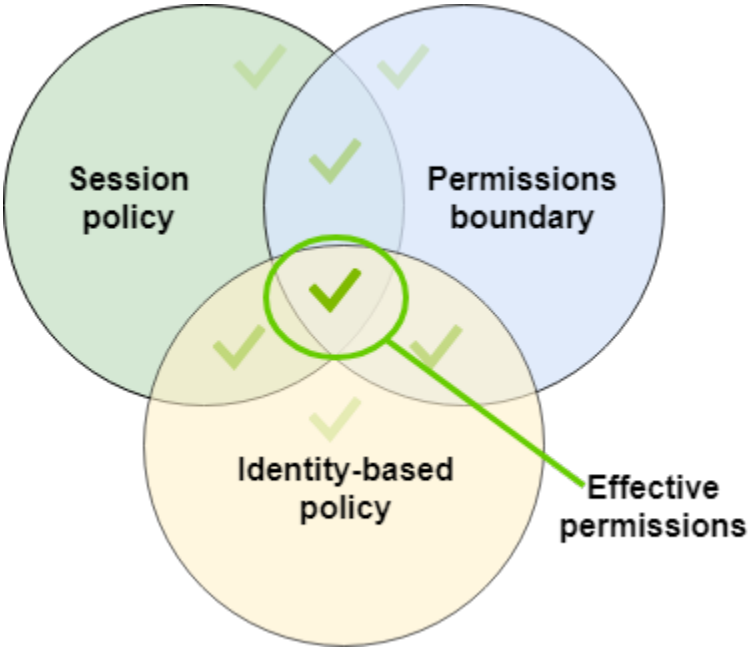


Una policy basata sulle risorse è in grado di specificare l'ARN della sessione come un principale. In questo caso, le autorizzazioni della policy basata sulle risorse vengono aggiunte dopo che la sessione viene creata. Le autorizzazioni della policy basate sulle risorse non sono limitate dalla policy di sessione. La sessione risultante dispone di tutte le autorizzazioni della policy basata sulle risorse più l'intersezione della policy basata su identità e della policy di sessione.



Un limite delle autorizzazioni è in grado di impostare il numero massimo di autorizzazioni per un utente o un ruolo che viene utilizzato per creare una sessione. In tal caso, le autorizzazioni della

sessione risultante sono l'intersezione della policy di sessione, il limite delle autorizzazioni e la policy basata su identità. Tuttavia, un limite delle autorizzazioni non limita le autorizzazioni concesse da una policy basata sulle risorse che specifica l'ARN della sessione risultante.



Policy e utente root

Utente root dell'account AWS È influenzato da alcuni tipi di policy ma non da altri. Non è possibile collegare policy basate su identità all'utente root e non è possibile impostare il limite delle autorizzazioni per questo utente. Tuttavia, è possibile specificare l'utente root come principale in una policy basata su risorse o un'ACL. Un utente root è ancora membro di un account. Se l'account è membro di un'organizzazione in AWS Organizations, l'utente root è interessato dagli SCP e dagli RCP relativi all'account.

Panoramica delle policy JSON

La maggior parte delle policy sono archiviate AWS come documenti JSON. Identity-based le policy e le policy utilizzate per impostare i limiti delle autorizzazioni sono documenti di policy JSON allegati a un utente o a un ruolo. Resource-based le policy sono documenti di policy JSON allegati a una risorsa. Le SCP e le RCP sono documenti di policy JSON con sintassi limitata da allegare alla radice dell'organizzazione, all' AWS Organizations unità organizzativa (OU) o a un account. Anche le ACL vengono collegate a una risorsa, ma è necessario utilizzare una sintassi differente. Le policy di sessione sono le policy JSON fornite quando si assume un ruolo o una sessione per l'utente federato.

Non è necessario conoscere la sintassi JSON. È possibile utilizzare l'editor visuale in Console di gestione AWS per creare e modificare politiche gestite dal cliente senza mai utilizzare JSON. Tuttavia, se utilizzi policy inline per gruppi o policy complesse, devi comunque creare e modificare tali policy nell'editor JSON tramite la console. Per ulteriori informazioni sull'uso dell'editor grafico, consultare Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente e Modificare le policy IAM.

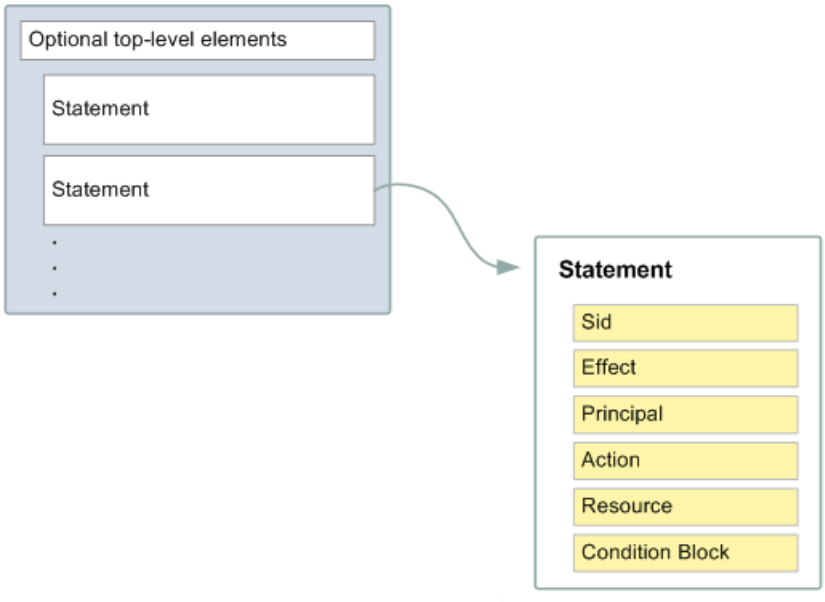
Quando crei o modifichi una policy JSON, IAM può eseguire la convalida delle policy per facilitare la creazione di una policy efficace. IAM identificherà gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente le policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

Struttura dei documenti di policy JSON

Come illustrato nella figura di seguito, un documento di policy JSON include questi elementi:

- Informazioni opzionali sulla policy nella parte superiore del documento
- Una o più istruzioni singole

Ogni istruzione include informazioni su una singola autorizzazione. Se una policy include più dichiarazioni, AWS applica una logica `OR` alle istruzioni durante la valutazione. Se a una richiesta si applicano più criteri, durante la valutazione AWS applica una logica a `OR` tutti i criteri.



Le informazioni di un'istruzione sono contenute all'interno di una serie di elementi.

- Version: specifica la versione del linguaggio di policy che desideri utilizzare. Consigliamo di utilizzare la versione `2012-10-17` più recente. Per ulteriori informazioni, consulta Elementi delle policy JSON IAM: Version
- Statement: utilizza questo elemento principale della policy come container per i seguenti elementi. Puoi includere più istruzioni in una policy.
- Sid (facoltativo): includi un ID istruzione opzionale per distinguere le varie istruzioni.
- Effect: utilizza `Allow` o `Deny` per indicare se la policy consente l'accesso o lo rifiuta.
- Principal(Obbligatorio in alcune circostanze): se crei una policy basata sulle risorse, devi indicare l'account, l'utente, il ruolo o l'entità utente AWS STS federata a cui desideri consentire o negare l'accesso. Nella creazione di una policy di autorizzazioni IAM da collegare a un utente o un ruolo, non è possibile includere questo elemento. L'entità principale è implicita come l'utente o il ruolo.
- Action: includi un elenco delle operazioni consentite o rifiutate dalla policy.
- Resource (obbligatorio solo in alcune circostanze): se crei una policy di autorizzazioni IAM, devi specificare un elenco di risorse a cui si applicano le operazioni. Se crei una policy basata sulle risorse, dipende dalla risorsa che stai utilizzando se questo elemento è obbligatorio o meno.
- Condition (facoltativo): specifica le circostanze in base alle quali la policy concede l'autorizzazione.

Per informazioni su questi e altri elementi di policy più avanzati, consulta Documentazione di riferimento degli elementi delle policy JSON IAM.

Istruzioni e policy multiple

Per definire più di un'autorizzazione per un'entità (utente o ruolo), puoi utilizzare più istruzioni in una singola policy. Puoi anche collegare più policy. Se tenti di definire più autorizzazioni in un'unica istruzione, la policy potrebbe non concedere l'accesso come previsto. Ti consigliamo di suddividere le policy in base al tipo di risorsa.

A causa delle dimensioni limitate delle policy, può essere necessario utilizzare più policy per le autorizzazioni più complesse. È inoltre consigliabile creare raggruppamenti funzionali di autorizzazioni in una policy gestita dal cliente separata. Ad esempio, crea una policy per la gestione degli utenti IAM, una per la gestione automatica e un'altra per la gestione dei bucket S3. Indipendentemente dalla combinazione di più dichiarazioni e più politiche, AWS valuta le politiche allo stesso modo.

Ad esempio, la policy seguente include tre istruzioni, ciascuna delle quali definisce un set di autorizzazioni separato all'interno di un unico account. Le istruzioni definiscono quanto segue:

- La prima istruzione, con un `Sid` (ID istruzione) di `FirstStatement`, consente all'utente con la policy collegata di modificare la propria password. In questa istruzione l'elemento `Resource` è * (che significa "tutte le risorse"). Tuttavia, in pratica, l'operazione API `ChangePassword` (o il comando CLI `change-password` equivalente) influisce solo sulla password per l'utente che effettua la richiesta.
- La seconda istruzione consente all'utente di elencare tutti i bucket Amazon S3 del proprio Account AWS. In questa istruzione l'elemento `Resource` è `"*"` (che significa "tutte le risorse"). Tuttavia, poiché le policy non concedono l'accesso alle risorse di altri account, l'utente può elencare solo i bucket del proprio Account AWS.
- La terza istruzione consente all'utente di elencare e recuperare qualsiasi oggetto all'interno di un bucket denominato `amzn-s3-demo-bucket-confidential-data`, ma solo quando l'utente viene autenticato con la multi-factor authentication (MFA). L'elemento `Condition` della policy applica l'autenticazione MFA.

  Quando un'istruzione della policy contiene un elemento `Condition`, l'istruzione risulta valida solo se per l'elemento `Condition` viene restituito un valore True. In questo caso, il risultato `Condition` è vero quando l'utente lo è. MFA-authenticated Se l'utente non lo è MFA-authenticated, il risultato `Condition` è falso. In tal caso, la terza istruzione della policy non è applicabile e l'utente non può accedere al bucket `amzn-s3-demo-bucket-confidential-data`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "FirstStatement",
      "Effect": "Allow",
      "Action": ["iam:ChangePassword"],
      "Resource": "*"
    },
    {
      "Sid": "SecondStatement",
      "Effect": "Allow",
```

```
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Sid": "ThirdStatement",
      "Effect": "Allow",
      "Action": [
        "s3:List*",
        "s3:Get*"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket-confidential-data",
        "arn:aws:s3:::amzn-s3-demo-bucket-confidential-data/*"
      ],
      "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
    }
  ]
}
```

Esempi di sintassi di policy JSON

La policy basata sulle identità riportata di seguito consente all'entità principale implicita di elencare un singolo bucket Amazon S3 denominato `amzn-s3-demo-bucket`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
  }
}
```

La policy basata su risorse riportata di seguito può essere collegata a un bucket Amazon S3. La policy consente ai membri di uno specifico di Account AWS eseguire qualsiasi azione Amazon S3 nel bucket denominato. `amzn-s3-demo-bucket` Consente qualsiasi operazione che possa essere eseguita su un bucket o sugli oggetti in esso contenuti. Poiché la policy concede la fiducia solo agli

account, i singoli utenti dell'account dovranno comunque ricevere le autorizzazioni per le operazioni Amazon S3 specificate.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Per alcuni esempi di policy per scenari comuni, consulta Esempi di policy basate su identità IAM.

Grant least privilege

Quando crei le policy IAM, segui i consigli di sicurezza standard sulla concessione di privilegi minimi o sulla concessione delle sole autorizzazioni richieste per eseguire un'attività. Determina i compiti di utenti e ruoli, quindi crea policy che consentono loro di eseguire solo tali attività.

Inizia con un set di autorizzazioni minimo e concedi autorizzazioni aggiuntive quando necessario. Questo è più sicuro che iniziare con autorizzazioni che siano troppo permissive e cercare di limitarle in un secondo momento.

In alternativa al minimo privilegio, puoi usare le autorizzazioni di policy gestite da AWS o policy con carattere jolly * per iniziare a utilizzare le policy. Considera il rischio per la sicurezza di concedere ai

principali più autorizzazioni di quelle necessarie per svolgere il proprio lavoro. Monitora tali principali per sapere quali autorizzazioni stanno utilizzando. Quindi scrivi le policy con il privilegio minimo.

IAM fornisce diverse opzioni che consentono di perfezionare le autorizzazioni concesse.

- Informazioni sui raggruppamenti a livello di accesso: puoi utilizzare i raggruppamenti a livello di accesso per comprendere il livello di accesso concesso da una policy. Le operazioni delle policy sono classificate come `List`, `Read`, `Write`, `Permissions management` o `Tagging`. Ad esempio, è possibile selezionare operazioni dai livelli di accesso `List` e `Read` per concedere accesso in sola lettura agli utenti. Per ulteriori informazioni su come utilizzare i riepiloghi delle policy per comprendere le autorizzazioni a livello di accesso, consultare Livelli di accesso nei riepiloghi delle policy.
- Convalida le policy: puoi eseguire la convalida delle policy utilizzando IAM Access Analyzer quando crei e modifichi le policy JSON. Consigliamo di rivedere e convalidare tutte le policy esistenti. Per convalidare le policy, IAM Access Analyzer fornisce oltre 100 controlli delle policy. Genera avvisi di sicurezza quando una istruzione nella tua policy consente l'accesso che consideriamo eccessivamente permissivo. È possibile utilizzare i suggerimenti utili forniti tramite gli avvisi di sicurezza mentre si lavora per concedere il minimo privilegio. Per ulteriori informazioni sui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.
- Genera una policy basata sull'attività di accesso: per ottimizzare le autorizzazioni concesse, puoi generare una policy IAM basata sull'attività di accesso per un'entità IAM (utente o ruolo). IAM Access Analyzer esamina AWS CloudTrail i tuoi log e genera un modello di policy che contiene le autorizzazioni utilizzate dall'entità nel periodo di tempo specificato. È possibile utilizzare il modello per creare una policy gestita con autorizzazioni granulari e quindi collegarla al ruolo IAM. In questo modo, concedi solo le autorizzazioni necessarie all'utente o al ruolo per interagire con le AWS risorse per il tuo caso d'uso specifico. Per ulteriori informazioni, consulta Generazione di policy per Sistema di analisi degli accessi IAM.
- Utilizza informazioni sull'ultimo accesso: un'altra funzionalità che può aiutarti con il minimo privilegio èInformazioni sull'ultimo accesso. Visualizza queste informazioni nella scheda Access Advisor nella pagina dei dettagli della console IAM per un utente, un gruppo, un ruolo o una policy IAM. Le informazioni sull'ultimo accesso includono anche informazioni sulle azioni a cui si è effettuato l'ultimo accesso per alcuni servizi, ad esempio Amazon EC2, IAM, Lambda e Amazon S3. Se accedi utilizzando le credenziali dell'account di AWS Organizations gestione, puoi visualizzare le informazioni sull'ultimo accesso al servizio nella AWS Organizations sezione della console IAM. Puoi anche utilizzare l' AWS API AWS CLI o per recuperare un report sulle ultime informazioni di accesso relative a entità o policy in IAM o. AWS Organizations Puoi utilizzare

queste informazioni per identificare le autorizzazioni non necessarie in modo da poter perfezionare il tuo IAM o AWS Organizations le tue politiche per aderire meglio al principio del privilegio minimo. Per ulteriori informazioni, consulta Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

- Esamina gli eventi dell'account in AWS CloudTrail: per ridurre ulteriormente le autorizzazioni, puoi visualizzare gli eventi del tuo account nella cronologia degli eventi. AWS CloudTrail CloudTrail i registri degli eventi includono informazioni dettagliate sugli eventi che puoi utilizzare per ridurre le autorizzazioni della politica. I log includono solo le operazioni e le risorse richieste dalle entità IAM. Per ulteriori informazioni, vedere Visualizzazione CloudTrail degli eventi nella CloudTrail console nella Guida per l'AWS CloudTrail utente.

Per ulteriori informazioni, consulta i seguenti argomenti di policy per singoli servizi, che forniscono esempi di come scrivere policy per le risorse specifiche del servizio.

- Utilizzo di policy basate sulle risorse per DynamoDB nella Guida per gli sviluppatori di Amazon DynamoDB
- Policy del bucket per Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.
- Panoramica della lista di controllo degli accessi (ACL) nella Guida per l'utente di Amazon Simple Storage Service

Policy gestite e policy inline

Quando imposti le autorizzazioni per un'identità in IAM, dovrai scegliere tra una policy gestita da AWS , una policy gestita dal cliente o una policy inline. Gli argomenti seguenti forniscono ulteriori informazioni su ciascuno dei tipi di policy basate sull'identità e su quando utilizzarli.

La tabella seguente descrive queste policy:

Tipo di policy	Description	Chi gestisce la policy?	Modificare le autorizzazioni?	Numero di principali applicati alla policy?
AWS policy gestite	Policy autonoma creata e	AWS	No	Molti

Tipo di policy	Description	Chi gestisce la policy?	Modificare le autorizzazioni?	Numero di principali applicati alla policy?
	amministrata da AWS.			
Policy gestite dal cliente	Policy che crei per casi d'uso specifici e che puoi modificare e aggiornare tutte le volte che desideri.	Utente corrente	Sì	Molti
Policy inline	Policy creata per una singola identità IAM (utente, gruppo o ruolo) che mantiene una stretta relazione uno a uno tra una policy e un'identità.	Utente corrente	Sì	One

Per istruzioni su come creare ruoli e allegare queste politiche, consulta. Creazione dei ruoli e collegamento delle policy (console)

Argomenti

- AWS policy gestite
- Policy gestite dal cliente
- Policy inline
- Scegliere tra policy gestite e policy in linea
- Convertire una policy in linea in una policy gestita

- Deprecated AWS policy gestite

AWS policy gestite

Una policy gestita da AWS è una policy autonoma che viene creata e amministrata da AWS. Una policy autonoma è una policy che ha un proprio nome della risorsa Amazon (ARN) che include il nome della policy. Ad esempio, `arn:aws:iam::aws:policy/IAMReadOnlyAccess` è una policy AWS gestita. Per ulteriori informazioni sugli ARN, consultare la pagina ARN IAM. Per un elenco delle politiche AWS gestite per Servizi AWS, vedere le politiche AWS gestite.

AWS le policy gestite consentono di assegnare comodamente le autorizzazioni appropriate a utenti, gruppi IAM e ruoli. È più veloce della scrittura delle policy in autonomia e include le autorizzazioni per molti casi d'uso comuni.

Non è possibile modificare le autorizzazioni definite nelle policy AWS gestite. AWS aggiorna occasionalmente le autorizzazioni definite in una policy AWS gestita. In questo caso, l'aggiornamento AWS ha effetto su tutte le entità principali (utenti IAM, gruppi IAM e ruoli IAM) a cui è associata la policy. AWS è più probabile che aggiorni una policy AWS gestita quando viene lanciato un nuovo AWS servizio o diventano disponibili nuove chiamate API per i servizi esistenti. Ad esempio, la policy AWS gestita denominata ReadOnlyAccess fornisce l'accesso in sola lettura a tutte Servizi AWS le risorse. Quando AWS avvia un nuovo servizio, AWS aggiorna la ReadOnlyAccess policy per aggiungere autorizzazioni di sola lettura per il nuovo servizio. Le autorizzazioni aggiornate vengono applicate a tutte le entità principali a cui la policy è collegata.

Policy AWS gestite con accesso completo: definiscono le autorizzazioni per gli amministratori del servizio concedendo l'accesso completo a un servizio. Gli esempi includono:

- AmazonDynamoDBFullAccess
- IAMFullAccess

Power-user AWS policy gestite: forniscono l'accesso completo alle Servizi AWS risorse, ma non consentono la gestione di utenti e gruppi IAM. Gli esempi includono:

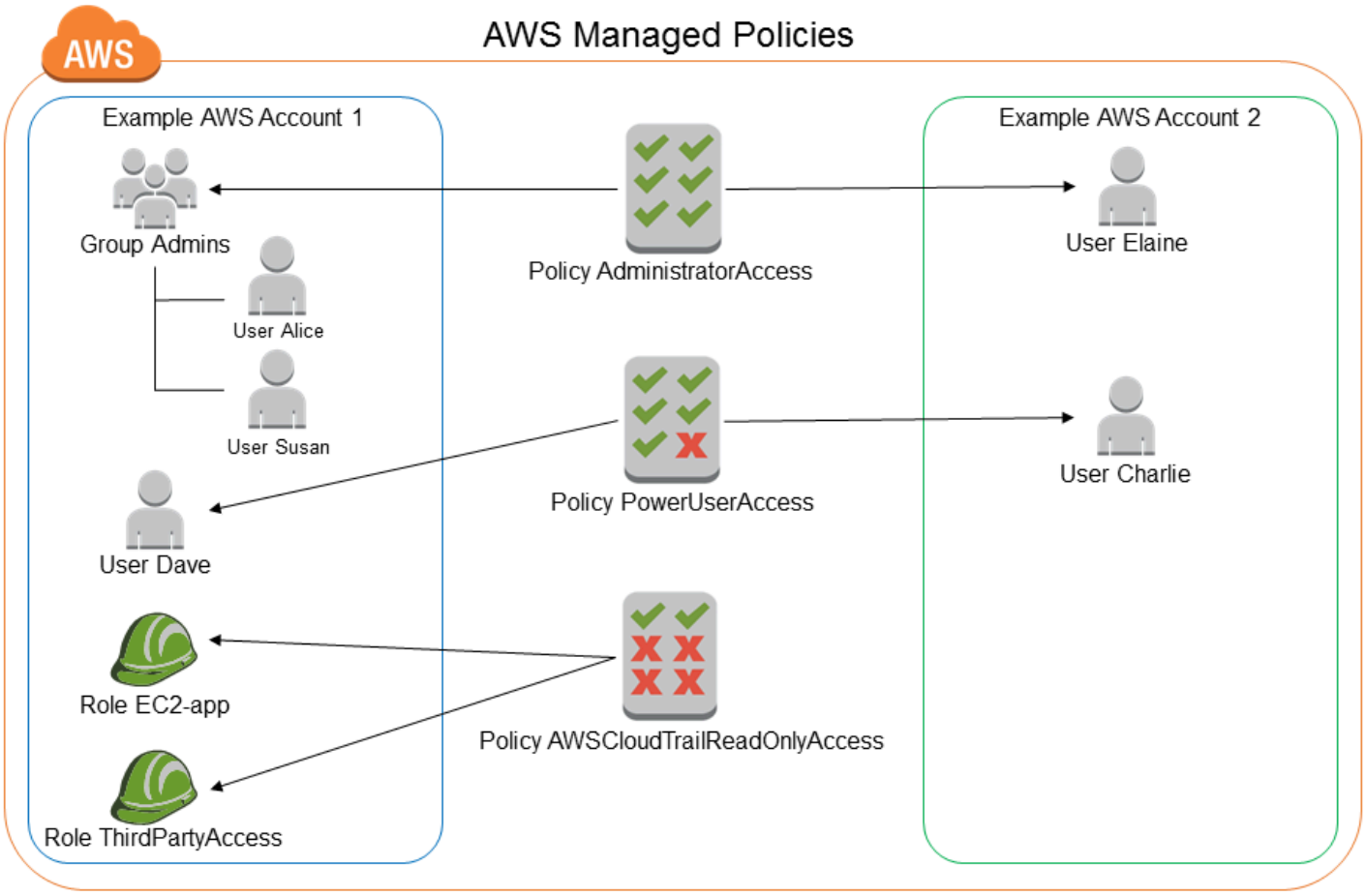
- AWSCodeCommitPowerUser
- AWSKeyManagementServicePowerUser

Partial-access AWS politiche gestite: forniscono livelli di accesso specifici Servizi AWS senza consentire la gestione delle AWS livelli di accesso autorizzazioni. Gli esempi includono:

- AmazonMobileAnalyticsWriteOnlyAccess
- AmazonEC2ReadOnlyAccess

Politiche AWS gestite dalle funzioni lavorative: queste politiche sono strettamente allineate alle funzioni lavorative comunemente utilizzate nel settore IT e facilitano la concessione delle autorizzazioni per queste funzioni lavorative. Uno dei principali vantaggi dell'utilizzo delle politiche relative alle funzioni lavorative è che vengono mantenute e aggiornate AWS man mano che vengono introdotti nuovi servizi e operazioni API. Ad esempio, la AdministratorAccess job function fornisce l'accesso completo e la delega delle autorizzazioni a tutti i servizi e le risorse presenti. AWS Si consiglia di utilizzare questa policy solo per l'amministratore dell'account. Per gli utenti esperti che richiedono l'accesso completo a tutti i servizi tranne l'accesso limitato a IAM AWS Organizations, utilizza la PowerUserAccess funzione job. Per un elenco e descrizioni delle policy delle mansioni lavorative, consulta AWS politiche gestite per le funzioni lavorative.

Il diagramma seguente illustra le policy AWS gestite. Il diagramma mostra tre politiche AWS gestite: AdministratorAccess, PowerUserAccess, e AWS CloudTrail _ ReadOnlyAccess. Si noti che una singola policy AWS gestita può essere associata a entità principali in diverse Account AWS entità principali e a diverse entità principali in un'unica Account AWS.



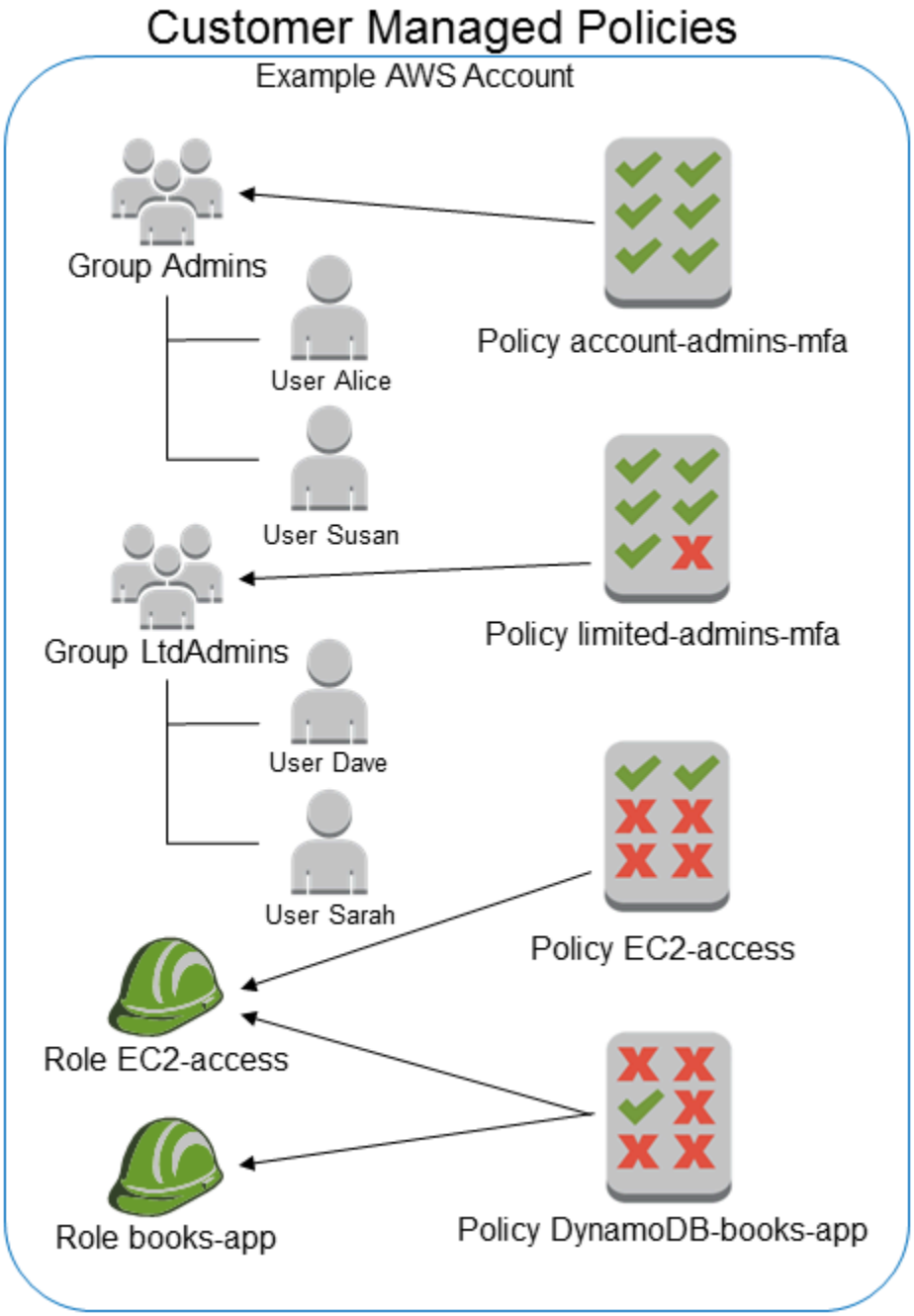
Policy gestite dal cliente

È possibile creare policy autonome da collegare alle entità principali (utenti IAM, gruppi IAM e ruoli IAM). Account AWS Puoi creare queste policy gestite dal cliente per i tuoi casi d'uso specifici e modificarle e aggiornarle tutte le volte che desideri. AWS Analogamente alle policy gestite, quando si allega una policy a un'entità principale, si assegnano all'entità le autorizzazioni definite nella policy. Quando le autorizzazioni della policy vengono aggiornate, le modifiche vengono applicate a tutte le entità principali a cui è collegata la policy.

Un ottimo modo per creare una policy gestita dal cliente è iniziare copiando una policy gestita da AWS esistente. In questo modo è possibile assicurarsi che la policy sia corretta come base ed è sufficiente personalizzarla per il proprio ambiente.

Il diagramma seguente illustra le policy gestite dal cliente. Ogni policy è un'entità in IAM con un proprio Amazon Resource Name (ARN) che include il nome della policy. Nota che la stessa policy può essere associata a più entità principali, ad esempio, la stessa DynamoDB-books-app policy è associata a due diversi ruoli IAM.

Per ulteriori informazioni, consulta Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente

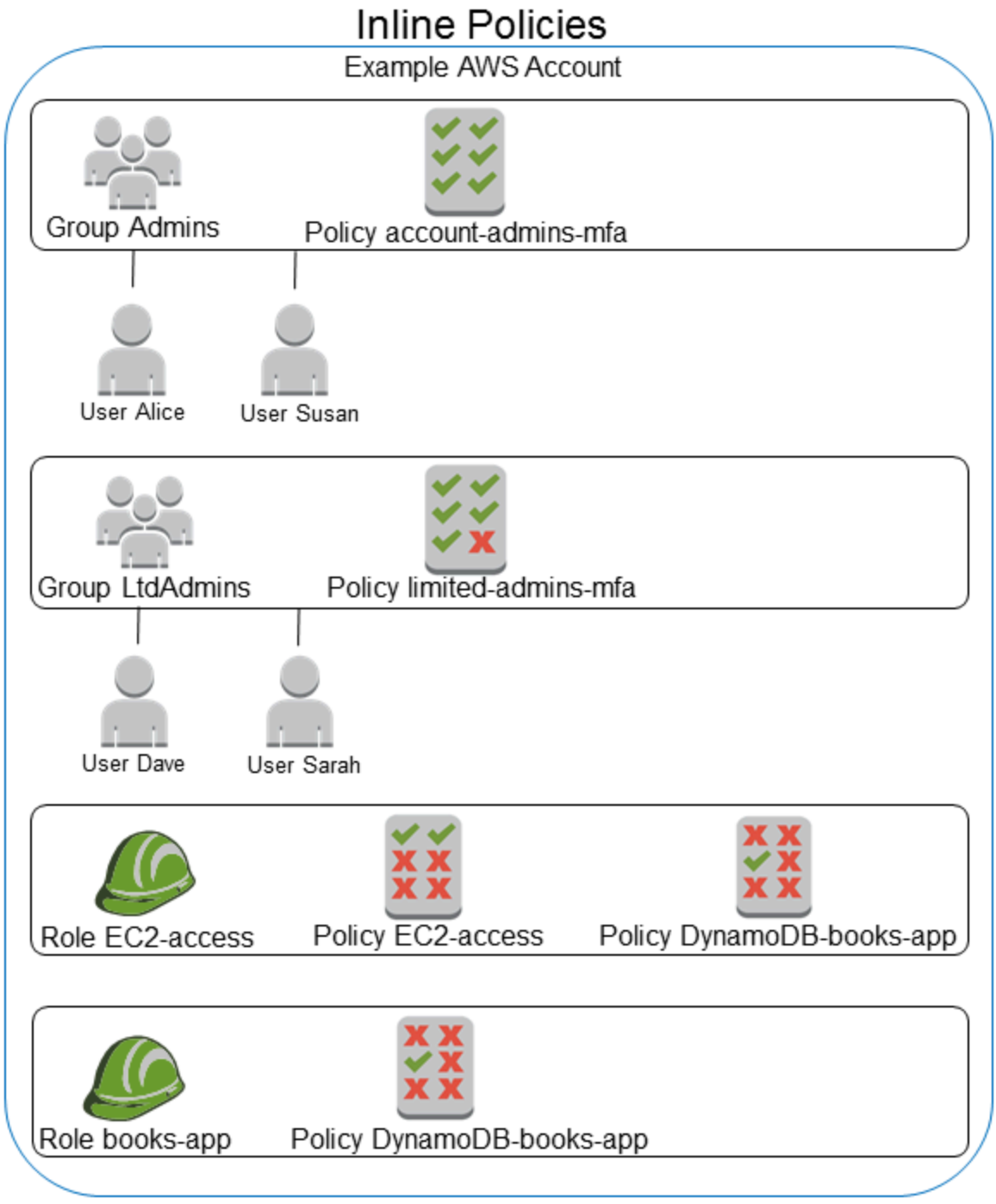


Policy inline

Una policy in linea è una policy creata per una singola identità IAM (utente, gruppo di utenti o ruolo). Le policy in linea sono utili per mantenere una stretta relazione uno a uno tra una policy e un'identità. Vengono eliminate quando elimini l'identità. È possibile creare una policy e incorporarla in un'identità,

sia quando si crea l'identità sia in un secondo momento. Se una policy può essere applicata a più di un'entità, è meglio utilizzare una policy gestita.

Il diagramma seguente illustra le policy inline. Ogni policy è parte integrante dell'utente, gruppo o ruolo. Nota che due ruoli includono la stessa policy (la DynamoDB-books-app policy), ma non condividono un'unica policy. Ogni ruolo dispone di una propria copia della policy.



Scegliere tra policy gestite e policy in linea

Al momento di scegliere tra policy gestite e policy inline, prendi in considerazione i casi d'uso. Nella maggior parte dei casi, si consiglia di usare le policy gestite anziché le policy inline.

> **Note**
>
> È possibile utilizzare insieme le policy gestite e policy inline per definire autorizzazioni comuni e univoche per un'entità principale.

Le policy gestite offrono le seguenti caratteristiche:

Riutilizzo

Una singola policy gestita può essere collegata a più entità principali (utenti, gruppi e ruoli). È possibile creare una libreria di politiche che definiscono le autorizzazioni utili per l'utente Account AWS e quindi allegarle alle entità principali in base alle esigenze.

Gestione centralizzata delle modifiche

Quando si modifica una policy gestita, la modifica viene applicata a tutte le entità principali a cui la policy è collegata. Ad esempio, se desideri aggiungere l'autorizzazione per una nuova AWS API, puoi aggiornare una politica gestita dal cliente o associare una politica AWS gestita per aggiungere l'autorizzazione. Se utilizzi una policy AWS gestita, AWS aggiorna la policy. Quando una policy gestita viene aggiornata, le modifiche vengono applicate a tutte le entità principali a cui è collegata la policy gestita. Al contrario, per modificare una policy in linea, è necessario modificare singolarmente ciascuna identità che contiene la policy in linea. Ad esempio, se un gruppo e un ruolo contengono la stessa policy inline, è necessario modificare individualmente entrambe le entità principali per modificare tale policy.

Controllo delle versioni e rollback

Quando si modifica una policy gestita dal cliente, la policy modificata non sovrascriverà la policy esistente. IAM crea invece una nuova versione della policy gestita. IAM memorizza fino a cinque versioni di una policy gestita dal cliente. È possibile utilizzare le versioni della policy per ripristinare una policy a una versione precedente, se necessario.

> **Note**
>
> Una versione di policy è diversa da un elemento `Version` della policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Per ulteriori informazioni sulle versioni di policy, consultare the section called "Controllo delle versioni delle policy IAM". Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version.

Delega della gestione delle autorizzazioni

Puoi consentire agli utenti della tua azienda Account AWS di allegare e scollegare le politiche mantenendo il controllo sulle autorizzazioni definite in tali politiche. A tale scopo, è possibile designare alcuni utenti come amministratori completi, ossia amministratori che possono creare, aggiornare ed eliminare le policy. È quindi possibile designare altri utenti come amministratori limitati. Tali amministratori limitati possono collegare delle policy ad altre entità principali, ma solo nel caso delle policy per le quali sono stati autorizzati.

Per ulteriori informazioni sulla delega della gestione delle autorizzazioni, consultare Controllo dell'accesso alle policy.

Limiti di caratteri per le policy più grandi

Il limite massimo di caratteri per le policy gestite è maggiore del limite di caratteri per le policy in linea del gruppo. Se raggiungi il limite di dimensione dei caratteri della policy in linea, puoi creare altri gruppi IAM e collegare la policy gestita al gruppo.

Per ulteriori informazioni su quote e limiti, consulta IAM e AWS STS quote.

Aggiornamenti automatici per AWS le politiche gestite

AWS mantiene le politiche AWS gestite e le aggiorna quando necessario, ad esempio per aggiungere autorizzazioni per nuovi AWS servizi, senza che l'utente debba apportare modifiche. Gli aggiornamenti vengono applicati automaticamente alle principali entità a cui è stata allegata la politica AWS gestita.

Nozioni di base sulle policy gestite

Consigliamo di utilizzare le policy che concedono il privilegio minimo o che concedono solo le autorizzazioni richieste per eseguire un processo. Il modo più sicuro per concedere il privilegio minimo consiste nello scrivere una policy gestita dal cliente solo con le autorizzazioni necessarie al

team. È necessario creare un processo per consentire al team di richiedere ulteriori autorizzazioni quando necessario. La creazione di policy gestite dai clienti IAM che forniscono al team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza.

Per iniziare ad aggiungere autorizzazioni alle tue identità IAM (utenti, gruppi di utenti e ruoli), puoi utilizzare. AWS policy gestite AWS le politiche gestite non concedono i permessi con il privilegio minimo. Considera il rischio per la sicurezza di concedere ai principali più autorizzazioni di quelle necessarie per svolgere il proprio lavoro.

Puoi allegare policy AWS gestite, incluse le funzioni lavorative, a qualsiasi identità IAM. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

Per passare alle autorizzazioni con privilegi minimi, puoi eseguire AWS Identity and Access Management Access Analyzer per monitorare i principali con policy gestite. AWS Dopo aver appreso quali autorizzazioni stanno utilizzando, puoi scrivere o generare una policy gestita dal cliente che contenga soltanto le autorizzazioni richieste per il team. È meno sicuro, ma offre maggiore flessibilità man mano che impari a utilizzare il tuo team. AWS Per ulteriori informazioni, consulta Generazione di policy per Sistema di analisi degli accessi IAM.

AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni. Per ulteriori informazioni sulle politiche AWS gestite progettate per funzioni lavorative specifiche, vedereAWS politiche gestite per le funzioni lavorative.

Per un elenco delle politiche AWS gestite, consulta la AWS Managed Policy Reference Guide.

Utilizzo delle policy inline

Le policy in linea sono utili se si desidera mantenere una stretta one-to-one relazione tra una policy e l'identità a cui viene applicata. Ad esempio, se si desidera essere certi che le autorizzazioni di una policy non vengano inavvertitamente assegnate a un'identità diversa da quella per la quale sono state concepite. Quando si utilizza una policy inline, le autorizzazioni della policy non possono essere collegate inavvertitamente a un'identità errata. Inoltre, quando si utilizza il Console di gestione AWS per eliminare tale identità, vengono eliminate anche le politiche incorporate nell'identità perché fanno parte dell'entità principale.

Convertire una policy in linea in una policy gestita

Se disponi di policy inline nell'account, puoi convertirle in policy gestite. A tale scopo, copia la policy in una nuova policy gestita, collega la nuova policy all'identità che ha la policy inline, quindi elimina la policy inline.

Conversione di una policy inline in una policy gestita

Per convertire una policy inline in una policy gestita

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione scegli Gruppi di utenti, Utenti o Ruoli.
3. Nell'elenco, scegli il nome del gruppo di utenti, dell'utente o del ruolo con la policy da rimuovere.
4. Scegli la scheda Autorizzazioni.
5. Per i gruppi IAM, seleziona il nome della policy in linea che desideri rimuovere. Per utenti e ruoli, scegli Mostra ***n*** altro, se necessario, quindi espandi la policy in linea che desideri rimuovere.
6. Scegli Copia per copiare il documento della policy in formato JSON.
7. Nel riquadro di navigazione, scegli Policy.
8. Seleziona Crea policy, quindi scegli l'opzione JSON.
9. Sostituisci il testo esistente con il testo della policy JSON, quindi scegli Verifica policy.
10. Immetti un nome e una descrizione facoltativa per la policy, quindi scegli Crea policy.
11. Nel pannello di navigazione, scegli Gruppi di utenti, Utenti o Ruoli e scegli di nuovo il nome del gruppo di utenti, dell'utente o del ruolo con la policy da rimuovere.
12. Seleziona la scheda Autorizzazioni e scegli Aggiungi autorizzazioni.
13. Per i gruppi IAM, seleziona la casella di controllo accanto al nome della nuova policy, seleziona Aggiungi autorizzazioni, quindi scegli Collega policy. Per gli utenti o i ruoli, scegliere Add permissions (Aggiungi autorizzazioni). Nella pagina successiva, scegli Collega direttamente policy esistenti, seleziona la casella di controllo accanto al nome della nuova policy, scegli Successivo, quindi seleziona Aggiungi autorizzazioni.

    Sarai riportato alla pagina Riepilogo per l'utente, il gruppo di utenti o il ruolo.
14. Seleziona la casella di controllo accanto alla policy in linea che desideri rimuovere, quindi scegli Rimuovi.

Deprecated AWS policy gestite

Per semplificare l'assegnazione delle autorizzazioni, AWS fornisce policy gestite, politiche predefinite pronte per essere collegate agli utenti, ai gruppi e ai ruoli IAM.

A volte è AWS necessario aggiungere una nuova autorizzazione a una policy esistente, ad esempio quando viene introdotto un nuovo servizio. L'aggiunta di una nuova autorizzazione a una policy esistente non disturba o rimuove qualsiasi caratteristica o possibilità.

Tuttavia, AWS potrebbe scegliere di creare una nuova politica quando le modifiche necessarie potrebbero avere un impatto sui clienti se applicate a una politica esistente. Ad esempio, la rimozione delle autorizzazioni da una policy esistente potrebbe violare le autorizzazioni di qualsiasi entità o applicazione IAM dalla quale dipendeva, disturbando potenzialmente un'operazione critica.

Pertanto, quando tale modifica è necessaria, AWS crea una politica completamente nuova con le modifiche necessarie e la mette a disposizione dei clienti. La policy vecchia viene contrassegnata come obsoleta. Per ulteriori informazioni, vedere Policy AWS gestite obsolete nella AWS Managed Policy Reference Guide.

Establish permissions guardrails using data perimeters

Le barriere perimetrali dei dati sono pensate per fungere da confini sempre attivi per aiutare a proteggere i dati su un'ampia gamma di account e risorse. AWS I perimetri dei dati seguono le best practice di sicurezza IAM per stabilire guardrail di autorizzazioni su più account. Questi guardrail di autorizzazione a livello di organizzazione non sostituiscono i controlli di accesso dettagliati esistenti. Funzionano invece come controlli di accesso generalizzati che aiutano a migliorare la strategia di sicurezza assicurando che utenti, ruoli e risorse aderiscano a una serie di standard di sicurezza definiti.

Un perimetro di dati è un insieme di barriere di autorizzazione nell' AWS ambiente in uso che aiutano a garantire che solo le identità attendibili accedano alle risorse attendibili dalle reti previste.

- Identità affidabili: i responsabili (ruoli o utenti IAM) dei tuoi AWS account e servizi che agiscono per tuo conto. AWS
- Risorse affidabili: risorse di proprietà AWS dei tuoi account o dei AWS servizi che agiscono per tuo conto.
- Reti previste: i tuoi data center locali e i cloud privati virtuali (VPC) o le reti di AWS servizi che agiscono per tuo conto.

> **Note**
>
> In alcuni casi, potrebbe essere necessario estendere il perimetro dei dati per includere anche l'accesso da parte dei partner commerciali di fiducia. È necessario prendere in

considerazione tutti i modelli di accesso ai dati previsti quando si crea una definizione di identità attendibili, risorse attendibili e reti previste specifiche per l'azienda e l'utilizzo dei Servizi AWS.

I controlli perimetrali dei dati devono essere trattati come qualsiasi altro controllo di sicurezza nell'ambito del programma di sicurezza delle informazioni e di gestione del rischio. Ciò significa che è necessario eseguire un'analisi delle minacce per identificare i potenziali rischi all'interno dell'ambiente cloud e quindi, in base ai propri criteri di accettazione del rischio, selezionare e implementare i controlli perimetrali dei dati appropriati. Per definire meglio l'approccio iterativo basato sul rischio all'implementazione del perimetro dei dati, è necessario comprendere quali rischi e vettori di minaccia vengono affrontati dai controlli perimetrali dei dati, nonché le priorità di sicurezza.

Controlli perimetrali dei dati

I controlli granulari sul perimetro dei dati aiutano a raggiungere sei obiettivi di sicurezza distinti su tre perimetri di dati attraverso l'implementazione di diverse combinazioni di Tipi di policy e chiavi di condizioni.

Perimetro	Obiettivo di controllo	Utilizzo	Applicato il	Chiavi di contesto della condizione globale
Identità	Solo le identità attendibili possono accedere alle mie risorse	RCP	Resources	leggi: ID PrincipalOrg era: Principal OrgPaths
	Nella mia rete sono consentite solo identità attendibili	Policy degli endpoint VPC	Rete	era: Principal Account era: Principal IsAwsService aws: SourceOrg ID

Perimetro	Obiettivo di controllo	Utilizzo	Applicato il	Chiavi di contesto della condizione globale
				era: SourceOrgPath era: SourceAccount
Resources	Le tue identità possono accedere solo a risorse attendibili	SCP	Identità	aws: ResourceOrg ID era: ResourceOrgPaths era: ResourceAccount
	Dalla rete è possibile accedere solo a risorse attendibili	Policy degli endpoint VPC	Rete	
Rete	Le tue identità possono accedere alle risorse solo dalle reti previste	SCP	Identità	era: SourceIp era: SourceVpc era: SourceVpce era: VpceAccount era: VpceOrgPaths aws: VpceOrg ID aws:ViaAWSService era: PrincipalIsAwsService
	Le tue risorse sono accessibili solo dalle reti previste	RCP	Resources	

Puoi pensare ai perimetri dei dati come alla creazione di un confine preciso attorno ai dati per prevenire schemi di accesso non intenzionali. Sebbene i perimetri dei dati possano impedire ampi accessi involontari, è comunque necessario prendere decisioni granulari sul controllo degli accessi. La definizione di un perimetro di dati non riduce la necessità di ottimizzare continuamente le autorizzazioni utilizzando strumenti come Sistema di analisi degli accessi IAM come parte del percorso verso il privilegio minimo.

Per applicare i controlli perimetrali dei dati su risorse che attualmente non sono supportate dagli RCP, puoi utilizzare policy basate sulle risorse collegate direttamente alle risorse. Per un elenco di servizi che supportano le RCP e le policy basate sulle risorse, consulta Policy di controllo delle risorse (RCP) e AWS servizi supportati da IAM.

Per applicare i controlli perimetrali di rete, ti consigliamo di utilizzare `aws:VpceOrgID`, `aws:VpceOrgPaths` e `aws:VpceAccount` solo se tutti i servizi a cui desideri limitare l'accesso sono attualmente supportati. L'utilizzo di queste chiavi di condizione con servizi non supportati può portare a esiti di autorizzazione non intenzionali. Per un elenco di servizi che supportano le chiavi, consulta AWS chiavi di contesto delle condizioni globali. Se hai bisogno di applicare i controlli su una gamma più ampia di servizi, prendi in considerazione l'utilizzo di `aws:SourceVpc` e `aws:SourceVpce` in alternativa.

Perimetro di identità

Un perimetro di identità è un insieme di controlli di accesso preventivi a grana grossolana che aiutano a garantire che solo le identità affidabili possano accedere alle tue risorse e che solo le identità affidabili siano consentite dalla tua rete. Le identità attendibili di solito includono i principali (ruoli o utenti) degli account e dei servizi che agiscono per conto dell'utente. AWS AWS Tutte le altre identità sono considerate non attendibili e sono impedite dal perimetro dell'identità, a meno che non venga concessa un'eccezione esplicita.

Le seguenti chiavi di condizione globali aiutano a far rispettare i controlli perimetrali delle identità in base alla tua definizione delle identità attendibili. Utilizza queste chiavi nelle policy di controllo delle risorse per limitare l'accesso alle risorse o nelle policy degli endpoint VPC per limitare l'accesso alle tue reti.

Identità di tua proprietà

Puoi utilizzare le seguenti chiavi di condizione per definire i principi IAM che crei e gestisci nel tuo. Account AWS

- aws: PrincipalOrg ID— Puoi utilizzare questa chiave di condizione per assicurarti che i principali IAM che effettuano la richiesta appartengano all'organizzazione specificata in. AWS Organizations
- era: PrincipalOrgPaths— È possibile utilizzare questa chiave di condizione per assicurarsi che l'utente IAM, il ruolo IAM, l'entità utente AWS STS federata, l'entità federata SAML, l'entità federata OIDC o che Utente root dell'account AWS effettua la richiesta appartengano all'unità organizzativa (OU) specificata in. AWS Organizations
- era: PrincipalAccount— È possibile utilizzare questa chiave di condizione per assicurarsi che le risorse siano accessibili solo dall'account principale specificato nella policy.

Identità di AWS servizi che agiscono per conto dell'utente

Puoi utilizzare le seguenti chiavi di condizione per consentire ai AWS servizi di utilizzare le proprie identità per accedere alle tue risorse quando agiscono per tuo conto.

- era: PrincipalIsAWSServicee aws: ID SourceOrg (o leggi: SourceOrgPaths eera: SourceAccount): puoi utilizzare queste chiavi di condizione per assicurarti che quando i responsabili Servizio AWS accedono alle tue risorse, lo facciano solo per conto di una risorsa nell'organizzazione, nell'unità organizzativa o in un account specificati. AWS Organizations

Per ulteriori informazioni, vedi Stabilire un perimetro di dati su AWS: Consenti solo a identità affidabili di accedere ai dati aziendali.

Perimetro di risorse

Un perimetro di risorse è un insieme di controlli di accesso preventivi a grana grossa che aiutano a garantire che le tue identità possano accedere solo a risorse affidabili e che solo le risorse affidabili siano accessibili dalla tua rete. Le risorse attendibili di solito includono risorse di proprietà dei tuoi AWS account o di servizi che agiscono per tuo conto. AWS

Le seguenti chiavi di condizione globali aiutano a far rispettare i controlli perimetrali delle risorse in base alla tua definizione delle risorse attendibili. Utilizza queste chiavi nelle policy di controllo dei servizi (SCP) per limitare le risorse a cui possono accedere le tue identità o nelle policy degli endpoint VPC per limitare le risorse a cui è possibile accedere dalle tue reti.

Risorse di tua proprietà

Puoi utilizzare le seguenti chiavi di condizione per definire AWS le risorse che crei e gestisci nel tuo Account AWS.

- aws: ResourceOrg ID— È possibile utilizzare questa chiave di condizione per assicurarsi che la risorsa a cui si accede appartenga all'organizzazione specificata in AWS Organizations.
- era: ResourceOrgPaths— È possibile utilizzare questa chiave di condizione per assicurarsi che la risorsa a cui si accede appartenga all'unità organizzativa (OU) specificata in AWS Organizations.
- era: ResourceAccount— È possibile utilizzare questa chiave di condizione per assicurarsi che la risorsa a cui si accede appartenga a quella specificata Account AWS.

Risorse di AWS servizi che agiscono per conto dell'utente

In alcuni casi, potrebbe essere necessario consentire l'accesso a risorse AWS di proprietà, risorse che non appartengono alla propria organizzazione e alle quali accedono i responsabili o i AWS servizi che agiscono per conto dell'utente. Per ulteriori informazioni su questi scenari, vedere Stabilire un perimetro di dati su AWS: consentire solo risorse attendibili della mia organizzazione.

Perimetro di rete

Un perimetro di rete è un insieme di controlli di accesso preventivi a grana grossa che aiutano a garantire che le tue identità possano accedere alle risorse solo dalle reti previste e che le tue risorse siano accessibili solo dalle reti previste. Le reti previste solitamente includono i data center on-premises, i cloud privati virtuali (VPC) e le reti dei servizi AWS che agiscono per tuo conto.

Le seguenti chiavi di condizione globali aiutano a far rispettare i controlli perimetrali della rete in base alla tua definizione delle reti previste. Utilizza queste chiavi nelle policy di controllo dei servizi (SCP) per limitare le reti da cui le identità possono comunicare o nelle policy di controllo delle risorse (RCP) per limitare l'accesso alle risorse alle reti previste.

Reti di tua proprietà

Puoi utilizzare le seguenti chiavi di condizione per definire le reti che i dipendenti e le applicazioni dovrebbero utilizzare per accedere alle tue risorse, come l'intervallo CIDR IP aziendale e i tuoi VPC.

- era: SourceIp— È possibile utilizzare questa chiave di condizione per assicurarsi che l'indirizzo IP del richiedente rientri in un intervallo IP specificato.
- era: SourceVpc— È possibile utilizzare questa chiave di condizione per assicurarsi che l'endpoint VPC attraversato dalla richiesta appartenga al VPC specificato.
- era: SourceVpce— È possibile utilizzare questa chiave di condizione per assicurarsi che la richiesta viaggi attraverso l'endpoint VPC specificato.

- era: VpceAccount— È possibile utilizzare questa chiave di condizione per assicurarsi che le richieste arrivino tramite gli endpoint VPC di proprietà dell'account specificato. AWS
- era: VpceOrgPaths— È possibile utilizzare questa chiave di condizione per assicurarsi che le richieste arrivino tramite endpoint VPC di proprietà di account che appartengono all'unità organizzativa (OU) specificata in. AWS Organizations
- aws: VpceOrg ID— È possibile utilizzare questa chiave di condizione per assicurarsi che le richieste arrivino tramite endpoint VPC di proprietà degli account dell'organizzazione specificata in. AWS Organizations

`aws:VpceAccount`, `aws:VpceOrgPaths` e `aws:VpceOrgID` sono particolarmente utili per implementare controlli perimetrali di rete che si adattano automaticamente all'utilizzo degli endpoint VPC, senza che si renda necessario aggiornare le policy quando si creano nuovi endpoint. Per un elenco di Servizi AWS che supportano queste chiavi, consulta AWS chiavi di contesto delle condizioni globali.

Reti di AWS servizi che agiscono per tuo conto

Puoi utilizzare le seguenti chiavi di condizione per consentire ai AWS servizi di accedere alle tue risorse dalle loro reti quando agiscono per tuo conto.

- aws:ViaAWSService— Puoi utilizzare questa chiave di condizione per assicurarti che Servizi AWS possa effettuare richieste per conto dell'utente principale Inoltro delle sessioni di accesso (FAS).
- era: PrincipalIsAWSService— Puoi utilizzare questa chiave di condizione per assicurarti che Servizi AWS possa accedere alle tue risorse utilizzandoAWS responsabili del servizio.

Esistono altri scenari in cui è necessario consentire l'accesso a Servizi AWS che accedono alle risorse dall'esterno della rete. Per ulteriori informazioni, vedere Stabilire un perimetro di dati su AWS: consentire l'accesso ai dati aziendali solo dalle reti previste.

Risorse per ulteriori informazioni sui perimetri di dati

Le seguenti risorse possono rivelarsi utili per saperne di più sui perimetri di dati su AWS.

- Perimetri di dati attivi AWS: scopri i perimetri dei dati e i loro vantaggi e casi d'uso.
- Serie di post sul blog: Stabilire un perimetro di dati AWS: questi post del blog forniscono linee guida prescrittive sulla definizione del perimetro dei dati su larga scala, comprese le principali considerazioni sulla sicurezza e sull'implementazione.

- Esempi di politiche relative al perimetro dei dati: questo GitHub archivio contiene esempi di policy che coprono alcuni modelli comuni per aiutarti a implementare un perimetro di dati. AWS
- Supporto per il perimetro dei dati: questo strumento consente di progettare e anticipare l'impatto dei controlli perimetrali dei dati analizzando l'attività di accesso nei log AWS CloudTrail.

Limiti delle autorizzazioni per le entità IAM

AWS supporta i limiti delle autorizzazioni per le entità IAM (utenti o ruoli). Un limite delle autorizzazioni è una funzione avanzata per l'utilizzo di una policy gestita per impostare il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM. Il limite delle autorizzazioni di un'entità consente di eseguire solo le operazioni consentite dalle sue policy basate su identità e dai suoi limiti delle autorizzazioni.

Per ulteriori informazioni sui tipi di policy, consulta Tipi di policy.

> **Important**
>
> Non utilizzare istruzioni di policy basate sulle risorse che includono un elemento di policy `NotPrincipal` con effetto `Deny` per gli utenti o i ruoli IAM ai quali è collegata una policy con limite delle autorizzazioni. L'elemento `NotPrincipal` con effetto `Deny` rifiuterà sempre qualsiasi principale IAM al quale è collegata una policy con limite delle autorizzazioni, indipendentemente dai valori specificati nell'elemento `NotPrincipal`. Ciò fa sì che alcuni utenti o ruoli IAM che altrimenti avrebbero accesso alla risorsa perdano l'accesso.
> Ti consigliamo di modificare le istruzioni di policy basate sulle risorse di modo che, per limitare l'accesso, utilizzino l'operatore di condizione `ArnNotEquals` con la chiave di contesto `aws:PrincipalArn` anziché l'elemento `NotPrincipal`. Per ulteriori informazioni sull'elemento `NotPrincipal`, consulta la pagina AWS Elementi della policy JSON: NotPrincipal.

Puoi utilizzare una policy AWS gestita o una policy gestita dal cliente per impostare il limite per un'entità IAM (utente o ruolo). La policy limita il numero massimo di autorizzazioni per l'utente o il ruolo.

Ad esempio, supponiamo che l'utente IAM indicato `Shirley` debba essere autorizzato a gestire solo Amazon S3 CloudWatch, Amazon e Amazon EC2. Per applicare la regola, puoi utilizzare la policy seguente per impostare il limite delle autorizzazioni per l'utente `Shirley`:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "cloudwatch:*",
                "ec2:*"
            ],
            "Resource": "*"
        }
    ]
}
```

Quando utilizzi una policy per impostare il limite delle autorizzazioni per un utente, questa limita le autorizzazioni dell'utente, ma non le fornisce di per sé. In questo esempio, la policy imposta le autorizzazioni massime per tutte `Shirley` le operazioni in Amazon S3 e Amazon EC2. CloudWatch Shirley non può eseguire operazioni negli altri servizi, incluso IAM, anche se dispone di una policy di autorizzazione che lo consente. Ad esempio, prova ad aggiungere la policy seguente all'utente `Shirley`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:CreateUser",
    "Resource": "*"
  }
}
```

Questa policy consente la creazione di un utente in IAM. Se colleghi questa policy di autorizzazione all'utente `Shirley` e Shirley tenta di creare un utente, l'operazione ha esito negativo. Non riesce perché il limite delle autorizzazioni non consente l'operazione `iam:CreateUser`. Date queste due

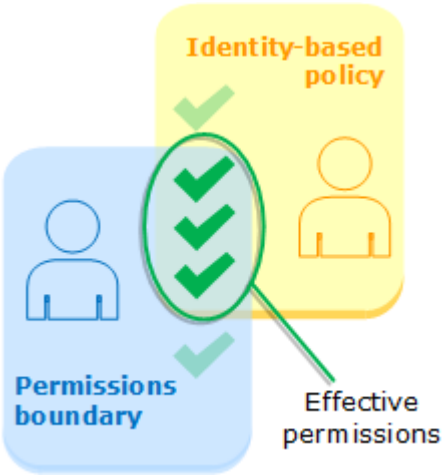
policy, Shirley non ha il permesso di eseguire alcuna operazione in AWS. È necessario aggiungere una policy di autorizzazioni diversa per consentire operazioni in altri servizi, ad esempio Amazon S3. In alternativa, è possibile aggiornare il limite delle autorizzazioni per consentirle di creare un utente in IAM.

Valutazione delle autorizzazioni valide con i limiti

Il limite delle autorizzazioni per un'entità IAM (utente o ruolo) imposta il numero massimo di autorizzazioni che è possibile concedere all'entità. Questo può influire sulle autorizzazioni valide per l'utente o il ruolo. Le autorizzazioni valide per un'entità sono quelle concesse da tutte le policy che interessano l'utente o il ruolo. In un account, le autorizzazioni per un'entità possono essere influenzate da policy basate su identità, policy basate su risorse, limiti delle autorizzazioni, SCP AWS Organizations o policy di sessione. Per ulteriori informazioni sui diversi tipi di policy, consulta Politiche e autorizzazioni in AWS Identity and Access Management.

Se uno di questi tipi di policy rifiuta esplicitamente l'accesso per un'operazione, la richiesta viene rifiutata. Le autorizzazioni concesse a un'entità in base a diversi tipi di autorizzazioni sono più complesse. Per maggiori dettagli su come AWS valuta le politiche, consulta. Logica di valutazione delle policy

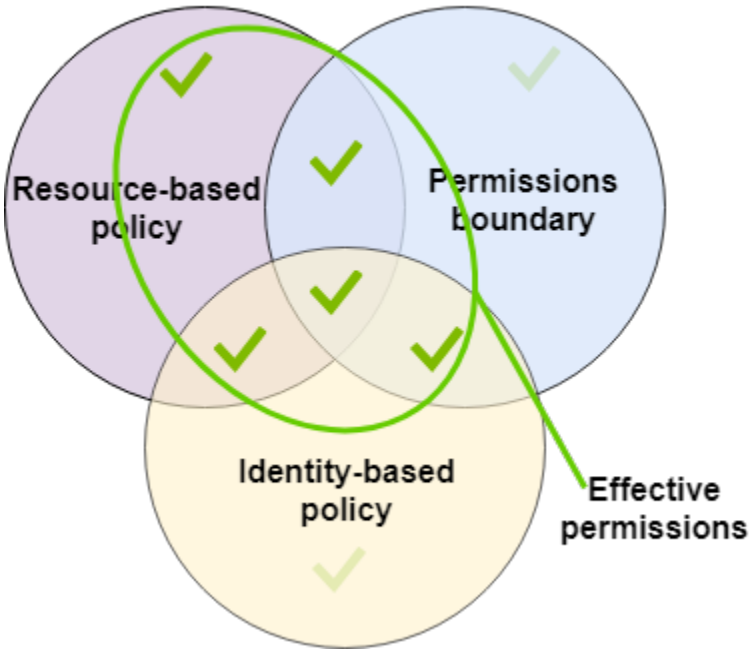
Identity-based politiche con limiti: Identity-based le politiche sono politiche in linea o gestite associate a un utente, a un gruppo di utenti o a un ruolo. Identity-based le politiche concedono l'autorizzazione all'entità e i limiti delle autorizzazioni limitano tali autorizzazioni. Le autorizzazioni effettive sono l'intersezione di entrambi i tipi di policy. Un rifiuto esplicito in una di queste policy sostituisce l'autorizzazione.



Resource-based politiche: le Resource-based politiche controllano il modo in cui il principale specificato può accedere alla risorsa a cui è collegata la politica.

Resource-based politiche per gli utenti IAM

All'interno dello stesso account, le politiche basate sulle risorse che concedono autorizzazioni a un ARN utente IAM (che non è una sessione principale utente AWS STS federata) non sono limitate da un rifiuto implicito di una politica basata sull'identità o sui limiti delle autorizzazioni.



Resource-based politiche per i ruoli IAM

Ruolo IAM: Resource-based le politiche che concedono autorizzazioni a un ruolo IAM ARN sono limitate da una negazione implicita in un limite di autorizzazioni o in una policy di sessione.

Sessione come ruolo IAM: all'interno dello stesso account, le policy basate sulle risorse che concedono le autorizzazioni all'ARN della sessione come ruolo IAM concedono le autorizzazioni direttamente alla sessione come ruolo assunto. Le autorizzazioni concesse direttamente a una sessione non sono limitate da un rifiuto implicito in una policy basata su identità, da un limite delle autorizzazioni o da una policy di sessione. Quando si assume un ruolo e si effettua una richiesta, il principale che effettua la richiesta è l'ARN della sessione come ruolo IAM e non l'ARN del ruolo stesso.

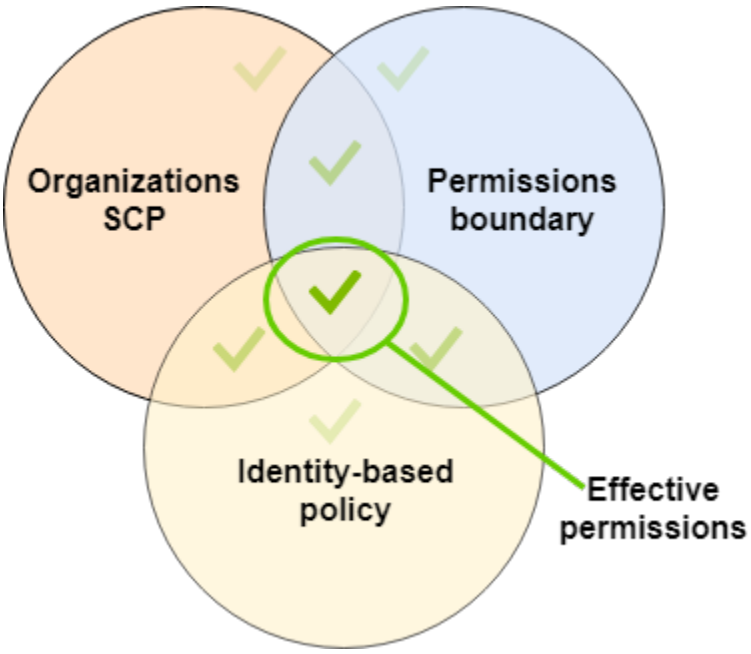
Resource-based politiche per le sessioni principali degli utenti federati AWS STS

AWS STS sessioni principali utente federate: una sessione principale utente AWS STS federata è una sessione creata chiamando. `GetFederationToken` Quando un utente federato effettua una richiesta, il principale che effettua la richiesta è l'ARN dell'utente federato e non l'ARN dell'utente IAM che ha eseguito la federazione. All'interno dello stesso account, le policy basate sulle

risorse che concedono le autorizzazioni all'ARN dell'utente federato concedono le autorizzazioni direttamente alla sessione. Le autorizzazioni concesse direttamente a una sessione non sono limitate da un rifiuto implicito in una policy basata su identità, da un limite delle autorizzazioni o da una policy di sessione.

Tuttavia, se una policy basata sulle risorse concede l'autorizzazione all'ARN dell'utente IAM che ha effettuato la federazione, le richieste effettuate dall'utente principale federato durante la sessione sono limitate da un AWS STS rifiuto implicito in un limite di autorizzazione o in una policy di sessione.

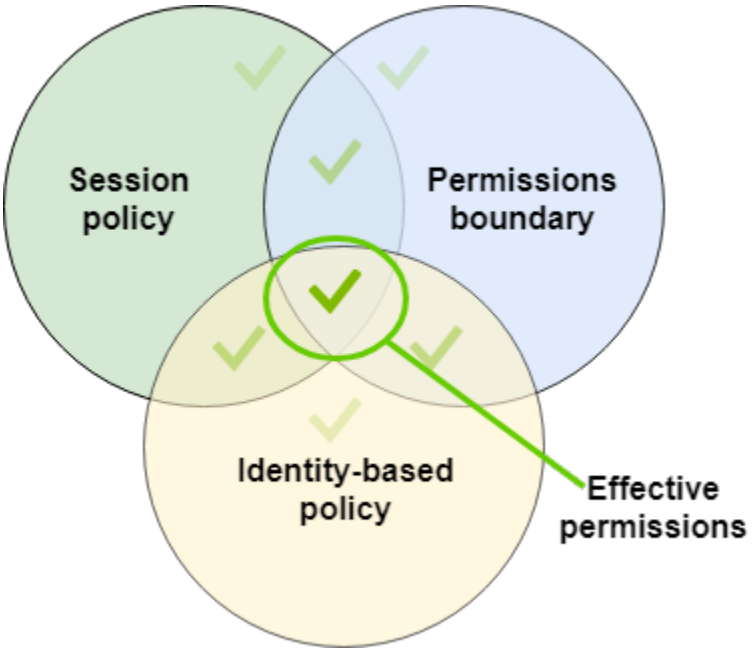
AWS Organizations SCP: Account AWS gli SCP vengono applicati a un intero. Limitano le autorizzazioni per ogni richiesta effettuata da un'entità principale all'interno dell'account. Un'entità IAM (utente o ruolo) può effettuare una richiesta che è influenzata da una SCP, un limite delle autorizzazioni e una policy basata su identità. In questo caso, la richiesta è consentita solo se tutti e tre i tipi di policy la consentono. Le autorizzazioni effettive sono l'intersezione di tutti e tre i tipi di policy. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione.



Puoi scoprire se il tuo account è un membro di un'organizzazione in AWS Organizations. I membri dell'organizzazione potrebbero essere influenzati da una SCP. Per visualizzare questi dati utilizzando il AWS CLI comando o l'operazione AWS API, devi disporre delle autorizzazioni per l'azione per la tua entità. `organizations:DescribeOrganization` AWS Organizations È necessario disporre di autorizzazioni aggiuntive per eseguire l'operazione nella AWS Organizations console. Per sapere

se un SCP sta negando l'accesso a una richiesta specifica o per modificare le tue autorizzazioni effettive, contatta l'amministratore. AWS Organizations

Policy di sessione - Le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni per una sessione provengono dall'entità IAM (utente o ruolo) utilizzata per creare la sessione e dalla policy di sessione. Le autorizzazioni della policy basata su identità dell'entità sono limitate dalla policy di sessione e dal limite delle autorizzazioni. Le autorizzazioni effettive per questo set di tipi di policy sono l'intersezione di tutti e tre i tipi di policy. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sulle policy di sessione, consulta la sezione relativa alle policy di sessione.



Delega di responsabilità ad altri mediante i limiti delle autorizzazioni

Puoi utilizzare il limiti delle autorizzazioni per delegare le attività di gestione delle autorizzazioni, ad esempio la creazione di utenti, agli utenti IAM nel tuo account. Questo consente ad altri di eseguire operazioni a tuo nome all'interno di un limite specifico di autorizzazioni.

Ad esempio, supponiamo che María sia l'amministratore di. X-Company Account AWS María vuole delegare l'attività di creazione di utenti a Zhang. Tuttavia, deve accertarsi che gli utenti creati da Zhang siano conformi alle seguenti regole aziendali:

- Gli utenti non possono utilizzare IAM per creare o gestire utenti, gruppi, ruoli o policy.

- Agli utenti viene rifiutato l'accesso al bucket `logs` di Amazon S3 e all'istanza `i-1234567890abcdef0` di Amazon EC2.
- Gli utenti non possono rimuovere le proprie policy limite.

Per applicare queste regole, María completa le attività seguenti, i cui dettagli sono riportati di seguito:

1. María crea la policy gestita `XCompanyBoundaries` da utilizzare come limite delle autorizzazioni per tutti i nuovi utenti nell'account.
2. María crea la policy gestita `DelegatedUserBoundary` e la assegna come limite delle autorizzazioni per Zhang. Maria prende nota dell'ARN del suo utente amministratore e lo usa nel criterio per impedire a Zhang di accedervi.
3. María crea la policy gestita `DelegatedUserPermissions` e la collega alla policy di autorizzazione per Zhang.
4. María comunica a Zhang le sue nuove responsabilità e limitazioni.

Attività 1: María deve prima creare una policy gestita per definire il limite per i nuovi utenti. María deve consentire a Zhang di concedere agli utenti le policy di autorizzazione necessarie, ma vuole che tali utenti abbiano delle limitazioni. A tale scopo, crea questa policy gestita dal cliente, denominata `XCompanyBoundaries`. Questa policy esegue le seguenti operazioni:

- Consente agli utenti l'accesso completo a diversi servizi
- Consente l'accesso autonomo limitato alla console IAM. Ciò significa che è possibile modificare la password dopo aver effettuato l'accesso alla console. Non è possibile impostare la password iniziale. Per consentire questa operazione, aggiungere l'operazione `"*LoginProfile"` all'istruzione `AllowManageOwnPasswordAndAccessKeys`.
- Rifiuta agli utenti l'accesso al bucket di log Amazon S3 o all'istanza Amazon EC2 `i-1234567890abcdef0`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ServiceBoundaries",
```

```
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "cloudwatch:*",
                "ec2:*",
                "dynamodb:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowIAMConsoleForCredentials",
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers",
                "iam:GetAccountPasswordPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnPasswordAndAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:*AccessKey*",
                "iam:ChangePassword",
                "iam:GetUser",
                "iam:*ServiceSpecificCredential*",
                "iam:*SigningCertificate*"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "DenyS3Logs",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::logs",
                "arn:aws:s3:::logs/*"
            ]
        },
        {
            "Sid": "DenyEC2Production",
            "Effect": "Deny",
            "Action": "ec2:*",
            "Resource": "arn:aws:ec2:*:*:instance/i-1234567890abcdef0"
```

```
        }
    ]
}
```

Ogni istruzione svolge una funzione diversa:

1. La `ServiceBoundaries` dichiarazione di questa politica consente l'accesso completo ai AWS servizi specificati. Ciò significa che le operazioni di un nuovo utente in questi servizi sono limitate solo dalle policy di autorizzazione collegate all'utente.
2. La dichiarazione `AllowIAMConsoleForCredentials` consente l'accesso per elencare tutti gli utenti IAM. Questo accesso è necessario per navigare nella pagina Users (Utenti) nella Console di gestione AWS. Inoltre, consente di visualizzare i requisiti associati alle password per l'account, necessari per modificare la password.
3. L'istruzione `AllowManageOwnPasswordAndAccessKeys` consente agli utenti di gestire solo le proprie chiavi di accesso a livello di programmazione e le password della console. Questo è importante se Zhang o un altro amministratore concede a un nuovo utente una policy di autorizzazione con accesso IAM completo. In tal caso, l'utente può modificare le proprie autorizzazioni o quelle di altri utenti. Questa istruzione impedisce che ciò si verifichi.
4. L'istruzione `DenyS3Logs` nega esplicitamente l'accesso al bucket `logs`.
5. L'istruzione `DenyEC2Production` nega esplicitamente l'accesso all'istanza `i-1234567890abcdef0`.

Attività 2: María vuole consentire a Zhang di creare tutti X-Company gli utenti, ma solo con il limite delle `XCompanyBoundaries` autorizzazioni. A tale scopo, crea questa policy gestita dal cliente, denominata `DelegatedUserBoundary`. Questa policy definisce il numero massimo di autorizzazioni di cui Zhang può disporre.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CreateOrChangeOnlyWithBoundary",
            "Effect": "Allow",
            "Action": [
```

```
                "iam:AttachUserPolicy",
                "iam:CreateUser",
                "iam:DeleteUserPolicy",
                "iam:DetachUserPolicy",
                "iam:PutUserPermissionsBoundary",
                "iam:PutUserPolicy"
            ],
            "Resource": "*",
            "Condition": {
               "StringEquals": {
                 "iam:PermissionsBoundary": "arn:aws:iam::123456789012:policy/
XCompanyBoundaries"
                }
            }
        },
        {
            "Sid": "CloudWatchAndOtherIAMTasks",
            "Effect": "Allow",
            "Action": [
              "cloudwatch:*",
              "iam:CreateAccessKey",
              "iam:CreateGroup",
              "iam:CreateLoginProfile",
              "iam:CreatePolicy",
              "iam:DeleteGroup",
              "iam:DeletePolicy",
              "iam:DeletePolicyVersion",
              "iam:DeleteUser",
              "iam:GetAccountPasswordPolicy",
              "iam:GetGroup",
              "iam:GetLoginProfile",
              "iam:GetPolicy",
              "iam:GetPolicyVersion",
              "iam:GetRolePolicy",
              "iam:GetUser",
              "iam:GetUserPolicy",
              "iam:ListAccessKeys",
              "iam:ListAttachedRolePolicies",
              "iam:ListAttachedUserPolicies",
              "iam:ListEntitiesForPolicy",
              "iam:ListGroups",
              "iam:ListGroupsForUser",
              "iam:ListMFADevices",
              "iam:ListPolicies",
```

```
                "iam:ListPolicyVersions",
                "iam:ListRolePolicies",
                "iam:ListSSHPublicKeys",
                "iam:ListServiceSpecificCredentials",
                "iam:ListSigningCertificates",
                "iam:ListUserPolicies",
                "iam:ListUsers",
                "iam:SetDefaultPolicyVersion",
                "iam:SimulateCustomPolicy",
                "iam:SimulatePrincipalPolicy",
                "iam:UpdateGroup",
                "iam:UpdateLoginProfile",
                "iam:UpdateUser"
            ],
            "NotResource": "arn:aws:iam::123456789012:user/Maria"
        },
        {
            "Sid": "NoBoundaryPolicyEdit",
            "Effect": "Deny",
            "Action": [
                "iam:CreatePolicyVersion",
                "iam:DeletePolicy",
                "iam:DeletePolicyVersion",
                "iam:SetDefaultPolicyVersion"
            ],
            "Resource": [
                "arn:aws:iam::123456789012:policy/XCompanyBoundaries",
                "arn:aws:iam::123456789012:policy/DelegatedUserBoundary"
            ]
        },
        {
            "Sid": "NoBoundaryUserDelete",
            "Effect": "Deny",
            "Action": "iam:DeleteUserPermissionsBoundary",
            "Resource": "*"
        }
    ]
}
```

Ogni istruzione svolge una funzione diversa:

1. L'istruzione `CreateOrChangeOnlyWithBoundary` consente a Zhang di creare utenti IAM ma solo se utilizza la policy `XCompanyBoundaries` per impostare il limite delle autorizzazioni. L'istruzione gli consente inoltre di impostare il limite delle autorizzazioni per gli utenti esistenti, ma solo utilizzando la stessa policy. Infine, consente a Zhang di gestire le policy di autorizzazione per gli utenti per i quali è stato impostato questo limite delle autorizzazioni.
2. L'istruzione `CloudWatchAndOtherIAMTasks` consente a Zhang di completare altre attività di gestione di utenti, gruppi e policy. Ha le autorizzazioni per reimpostare le password e creare chiavi di accesso per qualsiasi utente IAM non elencato nell'elemento della policy `NotResource`. Questo gli consente di aiutare gli utenti con problemi di accesso.
3. L'istruzione `NoBoundaryPolicyEdit` nega a Zhang l'accesso per aggiornare la policy `XCompanyBoundaries`. Zhang non può modificare alcuna policy utilizzata per impostare il limite delle autorizzazioni per sé o per altri utenti.
4. L'istruzione `NoBoundaryUserDelete` nega a Zhang l'accesso per eliminare il limite delle autorizzazioni per sé o per altri utenti.

María assegna quindi la policy `DelegatedUserBoundary` come limite delle autorizzazioni per l'utente `Zhang`.

Attività 3: poiché il limite delle autorizzazioni controlla il numero massimo di autorizzazioni, ma non concede l'accesso di per sé, María deve creare una policy di autorizzazione per Zhang. A tale scopo, crea questa policy, denominata `DelegatedUserPermissions`. Questa policy definisce le operazioni che Zhang può eseguire, entro il limite definito.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "IAM",
            "Effect": "Allow",
            "Action": "iam:*",
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchLimited",
            "Effect": "Allow",
            "Action": [
```

```
                "cloudwatch:GetDashboard",
                "cloudwatch:GetMetricData",
                "cloudwatch:ListDashboards",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3BucketContents",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::ZhangBucket"
        }
    ]
}
```

Ogni istruzione svolge una funzione diversa:

1. L'istruzione `IAM` della policy consente a Zhang l'accesso completo a IAM. Tuttavia, poiché il limite delle autorizzazioni di Zhang consente solo alcune operazioni in IAM, le sue autorizzazioni valide in IAM sono limitate solo dal relativo limite delle autorizzazioni.
2. La `CloudWatchLimited` dichiarazione consente a Zhang di eseguire cinque azioni in. CloudWatch Il limite delle sue autorizzazioni consente l'accesso a tutte le azioni CloudWatch, quindi le sue CloudWatch autorizzazioni effettive sono limitate solo dalla sua politica di autorizzazione.
3. L'istruzione `S3BucketContents` consente a Zhang di visualizzare il bucket `ZhangBucket` di Amazon S3. Tuttavia, il limite delle autorizzazioni di Zhang non gli consente alcuna operazione in Amazon S3, quindi non può eseguire operazioni S3, indipendentemente dalla sua policy di autorizzazione.

> **Note**
>
> Le policy di Zhang gli permettono di creare un utente in grado di accedere alle risorse Amazon S3 a cui lui non può accedere. Delegando queste operazioni amministrative, Maria di fatto si fida dell'accesso di Zhang ad Amazon S3.

María collega quindi la policy `DelegatedUserPermissions` come policy di autorizzazione per l'utente `Zhang`.

Attività 4: María fornisce a Zhang le istruzioni per creare un nuovo utente. Zhang può creare nuovi utenti con tutte le autorizzazioni necessarie, ma deve assegnare loro la policy `XCompanyBoundaries` come limite delle autorizzazioni.

Zhang completa le attività seguenti:

1. Zhang crea un utente con. Console di gestione AWS Digita il nome utente `Nikhil` e consente l'accesso alla console a tale utente. Cancella la casella di controllo accanto a Richiede reimpostazione della password poiché le policy sopra riportate consentono agli utenti di modificare la password solo dopo aver effettuato l'accesso alla console IAM.
2. Nella pagina Imposta le autorizzazioni, Zhang sceglie le politiche IAMFullAccess e le AmazonS3ReadOnlyAccess autorizzazioni che consentono a Nikhil di svolgere il suo lavoro.
3. Zhang salta la sezione Set permissions boundary (Imposta limite delle autorizzazioni), dimenticando le indicazioni di María.
4. Zhang esamina i dettagli utente e seleziona Create user (Crea utente).

   L'operazione ha esito negativo e l'accesso viene negato. In base al limite delle autorizzazioni di Zhang, `DelegatedUserBoundary`, qualsiasi utente da lui creato deve includere la policy `XCompanyBoundaries` come limite delle autorizzazioni.
5. Zhang torna alla pagina precedente. Nella sezione Set permissions boundary (Imposta limite delle autorizzazioni), seleziona la policy `XCompanyBoundaries`.
6. Zhang esamina i dettagli utente e seleziona Create user (Crea utente).

   L'utente viene creato.

Quando Nikhil esegue l'accesso, può accedere a IAM e Amazon S3, ma non alle operazioni rifiutate dal suo limite delle autorizzazioni. Ad esempio, può modificare la propria password in IAM ma non può creare un altro utente o modificare le policy. Nikhil ha accesso in sola lettura ad Amazon S3.

Se qualcuno aggiunge una policy basata sulle risorse al bucket `logs` che consente a Nikhil di inserire un oggetto nel bucket, significa che non può ancora accedere al bucket. Questo perché qualsiasi operazione sul bucket `logs` è esplicitamente rifiutata dal suo limite delle autorizzazioni. Un rifiuto esplicito in qualsiasi tipo di policy determina il rifiuto di una richiesta.

Tuttavia, se una policy basata sulle risorse collegata a un segreto di Secrets Manager consente a Nikhil di eseguire l'operazione `secretsmanager:GetSecretValue`, allora Nikhil potrà recuperare e decrittare il segreto. Questo perché le operazioni di Secrets Manager non sono esplicitamente rifiutate dal suo limite delle autorizzazioni e i rifiuti impliciti nei limiti delle autorizzazioni non limitano le policy basate sulle risorse.

Identity-based politiche e politiche basate sulle risorse

Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. Quando crei una policy di autorizzazione per limitare l'accesso a una risorsa, puoi scegliere una policy basata su identità o una policy basata su risorse.

Identity-based le policy sono collegate a un utente, gruppo o ruolo IAM. Queste policy consentono di specificare cosa può fare quell'identità (le sue autorizzazioni). Ad esempio, puoi collegare la policy all'utente IAM di nome John, dichiarando che a questo utente è autorizzato ad eseguire l'operazione `RunInstances` di Amazon EC2. La policy potrebbe inoltre dichiarare che a John è consentito ottenere oggetti da una tabella di Amazon DynamoDB denominata `MyCompany`. Puoi anche consentire a John di gestire le proprie credenziali di sicurezza IAM. Identity-based le politiche possono essere gestite o in linea.

Resource-based le politiche sono allegate a una risorsa. Ad esempio, puoi collegare policy basate sulle risorse a bucket Amazon S3, code Amazon SQS, endpoint VPC, chiavi di crittografia AWS Key Management Service , tabelle e flussi Amazon DynamoDB e risorse. AWS Sign-In Per un elenco dei servizi che supportano le policy basate su risorse, consulta AWS servizi supportati da IAM.

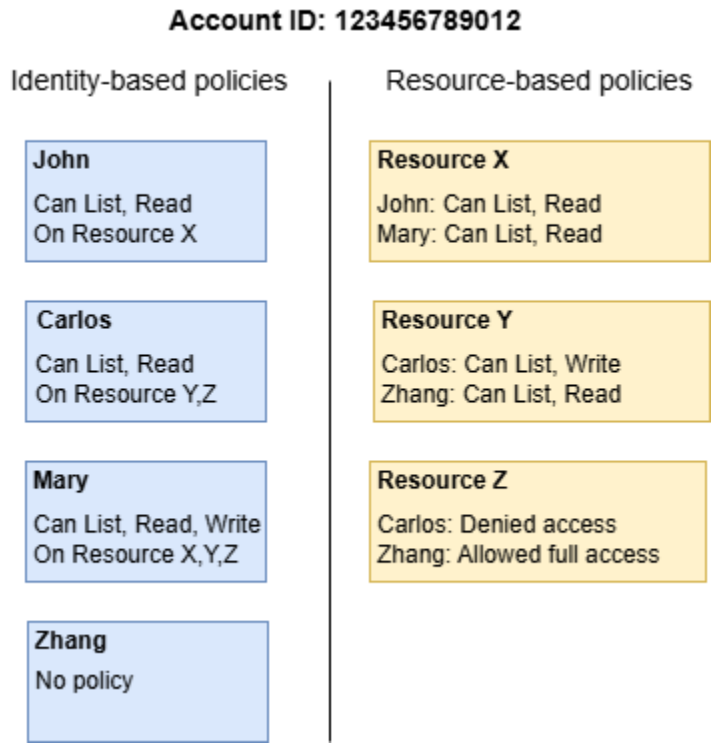
Le policy basate su risorse consentono di specificare quali utenti hanno accesso a una risorsa e quali operazioni possono eseguirvi. Per sapere se i responsabili di account esterni alla tua zona di fiducia (organizzazione o account attendibile) possono assumere i tuoi ruoli, consulta Cos'è IAM Access Analyzer? . Resource-based le politiche sono solo in linea, non gestite.

> **Note**
>
> Resource-basedle politiche differiscono dalle autorizzazioni a livello di risorsa. È possibile allegare politiche basate sulle risorse direttamente a una risorsa, come descritto in questo argomento. Resource-level le autorizzazioni si riferiscono alla possibilità di utilizzare gli ARN per specificare singole risorse in una policy. Resource-based le politiche sono supportate solo da alcuni AWS servizi. Per un elenco dei servizi che supportano delle policy basate su risorse e le autorizzazioni a livello di risorsa, consultare AWS servizi supportati da IAM.

Per informazioni su come interagiscono le policy basate su identità e le policy basate sulle risorse all'interno dello stesso account, consulta Valutazione delle policy per le richieste all'interno di un singolo account.

Per informazioni su come le policy interagiscono tra gli account, consulta Cross-account logica di valutazione delle politiche.

Per comprendere meglio questi concetti, visualizza la figura riportata di seguito. L'amministratore dell'account `123456789012` ha collegato policy basate su identità agli utenti `John`, `Carlos` e `Mary`. Alcune delle operazioni in queste policy possono essere eseguite su risorse specifiche. Ad esempio, l'utente `John` può eseguire alcune operazioni su `Resource X`. Si tratta di un'autorizzazione a livello di risorsa in una policy basata su identità. L'amministratore ha inoltre aggiunto politiche basate sulle risorse a`Resource X`, `Resource Y` e. `Resource Z` Resource-based le politiche consentono di specificare chi può accedere a quella risorsa. Ad esempio, la policy basata su risorsa su `Resource X` consente agli utenti `John` e `Mary` l'accesso all'elenco e in lettura a quella risorsa.



L'esempio di account `123456789012` consente agli utenti seguenti di eseguire le operazioni elencate:

- John: John può eseguire operazioni di lettura ed elenco su `Resource X`. Gli è concessa questa autorizzazione dalla policy basata su identità sul suo utente e dalla policy basata su risorsa su `Resource X`.

- Carlos: Carlos può eseguire operazioni di lettura, scrittura ed elenco su `Resource Y` ma gli viene rifiutato l'accesso a `Resource Z`. La policy basata su identità su Carlos gli consente di eseguire operazioni di lettura ed elenco su `Resource Y`. La policy basata sulla risorsa `Resource Y`, inoltre, gli concede autorizzazioni di scrittura. Tuttavia, anche se la sua policy basata su identità gli consente l'accesso a `Resource Z`, la policy basata sulla risorsa `Resource Z` nega tale accesso. Un `Deny` esplicito sostituisce un `Allow` e il suo accesso a `Resource Z` viene negato. Per ulteriori informazioni, consulta Logica di valutazione delle policy.
- Mary: Mary può eseguire operazioni di scrittura, lettura ed elenco su `Resource X`, `Resource Y` e `Resource Z`. La sua policy basata su identità le consente più operazioni su più risorse rispetto alle policy basate su risorse, ma nessuna di queste nega l'accesso.
- Zhang: Zhang ha accesso completo a `Resource Z`. Zhang non ha policy basate su identità, ma la policy basata sulla risorsa `Resource Z` gli consente l'accesso completo alla risorsa. Zhang può anche eseguire elenco e leggere operazioni su `Resource Y`.

Identity-based le politiche e le politiche basate sulle risorse sono entrambe politiche di autorizzazione e vengono valutate insieme. Per una richiesta a cui si applicano solo i criteri di autorizzazione, verifica AWS innanzitutto che tutti i criteri abbiano un. `Deny` Se ne esiste una, la richiesta viene negata. Quindi, AWS verifica ogni `Allow`. Se almeno un'istruzione della policy consente l'operazione nella richiesta, la richiesta è consentita. Non importa se l'`Allow` è concessa dalla policy basata su identità o dalla policy basata su risorse.

> **Important**
>
> Questa logica si applica solo quando la richiesta viene effettuata all'interno di un singolo Account AWS. Per le richieste effettuate da un account a un altro, il richiedente nell'`Account A` deve disporre di una policy basata su identità che gli consenta di effettuare una richiesta alla risorsa nell'`Account B`. Inoltre, la policy basata sulla risorsa nell'`Account B` deve consentire al richiedente nell'`Account A` di accedere alla risorsa. In entrambi gli account devono essere presenti policy che consentono l'operazione, altrimenti la richiesta non riesce. Per ulteriori informazioni sull'utilizzo delle policy basate sulle risorse per l'accesso tra account, consulta Accesso alle risorse multi-account in IAM.

Un utente che dispone di autorizzazioni specifiche potrebbe richiedere una risorsa a cui è collegata anche una policy di autorizzazione. In tal caso, AWS valuta entrambi i set di autorizzazioni per

determinare se concedere l'accesso alla risorsa. Per ulteriori informazioni su come vengono valutate le policy, consultare Logica di valutazione delle policy.

> **Note**
>
> Amazon S3 supporta le policy basate sulle identità e le policy basate su risorse (dette policy dei bucket). Inoltre, Amazon S3 supporta un meccanismo di autorizzazione noto come lista di controllo accessi (ACL) che è indipendente dalle policy e dalle autorizzazioni IAM. È possibile usare le policy IAM in combinazione con le liste di controllo accessi di Amazon S3. Per ulteriori informazioni, consulta Controllo accessi nella Guida per l'utente di Amazon Simple Storage Service.

Controlla l'accesso alle AWS risorse utilizzando le policy

Puoi utilizzare una policy per controllare l'accesso alle risorse all'interno di IAM o in tutto AWS.

Per utilizzare una policy per controllare l'accesso in AWS, è necessario comprendere in che modo AWS concede l'accesso. AWS è composto da raccolte di risorse. Un utente IAM è una risorsa. Un bucket Amazon S3 è una risorsa. Quando si utilizza l' AWS API AWS CLI, il o il Console di gestione AWS per eseguire un'operazione (come la creazione di un utente), si invia una richiesta per tale operazione. La richiesta specifica un'operazione, una risorsa, un'entità principale (utente o ruolo), un account principale e qualsiasi altra informazione necessaria per la richiesta. Tutte queste informazioni forniscono il contesto.

AWS verifica quindi che tu (il principale) sia autenticato (effettuato l'accesso) e autorizzato (disponi del permesso) a eseguire l'azione specificata sulla risorsa specificata. Durante l'autorizzazione, AWS controlla tutte le politiche che si applicano al contesto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON e specifica le autorizzazioni per le entità principali. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Politiche e autorizzazioni in AWS Identity and Access Management.

AWS autorizza la richiesta solo se ogni parte della richiesta è consentita dalle politiche. Per visualizzare un diagramma di questo processo, consultare Funzionamento di IAM. Per informazioni dettagliate su come AWS determinare se una richiesta è consentita, consultaLogica di valutazione delle policy.

Quando crei una policy IAM, puoi controllare l'accesso ai seguenti elementi:

- Principali: controlla quello che la persona che effettua la richiesta (principale) è autorizzata a fare.
- Identità IAM: controlla a quali identità IAM (gruppi IA;, utenti e ruoli) è possibile accedere e come.
- Policy IAM: controlla quali utenti possono creare, modificare ed eliminare le policy gestite dai clienti e quali utenti possono collegare e scollegare tutte le policy gestite.
- Risorse AWS: consente di controllare quali utenti hanno accesso alle risorse tramite una policy basata sulle identità o una policy basata sulle risorse.
- Account AWS: consente di controllare se una richiesta è consentita solo per i membri di un determinato account.

Le politiche consentono di specificare chi ha accesso alle AWS risorse e quali azioni può eseguire su tali risorse. Inizialmente, nessun utente IAM dispone di autorizzazioni. Ovvero, per impostazione predefinita, gli utenti non possono eseguire alcuna operazione, neppure visualizzare le proprie chiavi di accesso. Per fornire a un utente l'autorizzazione per eseguire un'operazione, è possibile aggiungere l'autorizzazione all'utente, ovvero collegare una policy all'utente. In alternativa, puoi aggiungere l'utente a un gruppo di utenti con l'autorizzazione desiderata.

Ad esempio, è possibile concedere a un utente l'autorizzazione per l'elencazione delle proprie chiavi di accesso. È inoltre possibile espandere tale autorizzazione e consentire inoltre a ciascun utente di creare, aggiornare ed eliminare le proprie chiavi.

Quando concedi le autorizzazioni a un gruppo di utenti, tutti gli utenti in quel gruppo potranno usufruire di tali autorizzazioni. Ad esempio, puoi concedere al gruppo di utenti Administrators l'autorizzazione a eseguire qualsiasi azione IAM su qualsiasi Account AWS risorsa. Un altro esempio: è possibile fornire al gruppo di utenti Manager l'autorizzazione per descrivere le istanze Amazon EC2 dell' Account AWS.

Per informazioni su come delegare le autorizzazioni di base per utenti, gruppi IAM e ruoli, consulta Autorizzazioni necessarie per accedere alle risorse IAM. Per ulteriori esempi di policy che utilizzano queste autorizzazioni, consultare Esempi di policy per amministrare le risorse IAM.

Controllo dell'accesso per le entità principali

È possibile usare le policy per controllare le operazioni che la persona da cui proviene la richiesta (entità principale) è autorizzata a effettuare. A tale scopo, è necessario collegare una policy basata su identità all'identità di questa persona (utente, gruppo di utenti o ruolo). È possibile utilizzare anche un limite di autorizzazioni per impostare il numero massimo di autorizzazioni che un'entità (utente o ruolo) può avere.

Ad esempio, supponiamo di volere che l'utente Zhang Wei abbia accesso completo ad Amazon DynamoDB CloudWatch, Amazon EC2 e Amazon S3. È possibile creare due policy diverse, in modo che successivamente sia possibile suddividerle nel caso sia necessario un set di autorizzazioni per un utente diverso. In alternativa, è possibile includere entrambe le autorizzazioni in una singola policy e quindi collegare tale policy all'utente IAM denominato Zhang Wei. È anche possibile collegare una policy a un gruppo di utenti a cui Zhang appartiene o a un ruolo che Zhang può assumere. Di conseguenza, quando Zhang visualizza i contenuti di un bucket di S3, le sue richieste vengono accettate. Se prova a creare un nuovo utente IAM, la richiesta viene rifiutata perché non dispone dell'autorizzazione.

È possibile utilizzare un limite delle autorizzazioni su Zhang per fare in modo che non gli venga mai dato accesso al bucket `amzn-s3-demo-bucket1` di S3. A tale scopo, è necessario determinare il numero massimo di autorizzazioni per Zhang. In questo caso è possibile controllare le sue attività con le policy di autorizzazione. L'importante è che non possa accedere al bucket riservato. Quindi usi la seguente policy per definire il limite di Zhang per consentire tutte le AWS azioni per Amazon S3 e alcuni altri servizi ma negare l'accesso al bucket S3. `amzn-s3-demo-bucket1` Poiché il limite delle autorizzazioni non consente alcuna operazione IAM, impedisce a Zhang di eliminare il proprio limite o quello di altri utenti.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PermissionsBoundarySomeServices",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:*",
                "dynamodb:*",
                "ec2:*",
                "s3:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PermissionsBoundaryNoConfidentialBucket",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": [
```

```
                "arn:aws:s3:::amzn-s3-demo-bucket1",
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ]
        }
    ]
}
```

Quando si assegna una policy come questa come limite delle autorizzazioni per un utente, la policy non concede alcuna autorizzazione. Imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM. Per ulteriori informazioni sui limiti delle autorizzazioni, consultare Limiti delle autorizzazioni per le entità IAM.

Per informazioni dettagliate sulle procedure precedenti, è possibile consultare le risorse seguenti:

- Per ulteriori informazioni sulla creazione di una policy IAM che è possibile collegare a un principale, consulta Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente.
- Per ulteriori informazioni su come collegare una policy IAM a un principale, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.
- Per consultare un esempio di policy per concedere accesso completo a EC2, consultare Amazon EC2: consente l'accesso completo a EC2 entro una regione specifica, a livello di programmazione e nella console.
- Per permettere l'accesso in sola lettura a un bucket S3, utilizzare le prime due istruzioni della seguente policy di esempio: Amazon S3: consente l'accesso in lettura e scrittura agli oggetti in un bucket S3, in modo programmatico e nella console.
- Per visualizzare una policy di esempio che consente agli utenti di impostare le credenziali, ad esempio la password della console, le chiavi di accesso a livello di programmazione e i dispositivi MFA, consulta la pagina AWS: consente agli utenti IAM autenticati con MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Controllo dell'accesso alle identità

Puoi usare le policy IAM per controllare le operazioni che tutti gli utenti possono eseguire per un'identità mediante la creazione di una policy da collegare a tutti gli utenti tramite un gruppo di utenti. Per eseguire questa operazione, è necessario creare una policy che limita le operazioni che possono essere eseguite per un'identità oppure gli utenti che possono accedere.

Ad esempio, puoi creare un gruppo di utenti denominato AllUserse quindi associare quel gruppo di utenti a tutti gli utenti. Quando si crea il gruppo di utenti, è possibile fornire a tutti gli utenti l'accesso per impostare le proprie credenziali come descritto nella sezione precedente. È quindi possibile creare una policy che rifiuti l'accesso alla modifica del gruppo di utenti a meno che il nome utente non sia incluso nella condizione della policy. Tuttavia, la parte della policy rifiuta solo l'accesso a chiunque eccetto gli utenti elencati. È inoltre necessario includere le autorizzazioni per permettere tutte le operazioni di gestione del gruppo di utenti per tutti gli utenti del gruppo. Infine, puoi collegare questa policy al gruppo di utenti in modo che venga applicata a tutti gli utenti. Di conseguenza, quando un utente non specificato nella policy cerca di apportare modifiche al gruppo di utenti, la richiesta viene rifiutata.

Per creare questa policy con l'editor visivo

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.
3. Scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione Visivo.
5. In Seleziona un servizio, scegli IAM.
6. In Operazioni consentite, digita **`group`** nella casella di ricerca. L'editor visivo mostra tutte le operazioni IAM che contengono la parola `group`. Selezionare tutte le caselle di controllo.
7. Selezionare Resources (Risorse) per specificare le risorse per la policy. In base alle operazioni scelte, dovrebbero venire visualizzati i tipi di risorse gruppo e utente.

   - gruppo: scegli Aggiungi ARNs. Per Risorse in, seleziona l'opzione Qualsiasi account. Seleziona la casella di controllo Qualsiasi nome di gruppo con percorso, quindi digita il nome del gruppo di utenti **`AllUsers`**. Quindi scegliere Add (Aggiungi) ARNs.
   - utente: seleziona la casella di controllo accanto a Qualsiasi in questo account.

   Una delle operazioni scelte, `ListGroups`, non supporta l'utilizzo di risorse specifiche. Non è necessario selezionare All resources (Tutte le risorse) per tale operazione. Quando salvi la policy o la visualizzi nell'editor JSON, puoi notare che IAM crea automaticamente un nuovo blocco di autorizzazioni che concede l'autorizzazione per questa operazione a tutte le risorse.

8. Per aggiungere un altro blocco di autorizzazioni, scegli Aggiungi altre autorizzazioni.
9. Scegli Seleziona un servizio e quindi IAM.
10. Scegli Operazioni consentite, quindi seleziona Passa ad autorizzazioni rifiutate. In questo caso, l'intero blocco viene utilizzato per rifiutare le autorizzazioni.
11. Nella casella di ricerca digitare **`group`**. L'editor visivo mostra tutte le operazioni IAM che contengono la parola `group`. Selezionare le caselle di controllo accanto alle seguenti operazioni:
    - CreateGroup
    - DeleteGroup
    - RemoveUserFromGroup
    - AttachGroupPolicy
    - DeleteGroupPolicy
    - DetachGroupPolicy
    - PutGroupPolicy
    - UpdateGroup
12. Selezionare Resources (Risorse) per specificare le risorse per la policy. In base alle operazioni scelte, dovrebbe venire visualizzato il tipo di risorse group (gruppo). Scegliere Aggiungi ARNs. Per Risorse in, seleziona l'opzione Qualsiasi account. Per Qualsiasi nome gruppo con percorso, digita il nome del gruppo di utenti **`AllUsers`**. Quindi scegliere Add (Aggiungi) ARNs.
13. Scegli Condizioni di richiesta - opzionale, quindi scegli Aggiungi altra condizione. Completare il modulo con i seguenti valori:
    - Chiave di condizione: scegli aws:username
    - Qualificatore: scegli Default
    - Operatore: scegli StringNotEquals
    - Valore: digita **`srodriguez`**, quindi scegli Aggiungi per aggiungere un altro valore. Digita **`mjackson`**, quindi scegli Aggiungi per aggiungere un altro valore. Digita **`adesai`** e quindi seleziona Aggiungi un altro valore di condizione.

    Questa condizione garantisce che l'accesso venga rifiutato per le operazioni di gestione del gruppo di utenti specificate se l'utente che effettua la chiamata non è incluso nell'elenco. Poiché l'autorizzazione viene rifiutata in modo esplicito, il blocco precedente che consentiva a tali utenti di chiamare le operazioni viene ignorato. Per gli utenti inclusi nell'elenco, l'accesso non viene

rifiutato e viene concessa dell'autorizzazione del primo blocco di autorizzazioni, in modo che possano gestire completamente il gruppo.

14. Quando hai terminato, seleziona Successivo.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'opzione dell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

15. Nella pagina Verifica e crea, digita **`LimitAllUserGroupManagement`** in Nome della policy. In Description (Descrizione), digitare **`Allows all users read-only access to a specific user group, and allows only specific users access to make changes to the user group`**. Rivedi il campo Autorizzazioni definite in questa policy per accertarti di disporre delle autorizzazioni previste. Quindi selezionare Create policy (Crea policy) per salvare la nuova policy.
16. Collega la policy al tuo gruppo di utenti. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

In alternativa, è possibile creare la stessa policy utilizzando questo esempio di documento di policy JSON. Per visualizzare questa policy JSON, consulta IAM: consente a utenti IAM specifici di gestire un gruppo a livello di programmazione e nella console. Per istruzioni dettagliate sulla creazione di una policy utilizzando un documento JSON, consulta the section called "Creazione di policy utilizzando l'editor JSON".

Controllo dell'accesso alle policy

Puoi controllare in che modo gli utenti possono applicare le policy AWS gestite. Per eseguire questa operazione, collegare questa policy per tutti gli utenti. In teoria, è possibile eseguire questa operazione utilizzando un gruppo di utenti.

Ad esempio, potresti creare una policy che consenta agli utenti di allegare solo IAMUserChangePasswordle policy PowerUserAccess AWS gestite a un nuovo utente, gruppo di utenti o ruolo IAM.

Per le policy gestite dal cliente, è possibile controllare chi può creare, aggiornare ed eliminare queste policy. È possibile controllare chi può collegare e scollegare le policy per entità principali (gruppi IAM,

utenti e ruoli). È inoltre possibile controllare quali policy un utente può collegare o distaccare e per quale entità.

Ad esempio, è possibile concedere autorizzazioni a un account amministratore per creare, aggiornare ed eliminare le policy. Quindi è possibile assegnare le autorizzazioni a un team leader o a un altro amministratore limitato collegare o distaccare queste policy a/da entità principali che l'amministratore limitato gestisce.

Per ulteriori informazioni, fare riferimento a queste risorse:

- Per ulteriori informazioni sulla creazione di una policy IAM che è possibile collegare a un principale, consulta Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente.
- Per ulteriori informazioni su come collegare una policy IAM a un principale, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.
- Per consultare un esempio di policy per limitare l'uso di policy gestito, consultare IAM: limita le policy gestite che possono essere applicate a un utente, un gruppo o un ruolo IAM..

Controllo delle autorizzazioni per la creazione, l'aggiornamento e l'eliminazione di policy gestite dal cliente

Puoi usare le policy IAM per controllare chi può creare, aggiornare ed eliminare le policy gestite dal cliente nell' Account AWS. L'elenco seguente contiene le operazioni di API che si riferiscono direttamente a creazione, aggiornamento ed eliminazione di policy o versioni di policy:

- CreatePolicy
- CreatePolicyVersion
- DeletePolicy
- DeletePolicyVersion
- SetDefaultPolicyVersion

Le operazioni API nell'elenco precedente corrispondono alle operazioni che è possibile consentire o rifiutare, ovvero le autorizzazioni che è possibile concedere, utilizzando una policy IAM.

Esaminiamo l'esempio di policy seguente. Permette a un utente di creare, aggiornare (ovvero creare una nuova versione della policy), eliminare e impostare una versione predefinita per tutte le policy gestite dal cliente nell' Account AWS. La policy di esempio, inoltre, consente all'utente di elencare e ottenere le policy. Per ulteriori informazioni su come creare una policy utilizzando questo esempio

di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

Example Esempio di policy che permette la creazione, l'aggiornamento, l'eliminazione, la visualizzazione, l'ottenimento e la configurazione della versione di default per tutte le policy

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "iam:CreatePolicy",
      "iam:CreatePolicyVersion",
      "iam:DeletePolicy",
      "iam:DeletePolicyVersion",
      "iam:GetPolicy",
      "iam:GetPolicyVersion",
      "iam:ListPolicies",
      "iam:ListPolicyVersions",
      "iam:SetDefaultPolicyVersion"
    ],
    "Resource": "*"
  }
}
```

È possibile creare policy che limitano l'uso di queste operazioni delle API che interessano solo le policy gestite specificate dall'utente. Ad esempio, è possibile permettere a un utente di impostare la versione predefinita ed eliminare le versioni di policy, ma solo per determinate policy gestite dal cliente. A tale scopo, è necessario specificare l'ARN della policy nell'elemento `Resource` della policy che concede l'autorizzazione.

L'esempio seguente mostra una policy che consente a un utente di eliminare versioni della policy e impostare la versione predefinita. Tuttavia, queste azioni sono consentite solo per le policy gestite dal cliente che includono il percorso /TEAM-A/. L'ARN della policy gestita dal cliente è specificato nell'elemento `Resource` della policy. In questo esempio l'ARN include un percorso e un carattere jolly e quindi corrisponde a tutte le policy gestite dal cliente che includono il percorso /TEAM-A/. Per

ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

Per ulteriori informazioni sull'utilizzo di percorsi di clienti nei nomi delle policy gestite dal cliente, consultare Nomi descrittivi e percorsi.

Example Esempio di policy che consente di eliminare le versioni di policy e impostare la versione di default solo per policy specifiche

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:DeletePolicyVersion",
            "iam:SetDefaultPolicyVersion"
        ],
        "Resource": "arn:aws:iam::111122223333:policy/TEAM-A/*"
    }
}
```

Controllo delle autorizzazioni per collegare e distaccare le policy gestite

È possibile utilizzare le policy IAM anche per permettere agli utenti di utilizzare policy gestite specifiche. In pratica, è possibile controllare le autorizzazioni che un utente può concedere ad altre entità principali.

L'elenco seguente mostra le operazioni di API che consentono direttamente di collegare e distaccare le policy gestite a/da entità principali:

- AttachGroupPolicy
- AttachRolePolicy
- AttachUserPolicy
- DetachGroupPolicy
- DetachRolePolicy
- DetachUserPolicy

È possibile creare policy che limitino l'uso di queste operazioni API in modo da influire solo sulle entità and/or principali delle politiche gestite specifiche specificate. Ad esempio, è possibile permettere a un utente di collegare le policy gestite, ma per le policy specificate dall'utente. Oppure, è possibile permettere a un utente di collegare le policy gestite, ma alle entità del principale specificate dall'utente.

L'esempio di policy seguente consente a un utente di collegare le policy gestite solo ai gruppi IAM e ai ruoli che includono il percorso /TEAM-A/. Il gruppo di utenti e il ruolo ARNs sono specificati nell'`Resource`elemento della politica. (In questo esempio ARNs includono un percorso e un carattere jolly e quindi corrispondono a tutti i gruppi e i ruoli IAM che includono il percorso /TEAM-A/). Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

Example Esempio di policy che consente di collegare policy gestite solo a gruppi di utenti o ruoli specifici

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:AttachGroupPolicy",
            "iam:AttachRolePolicy"
        ],
        "Resource": [
            "arn:aws:iam::111122223333:group/TEAM-A/*",
            "arn:aws:iam::111122223333:role/TEAM-A/*"
        ]
    }
}
```

È possibile limitare ulteriormente le operazioni dell'esempio precedente per influire solo su determinate policy. In altre parole, puoi controllare le autorizzazioni che un utente può collegare ad altre entità principali, aggiungendo una condizione alla policy.

In questo esempio, la condizione garantisce che le autorizzazioni `AttachGroupPolicy` e `AttachRolePolicy` siano consentite solo quando la policy da collegare corrisponde a uno delle

policy specificate. La condizione utilizza la chiave della condizione `iam:PolicyARN` per determinare quali policy possono essere collegate. L'esempio di policy seguente amplia il concetto espresso nell'esempio precedente. Consente a un utente di collegare solo le policy gestite che includono il percorso /TEAM-A/ solo ai gruppi IAM e ai ruoli che includono il percorso /TEAM-A/. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called "Creazione di policy utilizzando l'editor JSON".

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:AttachGroupPolicy",
            "iam:AttachRolePolicy"
        ],
        "Resource": [
            "arn:aws:iam::111122223333:group/TEAM-A/*",
            "arn:aws:iam::111122223333:role/TEAM-A/*"
        ],
        "Condition": {
            "ArnLike": {
                "iam:PolicyARN": "arn:aws:iam::111122223333:policy/TEAM-A/*"
            }
        }
    }
}
```

Questa policy utilizza l'operatore di condizione `ArnLike` ma puoi anche utilizzare l'operatore di condizione `ArnEquals` perché questi due operatori si comportano in modo identico. Per ulteriori informazioni su `ArnLike` e `ArnEquals`, consulta Operatori di condizione con Amazon Resource Name (ARN) nella sezione Tipi di condizione dei Riferimenti agli elementi della policy.

Ad esempio, è possibile limitare l'uso di operazioni per coinvolgere solo le policy gestite specificate dall'utente. A tale scopo, è necessario specificare l'ARN della policy nell'elemento `Condition` della policy che concede l'autorizzazione. Ad esempio, per specificare l'ARN di una policy gestita dal cliente:

```
"Condition": {"ArnEquals":
  {"iam:PolicyARN": "arn:aws:iam::123456789012:policy/POLICY-NAME"}
}
```

È inoltre possibile specificare l'ARN di una policy AWS gestita nell'elemento di `Condition` una policy. L'ARN di una policy AWS gestita utilizza l'alias speciale `aws` nella policy ARN anziché un ID account, come in questo esempio:

```
"Condition": {"ArnEquals":
  {"iam:PolicyARN": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"}
}
```

Controllo dell'accesso alle risorse

È possibile controllare chi ha accesso alle risorse utilizzando una policy basata sulle identità o una policy basata sulle risorse. In una policy basata sulle identità si collega la policy a un'identità e si specifica a quali risorse può accedere tale identità. In una policy basata sulle risorse, si collega una policy alla risorsa che si desidera controllare. Nella policy, è necessario specificare quali entità principali possono accedere a tale risorsa. Per ulteriori informazioni su entrambi questi tipi di policy, consultare Identity-based politiche e politiche basate sulle risorse.

Per ulteriori informazioni, fare riferimento a queste risorse:

- Per ulteriori informazioni sulla creazione di una policy IAM che è possibile collegare a un principale, consulta Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente.
- Per ulteriori informazioni su come collegare una policy IAM a un principale, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.
- Amazon S3 supporta l'utilizzo delle policy basate su risorse nei relativi bucket. Per ulteriori informazioni, consulta Esempi di policy di bucket.

Gli autori delle risorse non dispongono automaticamente di autorizzazioni

Se accedi utilizzando le Utente root dell'account AWS credenziali, sei autorizzato a eseguire qualsiasi azione sulle risorse che appartengono all'account. Tuttavia, ciò non è valido per gli utenti IAM. Un utente IAM potrebbe disporre dell'accesso per creare una risorsa, ma le autorizzazioni dell'utente, anche per tale risorsa, sono limitate a quanto è stato concesso esplicitamente. Ciò significa che la semplice creazione di una risorsa, ad esempio di un ruolo IAM, non garantisce automaticamente l'autorizzazione per la modifica o l'eliminazione di tale ruolo. Inoltre, l'autorizzazione può essere

revocata in qualsiasi momento dal proprietario dell'account o da un altro utente che dispone dell'accesso per gestire le autorizzazioni.

Controllo dell'accesso ai principali in un account specifico

È possibile concedere direttamente agli utenti IAM dell'account l'accesso alle risorse. Se gli utenti di un altro account devono accedere alle risorse, è possibile creare un ruolo IAM. Un ruolo è un'entità che include autorizzazioni, ma non è associata a un utente specifico. Gli utenti di altri account possono assumere quel ruolo e accedere alle risorse in base alle autorizzazioni assegnate al ruolo. Per ulteriori informazioni, consulta Accesso per un utente IAM in un altro Account AWS che possiedi.

> **Note**
>
> Alcuni servizi supportano le policy basate sulle risorse, come descritto in Identity-based politiche e politiche basate sulle risorse (ad esempio Amazon S3, Amazon SNS e Amazon SQS). Per tali servizi, un'alternativa all'utilizzo dei ruoli consiste nel collegare una policy alla risorsa (bucket, argomento o coda) che si desidera condividere. La politica basata sulle risorse può specificare l' AWS account che dispone delle autorizzazioni per accedere alla risorsa.

Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag

Utilizza le informazioni nella sezione seguente per controllare chi può accedere agli utenti e ai ruoli IAM e a quali risorse gli utenti e i ruoli possono accedere. Per informazioni più generali ed esempi di controllo dell'accesso ad altre AWS risorse, incluse altre risorse IAM, consultaTag per AWS Identity and Access Management risorse.

> **Note**
>
> Per dettagli sulla distinzione tra maiuscole e minuscole per le chiavi dei tag e i valori delle chiavi dei tag, consulta Case sensitivity.

I tag possono essere collegati alla risorsa IAM, trasferiti nella richiesta o collegati al principale che effettua la richiesta. Un utente o ruolo IAM può essere sia una risorsa che un principale. Ad esempio, puoi scrivere una policy che consente a un utente di elencare i gruppi per un utente. Questa operazione è consentita solo se l'utente che effettua la richiesta (principale) ha lo stesso

tag `project=blue` dell'utente che sta tentando di visualizzare. In questo esempio, l'utente può visualizzare l'appartenenza al gruppo per qualsiasi utente, incluso se stesso, purché stia lavorando sullo stesso progetto.

Per controllare gli accessi in base ai tag, devi fornire informazioni sui tag nell'elemento condizione di una policy. Quando crei una policy IAM, puoi utilizzare i tag IAM e la chiave di condizione tag associata per controllare l'accesso a quanto segue:

- Risorsa: controlla l'accesso alle risorse di utente o ruolo in base ai relativi tag. Per fare ciò, usa la chiave aws:ResourceTag/***key-name***condition per specificare quale coppia chiave-valore di tag deve essere associata alla risorsa. Per ulteriori informazioni, consulta Controllo dell'accesso alle AWS risorse.
- Richiesta: controlla quali tag possono essere passati in una richiesta IAM. A tale scopo, usa la chiave aws:RequestTag/***key-name***condition per specificare quali tag possono essere aggiunti, modificati o rimossi da un utente o ruolo IAM. Questa chiave viene utilizzata allo stesso modo per le risorse IAM e altre AWS risorse. Per ulteriori informazioni, consulta Controllo dell'accesso durante AWS le richieste.
- Principale: controlla le operazioni consentite alla persona che effettua la richiesta (il principale) in base ai tag collegati all'utente o al ruolo IAM di tale persona. Per fare ciò, usa la chiave aws:PrincipalTag/***key-name***condition per specificare quali tag devono essere allegati all'utente o al ruolo IAM prima che la richiesta sia consentita.
- Qualsiasi parte del processo di autorizzazione: utilizza la chiave aws: TagKeys condition per controllare se chiavi di tag specifiche possono essere utilizzate in una richiesta o da un principale. In questo caso, il valore chiave non è importante. Questa chiave si comporta in modo simile per IAM e altri AWS servizi. Tuttavia, quando aggiungi tag a un utente in IAM, questo controlla anche se il principale può effettuare la richiesta a qualsiasi servizio. Per ulteriori informazioni, consulta Controllo dell'accesso in base alle chiavi di tag.

È possibile creare una policy IAM utilizzando l'editor visivo, tramite JSON o importando una policy gestita esistente. Per informazioni dettagliate, vedi Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente.

> **Note**
>
> Puoi anche passare tag di sessione quando assumi un ruolo IAM o esegui la federazione di un utente. Questi tag sono validi solo per la durata della sessione.

Controllo dell'accesso per i principali IAM

È possibile controllare le operazioni che il principale è autorizzato a eseguire in base ai tag collegati all'identità di tale persona.

Questo esempio mostra come creare una policy basata sull'identità che consenta a qualsiasi utente in questo account di visualizzare l'appartenenza al gruppo per qualsiasi utente, incluso sé stesso, purché stia lavorando sullo stesso progetto. Questa operazione è consentita solo quando il tag della risorsa dell'utente e il tag del principale hanno lo stesso valore per la chiave del tag `project`. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "iam:ListGroupsForUser",
            "Resource": "arn:aws:iam::111222333444:user/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/project":
 "${aws:PrincipalTag/project}"}
            }
        }]
}
```

Controllo dell'accesso in base alle chiavi di tag

Puoi utilizzare i tag nelle policy IAM per controllare se chiavi di tag specifiche possono essere utilizzate in una richiesta o da un principale.

Questo esempio mostra come creare una policy basata sull'identità che consenta di rimuovere solo il tag con la chiave `temporary` da parte degli utenti. Per utilizzare questa politica, sostituisci la *`italicized placeholder text`* politica dell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "iam:UntagUser",
        "Resource": "*",
        "Condition": {"ForAllValues:StringEquals": {"aws:TagKeys":
 ["temporary"]}}
    }]
}
```

Controllo dell'accesso alle AWS risorse tramite tag

Puoi utilizzare i tag per controllare l'accesso alle tue AWS risorse che supportano l'etichettatura, incluse le risorse IAM. Puoi aggiungere i tag a utenti e ruoli IAM per controllare a cosa possono accedere. Per ulteriori informazioni su come applicare tag a utenti e ruoli IAM, consulta Tag per AWS Identity and Access Management risorse. Inoltre, puoi controllare l'accesso alle seguenti risorse IAM: policy gestite dal cliente, provider di identità IAM, profili delle istanze, certificati server e dispositivi MFA virtuali. Per visualizzare un tutorial per la creazione e il test di una policy che consente ai ruoli IAM con tag principali di accedere alle risorse con tag corrispondenti, consulta Tutorial IAM: Definizione delle autorizzazioni di accesso AWS risorse basate sui tag. Utilizza le informazioni contenute nella sezione seguente per controllare l'accesso ad altre AWS risorse, incluse le risorse IAM, senza etichettare gli utenti o i ruoli IAM.

Prima di utilizzare i tag per controllare l'accesso alle tue AWS risorse, devi capire come AWS concedere l'accesso. AWS è composto da raccolte di risorse. Una istanza Amazon EC2 è una risorsa. Un bucket Amazon S3 è una risorsa. Puoi utilizzare l' AWS API AWS CLI, the o the Console di gestione AWS per eseguire un'operazione, come la creazione di un bucket in Amazon S3. In questo caso, invii una richiesta per tale operazione. La richiesta specifica un'operazione, una risorsa, un'entità principale (utente o ruolo), un account principale e qualsiasi altra informazione necessaria per la richiesta. Tutte queste informazioni forniscono il contesto.

AWS verifica quindi che tu (l'entità principale) sia autenticato (effettuato l'accesso) e autorizzato (disponi del permesso) a eseguire l'azione specificata sulla risorsa specificata. Durante l'autorizzazione, AWS controlla tutte le politiche che si applicano al contesto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON e specifica le

autorizzazioni per le entità principali. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Politiche e autorizzazioni in AWS Identity and Access Management.

AWS autorizza la richiesta solo se ogni parte della richiesta è consentita dalle politiche. Per visualizzare un diagramma e per ulteriori informazioni sull'infrastruttura IAM, consulta Funzionamento di IAM. Per ulteriori informazioni su come IAM determina se una richiesta è consentita, consulta Logica di valutazione delle policy.

I tag possono complicare questo processo perché possono essere collegati alla risorsa o trasferiti nella richiesta verso servizi che supportano il tagging. Per controllare gli accessi in base ai tag, devi fornire informazioni sui tag nell'elemento condizione di una policy. Per sapere se un AWS servizio supporta il controllo dell'accesso tramite tag, consulta AWS servizi supportati da IAM e cerca i servizi con Sì nella colonna ABAC. Scegli il nome del servizio per visualizzarne la documentazione sul controllo degli accessi e delle autorizzazioni.

Puoi quindi creare una policy IAM che consenta o rifiuti l'accesso a una risorsa in base al tag di quella risorsa. In quella policy, puoi utilizzare chiavi di condizione tag per controllare gli accessi a quanto segue:

- Risorsa: controlla l'accesso alle risorse del AWS servizio in base ai tag presenti su tali risorse. Per fare ciò, usa la chiave aws:ResourceTag/***key-name***condition per determinare se consentire l'accesso alla risorsa in base ai tag allegati alla risorsa.
- Risorsa: controlla quali tag possono essere passati in una richiesta. Per fare ciò, usa la chiave di *key-name* condizione aws:RequestTag/per specificare quali coppie chiave-valore di tag possono essere passate in una richiesta di taggare una AWS risorsa.
- Qualsiasi parte del processo di autorizzazione: usa la chiave aws: TagKeys condition per controllare se in una richiesta possono essere presenti chiavi di tag specifiche.

È possibile creare una policy IAM visivamente, utilizzando JSON o importando una policy gestita esistente. Per informazioni dettagliate, vedi Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente.

> **Note**
>
> Alcuni servizi consentono agli utenti di specificare tag durante la creazione della risorsa, se dispongono delle autorizzazioni per utilizzare l'operazione che crea la risorsa.

Controllo dell'accesso alle AWS risorse

Puoi utilizzare le condizioni delle tue policy IAM per controllare l'accesso alle AWS risorse in base ai tag presenti su quella risorsa. Puoi eseguire questa operazione utilizzando la chiave di condizione `aws:ResourceTag/tag-key` globale o una chiave specifica del servizio. Alcuni servizi, supportano solo la versione specifica del servizio di questa chiave e non la versione globale.

> **Warning**
>
> Non cercare di controllare chi può passare un ruolo assegnando tag al ruolo e utilizzando la chiave di condizione `ResourceTag` in una policy con l'operazione `iam:PassRole`. Questo approccio non produce risultati affidabili. Per ulteriori informazioni sulle autorizzazioni richieste per trasferire un ruolo a un servizio, consulta Concedi a un utente le autorizzazioni per passare un ruolo a un AWS service.

Questo esempio mostra come creare una policy basata sull'identità che consenta di avviare o arrestare le istanze Amazon EC2. Queste operazioni sono consentite solo se il tag dell'istanza `Owner` ha il valore del nome utente. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
```

```
            "Resource": "*"
        }
    ]
}
```

Puoi allegare questa policy agli utenti IAM nel tuo account. Se un utente denominato `richard` prova ad avviare un'istanza Amazon EC2, l'istanza deve avere il tag `0wner=richard` o `owner=richard`. In caso contrario, gli verrà negato l'accesso. La chiave di tag `0wner` corrisponde sia a `0wner` sia a `owner` perché i nomi delle chiavi di condizione non distinguono tra maiuscole e minuscole. Per ulteriori informazioni, consulta Elementi della policy JSON IAM: Condizione.

Questo esempio mostra come creare una policy basata sull'identità che utilizza il tag del principale `team` nell'ARN della risorsa. La policy concede l'autorizzazione per eliminare le code del servizio di coda semplice Amazon (Amazon SQS), ma solo se il nome della coda inizia con il nome del team seguito da `-queue`. Ad esempio, `qa-queue` se `qa` è il nome del team per il tag del principale `team`.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": {
        "Sid": "AllQueueActions",
        "Effect": "Allow",
        "Action": "sqs:DeleteQueue",
        "Resource": "arn:aws:sqs:us-east-2:111122223333:${aws:PrincipalTag/team}-
queue"
      }
}
```

Controllo dell'accesso durante AWS le richieste

Puoi utilizzare le condizioni nelle tue policy IAM per controllare quali coppie chiave-valore di tag possono essere passate in una richiesta che applica i tag a una AWS risorsa.

Questo esempio mostra come creare una policy basata sull'identità che consenta di utilizzare l'operazione `CreateTags` di Amazon EC2 per assegnare tag a un'istanza. Puoi collegare i tag solo se il tag contiene la chiave `environment` e i valori `production` o `preprod`. Se lo desideri, puoi utilizzare il modificatore `ForAllValues` con la chiave di condizione `aws:TagKeys` per indicare che

nella richiesta è ammessa solo la chiave `environment`. In questo modo gli utenti smetteranno di includere altre chiavi ad esempio utilizzando per errore `Environment` invece di `environment`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "ec2:CreateTags",
        "Resource": "arn:aws:ec2:*:*:instance/*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/environment": [
                    "preprod",
                    "production"
                ]
            },
            "ForAllValues:StringEquals": {"aws:TagKeys": "environment"}
        }
    }
}
```

Controllo dell'accesso in base alle chiavi di tag

Puoi utilizzare una condizione nelle policy IAM per controllare se determinate chiavi di tag possono essere utilizzate in una richiesta.

Quando utilizzi le policy per controllare l'accesso tramite tag, ti consigliamo di utilizzare la chiave `aws:TagKeys`condition. AWS i servizi che supportano i tag potrebbero consentire di creare più nomi di chiavi di tag che differiscono solo in base alle maiuscole e minuscole, ad esempio etichettare un'istanza `stack=production` Amazon EC2 con e. `Stack=test` Nelle condizioni delle policy i nomi delle chiavi non distinguono tra maiuscole e minuscole. Questo significa che se specifichi `"aws:ResourceTag/TagKey1": "Value1"` nell'elemento condizione della policy, la condizione corrisponderà a una chiave di tag della risorsa denominata `TagKey1` o `tagkey1`, ma non a entrambe. Per impedire la duplicazione dei tag con una chiave che cambia solo per le dimensioni dei caratteri, utilizza la condizione `aws:TagKeys` per definire le chiavi di tag applicabili dagli utenti, o usa le policy di tag disponibili con AWS Organizations. Per ulteriori informazioni, consulta Tag Policies nella Guida per l'utente di AWS Organizations .

Questo esempio mostra come creare una policy basata sull'identità che consenta di creare e assegnare tag a un secreto di Secrets Manager, ma solo con le chiavi di tag `environment` o `cost-center`. La condizione `Null` garantisce che la condizione sia `false` in assenza di tag nella richiesta.

```
{
        "Effect": "Allow",
        "Action": [
            "secretsmanager:CreateSecret",
            "secretsmanager:TagResource"
        ],
        "Resource": "*",
        "Condition": {
            "Null": {
                "aws:TagKeys": "false"
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": [
                    "environment",
                    "cost-center"
                ]
            }
        }
}
```

Accesso alle risorse multi-account in IAM

Per alcuni AWS servizi, puoi concedere l'accesso a più account alle tue risorse utilizzando IAM. A tale scopo, è possibile collegare una policy direttamente alla risorsa da condividere oppure utilizzare un ruolo come proxy.

Per condividere direttamente la risorsa, la risorsa da condividere deve supportare le policy basate su risorse. A differenza di una policy basata su identità per un ruolo, una policy basata su risorse specifica chi (quale principale) può accedere a tale risorsa.

Utilizza un ruolo come proxy quando desideri accedere a risorse di un altro account che non supportano le policy basate su risorse.

Per ulteriori dettagli sulle differenze tra questi tipi di policy, consulta la sezione Identity-based politiche e politiche basate sulle risorse.

> **Note**
>
> I ruoli IAM e le policy basate sulle risorse delegano l'accesso tra account solo all'interno di una singola partizione. Ad esempio, poniamo che tu abbia un account nella Regione Stati Uniti occidentali (California settentrionale) nella partizione `aws` standard. Hai anche un account in Cina nella partizione `aws-cn`. Non puoi utilizzare una politica basata sulle risorse nel tuo account in Cina per consentire l'accesso agli utenti del tuo account standard. AWS

Cross-account accesso tramite ruoli

Non tutti i AWS servizi supportano politiche basate sulle risorse. Per questi servizi, puoi utilizzare i ruoli IAM multi-account per centralizzare la gestione delle autorizzazioni quando fornisci l'accesso multi-account a più servizi. Un ruolo IAM su più account è un ruolo IAM che include una politica di fiducia che consente ai responsabili IAM di un altro AWS account di assumere il ruolo. In parole povere, puoi creare un ruolo in un AWS account che delega autorizzazioni specifiche a un altro account. AWS

Per informazioni sul collegamento di una policy a un'identità IAM, consulta Gestire le policy IAM.

> **Note**
>
> Quando un principale passa a un ruolo per utilizzare temporaneamente le rispettive autorizzazioni, rinuncia alle autorizzazioni originali e assume quelle assegnate al ruolo che ha assunto.

Diamo un'occhiata al processo complessivo prendendo in esame un software di un partner APN che deve accedere a un account cliente.

1. Il cliente crea un ruolo IAM nel proprio account con una policy IAM che consente l'accesso alle risorse Amazon S3 richieste dal partner APN. In questo esempio, il nome del ruolo è `APNPartner`.

   JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::bucket-name"
            ]
        }
    ]
}
```

2. Quindi, il cliente specifica che il ruolo può essere assunto dall' AWS account del partner fornendo l' Account AWS ID del partner APN nella politica di fiducia per il ruolo. APN`Partner`

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/APN-user-name"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

3. Il cliente fornisce il nome della risorsa Amazon (ARN) del ruolo al partner APN. L'ARN è il nome completo del ruolo.

```
arn:aws:iam::Customer-Account-ID:role/APNPartner
```

> **Note**
>
> Ti consigliamo di utilizzare un ID esterno in situazioni multi-tenant. Per informazioni dettagliate, vedi Accesso a Account AWS di proprietà di terzi.

4. Quando il software del Partner APN deve accedere all'account del cliente, il software richiama l'AssumeRoleAPI Servizio di token di sicurezza AWS con l'ARN del ruolo nell'account del cliente.

STS restituisce una AWS credenziale temporanea che consente al software di svolgere il proprio lavoro.

Per un altro esempio di concessione dell'accesso multi-account utilizzando i ruoli, consulta la sezione Accesso per un utente IAM in un altro Account AWS che possiedi. Puoi anche seguire il Tutorial IAM: Delega l'accesso attraverso AWS account che utilizzano ruoli IAM.

Cross-account accesso tramite politiche basate sulle risorse

Quando un account accede a una risorsa tramite un altro account utilizzando una policy basata su risorse, il principale continua a utilizzare l'account attendibile e non deve rinunciare alle proprie autorizzazioni per ricevere quelle del ruolo. In altre parole, il principale ha accesso alle risorse dell'account attendibile e contemporaneamente alla risorsa nell'account che concede fiducia. Questa funzione è utile per attività come la copia di informazioni da o verso la risorsa condivisa nell'altro account.

I principi che è possibile specificare in una policy basata sulle risorse includono account, utenti IAM, utenti principali federati, responsabili AWS STS federati SAML, responsabili federati OIDC, ruoli IAM, sessioni con ruolo presunto o servizi. AWS Per ulteriori informazioni, consulta la sezione Specifica di un principale.

Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta la sezione Identificazione di risorse condivise con un'entità esterna.

L'elenco seguente include alcuni dei servizi che supportano le policy basate sulle risorse. AWS Per un elenco completo del crescente numero di AWS servizi che supportano l'aggiunta di criteri di autorizzazione alle risorse anziché ai principali, vedi AWS servizi supportati da IAM e cerca i servizi che hanno Sì nella colonna Basato sulle risorse.

- Bucket Amazon S3: la policy è collegata al bucket, ma controlla l'accesso sia al bucket sia agli oggetti in esso contenuti. Per maggiori informazioni, consulta Policy del bucket per Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service. In alcuni casi, può essere opportuno utilizzare ruoli per l'accesso per più account ad Amazon S3. Per ulteriori informazioni, consulta le spiegazioni passo per passo di esempio nella Guida per l'utente di Amazon Simple Storage Service.

- Argomenti di Amazon Simple Notification Service (Amazon SNS): per ulteriori informazioni, consulta Esempi di casi per il controllo degli accessi di Amazon SNS nella Amazon Simple Notification Service Developer Guide.
- Code di Amazon Simple Queue Service (Amazon SQS): per ulteriori informazioni, consulta Appendice: The Access Policy Language nella Amazon Simple Queue Service Developer Guide.

Resource-based politiche da delegare AWS autorizzazioni

Se una risorsa concede le autorizzazioni ai principali nell'account, puoi delegare tali autorizzazioni a identità IAM specifiche. Le identità sono utenti, gruppi di utenti o ruoli nell'account. Per delegare le autorizzazioni, collegare una policy all'identità. È possibile concedere fino al numero massimo di autorizzazioni consentite dall'account proprietario della risorsa.

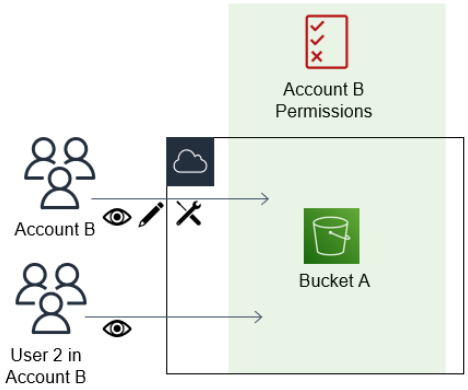
> **Important**
>
> Nell'accesso multi-account, il principale deve disporre della condizione `Allow` nella policy di identità e nella policy basata su risorse.

Si supponga che una policy basata sulle risorse consenta a tutti i principali nell'account l'accesso amministrativo completo a una risorsa. Quindi puoi delegare l'accesso completo, l'accesso in sola lettura o qualsiasi altro accesso parziale ai principali del tuo account. AWS In alternativa, se la policy basata sulle risorse consente solo le autorizzazioni per la presentazione di elenchi, è possibile delegare solo l'accesso all'elenco. Se si tenta di delegare più autorizzazioni rispetto a quelle possedute dall'account, i principali avranno comunque solo accesso all'elenco.

Per ulteriori informazioni su come vengono prese queste decisioni, consulta la sezione Determinare se una richiesta è consentita o rifiutata all'interno di un account.

> **Note**
>
> I ruoli IAM e le policy basate sulle risorse delegano l'accesso tra account solo all'interno di una singola partizione. Ad esempio, non è possibile aggiungere l'accesso tra account tra un account nella partizione `aws` standard e un account nella partizione `aws-cn`.

Ad esempio, si supponga di gestire `AccountA` e `AccountB`. Nell'AccountA, disponi di un bucket Amazon S3 denominato `BucketA`.



1. Colleghi una policy basata su risorse ad `BucketA` che consente a tutti i principali nell'AccountB l'accesso completo agli oggetti nel bucket. Possono creare, leggere o eliminare qualsiasi oggetto in tale bucket.

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Sid": "PrincipalAccess",
               "Effect": "Allow",
               "Principal": {
                   "AWS": "arn:aws:iam::111122223333:root"
               },
               "Action": "s3:*",
               "Resource": "arn:aws:s3:::BucketA/*"
           }
       ]
   }
   ```

   AccountA fornisce all'AccountB l'accesso completo al BucketA denominando AccountB come un principale nella policy basata su risorse. Di conseguenza, l'AccountB è autorizzato a eseguire qualsiasi operazione nel BucketA e l'amministratore dell'AccountB può delegare l'accesso ai propri utenti nell'AccountB.

   L'utente root dell'AccountB dispone di tutte le autorizzazioni concesse all'account. Pertanto, l'utente root dispone di accesso completo al BucketA.

2. Nell'AccountB, collega una policy all'utente IAM denominato User2. Tale policy consente all'utente l'accesso in sola lettura agli oggetti nel BucketA. Ciò significa che User2 può visualizzare gli oggetti, ma non crearli, modificarli o eliminarli.

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Effect" : "Allow",
               "Action" : [
                   "s3:Get*",
                   "s3:List*" ],
                   "Resource" : "arn:aws:s3:::BucketA/*"
           }
       ]
   }
   ```

   Il livello massimo di accesso che AccountB è in grado di delegare è il livello di accesso concesso all'account. In questo caso, la policy basata su risorse ha concesso l'accesso completo all'AccountB, ma User2 dispone solo dell'accesso di sola lettura.

   L'amministratore dell'AccountB non concede l'accesso a User1. Per impostazione predefinita, gli utenti non dispongono di autorizzazioni ad eccezione di quelle concesse in modo esplicito, pertanto User1 non ha accesso al BucketA.

IAM valuta le autorizzazioni di un principale nel momento in cui il principale effettua una richiesta. Se utilizzi i caratteri jolly (*) per consentire agli utenti l'accesso completo alle tue risorse, i titolari possono accedere a tutte le risorse a cui ha accesso il tuo account. AWS Ciò vale anche per le risorse che vengono aggiunte o a cui si ha accesso dopo aver creato le policy dell'utente.

Nell'esempio precedente, se l'AccountB avesse collegato a User2 una policy che concedeva l'accesso completo a tutte le risorse in tutti gli account, User2 avrebbe automaticamente avuto accesso a qualsiasi risorsa alla quale ha accesso l'AccountB. Ciò include l'accesso al BucketA e l'accesso a qualsiasi altra risorsa concesso dalle policy basate su risorse nell'AccountA.

Per ulteriori informazioni sugli usi complessi dei ruoli, come la concessione dell'accesso ad applicazioni e servizi, consulta la sezione Scenari comuni per i ruoli IAM.

> **Important**
>
> Concedere l'accesso solo a entità attendibili e fornire il livello minimo di accesso necessario. Ogni volta che l'entità attendibile è un altro AWS account, a qualsiasi responsabile IAM può essere concesso l'accesso alla tua risorsa. L' AWS account fidato può delegare l'accesso solo nella misura in cui gli è stato concesso l'accesso; non può delegare più accessi di quelli concessi all'account stesso.

Per ulteriori informazioni sulle autorizzazioni, le policy e il linguaggio di policy di autorizzazioni che è possibile utilizzare per scrivere policy, consultare Gestione degli accessi per AWS risorse.

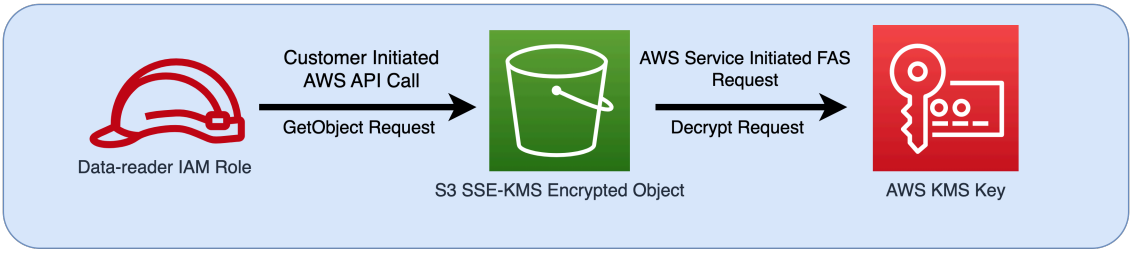
Inoltro delle sessioni di accesso

Le sessioni di accesso diretto (FAS) sono una tecnologia IAM utilizzata dai AWS servizi per trasmettere l'identità, le autorizzazioni e gli attributi di sessione dell'utente quando un AWS servizio effettua una richiesta per conto dell'utente. FAS utilizza le autorizzazioni dell'identità che chiama un AWS servizio, combinate con l'identità di un AWS servizio per effettuare richieste ai servizi a valle. Le richieste FAS vengono effettuate ai AWS servizi per conto di un responsabile IAM solo dopo che un servizio ha ricevuto una richiesta che richiede interazioni con altri AWS servizi o risorse per essere completata. Quando viene effettuata una richiesta FAS:

- Il servizio che riceve la richiesta iniziale da un principale IAM controlla le autorizzazioni di tale principale IAM.
- Anche il servizio che riceve la conseguente richiesta FAS controlla le autorizzazioni dello stesso principale IAM.

Ad esempio, FAS viene utilizzato da Amazon S3 per effettuare chiamate e decrittografare un oggetto quando SSE-KMS è stato utilizzato AWS Key Management Service per crittografarlo. Quando si scarica un oggetto SSE-KMS crittografato, un ruolo denominato data-reader richiama l'oggetto GetObject su Amazon S3 e non chiama direttamente. AWS KMS Dopo aver ricevuto la GetObject richiesta e aver autorizzato il lettore di dati, Amazon S3 effettua quindi una richiesta FAS a per decrittografare AWS KMS l'oggetto Amazon S3. Quando KMS riceve la richiesta FAS, controlla le autorizzazioni del ruolo e autorizza la richiesta di decrittografia solamente se data-reader dispone

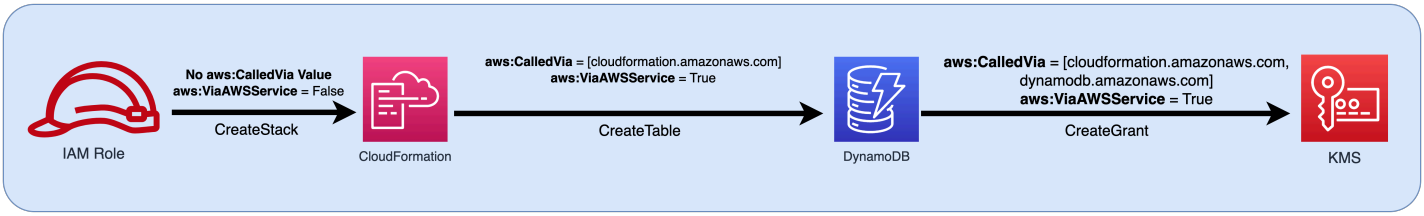
delle autorizzazioni corrette sulla chiave KMS. Le richieste AWS KMS vengono inviate sia ad Amazon S3 che a Amazon S3 utilizzando le autorizzazioni del ruolo e hanno esito positivo solo se il lettore di dati dispone delle autorizzazioni sia per l'oggetto Amazon S3 che per la chiave KMS. AWS



> **Note**
>
> I servizi che hanno ricevuto una richiesta FAS possono a loro volta effettuare richieste FAS aggiuntive. In questi casi, il principale che esegue la richiesta deve disporre delle autorizzazioni per tutti i servizi chiamati da FAS.

Richieste FAS e condizioni delle policy IAM

Quando vengono effettuate richieste FAS, le chiavi di condizione ha detto: CalledVia, aws: CalledViaFirst e era: CalledViaLast vengono compilate con il principale di servizio del servizio che ha avviato la chiamata FAS. Il valore della chiave di condizione aws:ViaAWSService viene impostato su `true` ogni volta che viene effettuata una richiesta FAS. Nel diagramma seguente, per la richiesta diretta non è stata impostata CloudFormation alcuna chiave di condizione. `aws:CalledVia` `aws:ViaAWSService` Quando CloudFormation e DynamoDB effettuano richieste FAS downstream per conto del ruolo, i valori di queste chiavi di condizione vengono compilati.



Per consentire l'esecuzione di una richiesta FAS che altrimenti verrebbe negata da un'istruzione di una policy di rifiuto con una chiave di condizione che verifica gli indirizzi IP o i VPC di origine, è necessario utilizzare le chiavi di condizione per impostare un'eccezione per le richieste FAS nella policy di rifiuto. Questa operazione può essere eseguita per tutte le richieste FAS utilizzando la

chiave di condizione `aws:ViaAWSService`. Per consentire solo a AWS servizi specifici di effettuare richieste FAS, usa. `aws:CalledVia`

> **Important**
>
> Quando viene effettuata una richiesta FAS in seguito a una richiesta iniziale effettuata tramite un endpoint VPC, i valori delle chiavi di condizione per `era: SourceVpce`, `era: SourceVpc` e `leggi: VpcSourceIp` della richiesta iniziale non vengono utilizzati nelle richieste FAS. Quando si scrivono policy utilizzando `aws:VpcSourceIp` o `aws:SourceVpce` per concedere l'accesso in modo condizionale, è inoltre necessario utilizzare `aws:ViaAWSService` o `aws:CalledVia` per consentire le richieste FAS. Quando una richiesta FAS viene effettuata dopo la ricezione di una richiesta iniziale da un endpoint di AWS servizio pubblico, le richieste FAS successive verranno effettuate con lo stesso `aws:SourceIP` valore di chiave di condizione.

Esempio: consentire ad Amazon S3 l'accesso da un VPC o tramite FAS

Nel seguente esempio di policy IAM, le richieste Amazon S3 GetObject e Athena sono consentite solo se provengono da endpoint VPC collegati a*`example_vpc`*, o se la richiesta è una richiesta FAS effettuata da Athena.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "OnlyAllowMyIPs",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject*",
        "athena:StartQueryExecution",
        "athena:GetQueryResults",
        "athena:GetWorkGroup",
        "athena:StopQueryExecution",
        "athena:GetQueryExecution"
      ],
      "Resource": "*",
      "Condition": {
```

```
        "StringEquals": {
          "aws:SourceVPC": [
          "vpc-111bbb22"
          ]
        }
      }
    },
    {
      "Sid": "OnlyAllowFAS",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject*"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": "athena.amazonaws.com"
        }
      }
    }
  ]
}
```

Per ulteriori esempi di utilizzo delle chiavi di condizione per consentire l'accesso FAS, consulta il repository di esempi di policy per implementare un perimetro di dati.

Esempi di policy basate su identità IAM

Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un responsabile IAM (utente o ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle policy viene archiviata AWS come documenti JSON allegati a un'identità IAM (utente, gruppo di utenti o ruolo). Le policy basate sulle identità includono policy gestite da AWS , policy gestite dal cliente e policy inline. Per ulteriori informazioni su come creare una policy IAM utilizzando questi documenti di policy JSON, consulta the section called "Creazione di policy utilizzando l'editor JSON".

Per impostazione predefinita, tutte le richieste vengono rifiutate, pertanto è necessario fornire l'accesso a servizi, azioni e risorse da rendere disponibili per l'identità. Se desideri consentire

l'accesso anche per completare le operazioni specificate nella console IAM, dovrai fornire ulteriori autorizzazioni.

La seguente libreria di policy può aiutarti a definire le autorizzazioni per le tue identità IAM. Una volta trovata la policy desiderata, seleziona view this policy (visualizza la policy) per consultare il JSON della policy. Puoi utilizzare il documento di policy JSON come modello per le tue policy.

> **Note**
>
> Per inviare una policy e includerla in questa guida di riferimento, utilizza il pulsante Feedback in fondo a questa pagina.

Politiche di esempio: AWS

- Consente l'accesso in un intervallo di date specifico. (Visualizzare questa policy).
- Consente di abilitare e disabilitare AWS le regioni. (Visualizzare questa policy).
- Consente agli utenti autenticati con MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Consente un accesso specifico quando utilizzi MFA durante un determinato intervallo di date. (Visualizzare questa policy).
- Consente agli utenti di gestire le proprie credenziali nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Consente agli utenti di gestire il proprio dispositivo MFA nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Consente agli utenti di gestire la propria password nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Consente agli utenti di gestire la propria password, le chiavi di accesso e le chiavi pubbliche SSH nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Nega l'accesso a in AWS base alla regione richiesta. (Visualizzare questa policy).
- Nega l'accesso a in AWS base all'indirizzo IP di origine. (Visualizzare questa policy).

Politica di esempio: AWS Data Exchange

- Nega l'accesso alle risorse Amazon S3 al di fuori del tuo account, tranne AWS Data Exchange. (Visualizzare questa policy).

Politiche di esempio: AWS Data Pipeline

- Rifiuta l'accesso alle pipeline non create dall'utente (Visualizzare questa policy).

Policy di esempio: Amazon DynamoDB

- Consente l'accesso a una specifica tabella Amazon DynamoDB (Visualizza questa policy).
- Consente l'accesso ad attributi Amazon DynamoDB specifici (Visualizza questa policy).
- Consente l'accesso a livello di elemento ad Amazon DynamoDB in base a un ID Amazon Cognito (Visualizza questa policy).

Politiche di esempio: Amazon EC2

- Consente di allegare o scollegare i volumi Amazon EBS alle EC2 istanze Amazon in base ai tag (Visualizza questa policy).
- Consente l'avvio di EC2 istanze Amazon in una sottorete specifica, a livello di codice e nella console (Visualizza questa politica).
- Consente la gestione dei gruppi EC2 di sicurezza Amazon associati a un VPC specifico, in modo programmatico e nella console (Visualizza questa policy).
- Consente di avviare o arrestare EC2 le istanze Amazon che un utente ha taggato, a livello di codice e nella console (Visualizza questa politica).
- Consente di avviare o arrestare EC2 istanze Amazon in base alle risorse e ai tag principali, a livello di codice e nella console (Visualizza questa politica).
- Consente di avviare o arrestare EC2 le istanze Amazon quando la risorsa e i tag principali corrispondono (Visualizza questa politica).
- Consente EC2 l'accesso completo ad Amazon all'interno di una regione specifica, a livello di programmazione e nella console. (Visualizzare questa policy).
- Consente l'avvio o l'arresto di un' EC2 istanza Amazon specifica e la modifica di un gruppo di sicurezza specifico, a livello di codice e nella console (Visualizza questa politica).
- Nega l'accesso a specifiche EC2 operazioni di Amazon senza MFA (Visualizza questa politica).
- Limita la terminazione EC2 delle istanze Amazon a un intervallo di indirizzi IP specifico (Visualizza questa politica).

Politiche di esempio: AWS Identity and Access Management (IAM)

- Consente l'accesso all'API del simulatore di policy (Visualizzare questa policy).
- Consente l'accesso alla console del simulatore di policy (Visualizzare questa policy).
- Consente l'assunzione di qualsiasi ruolo che dispone di un tag specifico, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente e nega l'accesso a più servizi, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente l'aggiunta di un tag specifico a un utente IAM con un altro tag specifico, a livello di programmazione e nella console (Visualizza questa policy).
- Consente l'aggiunta di un tag specifico a qualsiasi utente o ruolo IAM, a livello di programmazione e nella console (Visualizza questa policy).
- Consente la creazione di un nuovo utente solo con tag specifici (Visualizzare questa policy).
- Consente la generazione e il recupero di report delle credenziali IAM (Visualizzare questa policy).
- Consente la gestione dell'appartenenza a un gruppo, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente la gestione di un tag specifico (Visualizzare questa policy).
- Consente di passare un ruolo IAM a un servizio specifico (Visualizza questa policy).
- Consente l'accesso in sola lettura alla console IAM senza la creazione di report (Visualizza questa policy).
- Consente l'accesso in sola lettura alla console IAM (Visualizza questa policy).
- Consente a utenti specifici di gestire un gruppo, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente l'impostazione di requisiti della password dell'account, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente l'utilizzo dell'API del simulatore di policy per gli utenti con un percorso specifico (Visualizzare questa policy).
- Consente l'utilizzo della console del simulatore di policy per gli utenti con un percorso specifico (Visualizzare questa policy).
- Consente agli utenti IAM di gestire in modo autonomo un dispositivo MFA (Visualizzare questa policy).
- Consente agli utenti IAM di impostare le credenziali a livello di programmazione e nella console. (Visualizzare questa policy).

- Consente di visualizzare le informazioni sull'ultimo accesso al servizio per una AWS Organizations policy nella console IAM. (Visualizzare questa policy).
- Limita le policy gestite che possono essere applicate a un utente, un gruppo o un ruolo IAM (Visualizza questa policy).
- Consente l'accesso alle policy IAM solo nel tuo account. Visualizza questa policy.

Politiche di esempio: AWS Lambda

- Consente a una AWS Lambda funzione di accedere a una tabella Amazon DynamoDB (Visualizza questa policy).

Policy di esempio: Amazon RDS

- Consente l'accesso completo al database Amazon RDS in una regione specifica. (Visualizzare questa policy).
- Consente il ripristino dei database Amazon RDS, in modo programmatico e nella console (Visualizza questa policy)
- Consente ai proprietari di tag l'accesso completo alle risorse Amazon RDS che hanno taggato (Visualizza questa policy).

Policy di esempio: Amazon S3

- Consente a un utente Amazon Cognito di accedere a oggetti del proprio bucket Amazon S3 (Visualizza questa policy)
- Consente agli utenti con credenziali temporanee di accedere alla propria home directory in Amazon S3, in modo programmatico e nella console (Visualizza questa policy).
- Consente l'accesso S3 completo, ma nega esplicitamente l'accesso al bucket di produzione se l'amministratore non ha effettuato l'accesso utilizzando MFA negli ultimi trenta minuti (Visualizzare questa policy).
- Consente agli utenti IAM di accedere alla propria directory home in Amazon S3, in modo programmatico e nella console (Visualizza questa policy).
- Consente a un utente di gestire un singolo bucket Amazon S3 e nega ogni altra AWS azione e risorsa (Visualizza questa policy).

- Consente un accesso di tipo `Read` e `Write` a un bucket Amazon S3 specifico (Visualizza questa policy).
- Consente un accesso di tipo `Read` e `Write` a un bucket Amazon S3 specifico, in modo programmatico e nella console (Visualizza questa policy).

AWS: consente l'accesso in base alla data e all'ora

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso alle operazioni in base alla data e all'ora. Questa policy limita l'accesso alle operazioni che si verificano tra il 1° aprile 2020 e il 30 giugno 2020 (UTC), inclusi. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per ulteriori informazioni sull'utilizzo di condizioni multiple all'interno di un blocco `Condition` di una policy IAM, consultare Valori multipli in una condizione

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "service-prefix:action-name",
            "Resource": "*",
            "Condition": {
                "DateGreaterThan": {"aws:CurrentTime": "2020-04-01T00:00:00Z"},
                "DateLessThan": {"aws:CurrentTime": "2020-06-30T23:59:59Z"}
            }
        }
    ]
}
```

> **Note**
>
> Non è possibile utilizzare una variabile di policy con l'operatore di condizione Date. Per ulteriori informazioni, consulta la sezione Elemento condizione

AWS: consente di abilitare e disabilitare le regioni AWS

Questo esempio mostra come creare una policy basata sull'identità che consenta a un amministratore di abilitare e disabilitare la regione Asia Pacifico (Hong Kong) (ap-east-1). Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Questa impostazione viene visualizzata nella pagina Account settings (Impostazioni dell'account) nella Console di gestione AWS. Questa pagina include informazioni sensibili a livello di account, che devono essere visualizzate e gestite solo da amministratori dell'account. Per utilizzare questo criterio, sostituisci il criterio riportato *italicized placeholder text* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Important**
>
> Non è possibile abilitare o disabilitare le regioni abilitate per impostazione predefinita. Puoi includere solo le regioni disabilitate per impostazione predefinita. Per ulteriori informazioni, consulta Gestione delle regioni AWS nella Riferimenti generali di AWS.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EnableDisableHongKong",
            "Effect": "Allow",
            "Action": [
                "account:EnableRegion",
                "account:DisableRegion"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {"account:TargetRegion": "ap-east-1"}
```

```
            }
        },
        {
            "Sid": "ViewConsole",
            "Effect": "Allow",
            "Action": [
                "account:ListRegions"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS: consente agli utenti IAM autenticati con MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM autenticati tramite l'autenticazione a più fattori (MFA) di gestire le proprie credenziali sulla pagina Credenziali di sicurezza. Questa pagina della Console di gestione AWS mostra informazioni sull'account, come l'ID account e l'ID utente canonico. Gli utenti possono anche visualizzare e modificare le password, le chiavi di accesso, i dispositivi MFA, i certificati X.509, le chiavi SSH e le credenziali Git. Questa policy di esempio include le autorizzazioni necessarie per visualizzare e modificare tutte le informazioni sulla pagina. Richiede inoltre che l'utente si configuri e si autentichi tramite MFA prima di eseguire qualsiasi altra operazione in. AWS Per consentire agli utenti di gestire le proprie credenziali senza MFA, consulta AWS: consente agli utenti IAM di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Per ulteriori informazioni su come gli utenti possono accedere alla pagina Credenziali di sicurezza, consulta Come gli utenti IAM possono cambiare le proprie password (console).

> **Note**
>
> - Questo criterio di esempio non consente agli utenti di reimpostare una password durante Console di gestione AWS il primo accesso. Ti consigliamo di non concedere autorizzazioni ai nuovi utenti fino a quando non hanno effettuato l'accesso. Per ulteriori informazioni, consulta Come posso creare utenti IAM in modo sicuro?. Inoltre, ciò impedisce agli utenti con una password scaduta di reimpostare la password durante l'accesso. Per consentire questa operazione, aggiungere `iam:ChangePassword` e

`iam:GetAccountPasswordPolicy` all'istruzione `DenyAllExceptListedIfNoMFA`. Tuttavia, non ti consigliamo di farlo perché consentire agli utenti di cambiare la password senza MFA può costituire un rischio per la sicurezza.

- Se intendi utilizzare questa policy per l'accesso programmatico, devi chiamare `GetSessionToken` per l'autenticazione con l'MFA. Per ulteriori informazioni, consulta Accesso sicuro alle API con MFA.

Che cosa fa questa policy?

- L'istruzione `AllowViewAccountInfo` consente all'utente di visualizzare le informazioni a livello di account. Queste autorizzazioni devono essere nella propria istruzione perché non supportano o non devono specificare l'ARN di una risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`. Questa istruzione include le seguenti operazioni che consentono all'utente di visualizzare informazioni specifiche:
  - `GetAccountPasswordPolicy`: visualizza i requisiti della password dell'account cambiando la propria password utente IAM.
  - `ListVirtualMFADevices`: consente di visualizzare i dettagli di un dispositivo MFA virtuale abilitato per l'utente.
- L'istruzione `AllowManageOwnPasswords` consente all'utente di modificare la propria password. Questa istruzione include anche l'operazione `GetUser`, obbligatoria per visualizzare la maggior parte delle informazioni nella pagina My security credentials (Le mie credenziali di sicurezza).
- L'istruzione `AllowManageOwnAccessKeys` consente all'utente di creare, aggiornare ed eliminare le proprie chiavi di accesso. L'utente può anche ottenere informazioni su quando è stata utilizzata l'ultima volta la chiave di accesso specificata.
- L'istruzione `AllowManageOwnSigningCertificates` consente all'utente di caricare, aggiornare ed eliminare i propri certificati di firma.
- L'istruzione `AllowManageOwnSSHPublicKeys` consente all'utente di caricare, aggiornare ed eliminare le proprie chiavi pubbliche SSH per CodeCommit.
- L'istruzione `AllowManageOwnGitCredentials` consente all'utente di creare, aggiornare ed eliminare le proprie credenziali Git per CodeCommit.
- L'istruzione `AllowManageOwnVirtualMFADevice` consente all'utente di creare il proprio dispositivo virtuale MFA. L'ARN della risorsa in questa istruzione consente all'utente di creare un dispositivo MFA con qualsiasi nome, ma le altre istruzioni nella policy consentono all'utente soltanto di collegare il dispositivo all'utente correntemente registrato.

- L'istruzione `AllowManageOwnUserMFA` consente all'utente di visualizzare o gestire il dispositivo MFA virtuale, U2F o hardware per il proprio utente. L'ARN della risorsa in questa istruzione consente di accedere solo all'utente IAM dell'utente stesso. Gli utenti non possono visualizzare o gestire il dispositivo MFA per altri utenti.
- L'`DenyAllExceptListedIfNoMFA`istruzione nega l'accesso a tutte le azioni in tutti i AWS servizi, ad eccezione di alcune azioni elencate, ma solo se l'utente non ha effettuato l'accesso con MFA. L'istruzione utilizza una combinazione di `"Deny"` e `"NotAction"` per negare esplicitamente l'accesso a tutte le operazioni che non sono nell'elenco. Gli elementi elencati non sono negati o consentiti da questa istruzione. Tuttavia, le azioni sono consentite da altre istruzioni della policy. Per ulteriori informazioni sulla logica di questa istruzione, vedere NotAction with Deny. Se l'utente ha eseguito l'accesso con l'autenticazione MFA, il test `Condition` dà esito negativo e questa istruzione non produce effetti. In questo caso, altre policy o dichiarazioni per l'utente determinano le autorizzazioni dell'utente.

  Questa istruzione garantisce che quando l'utente non ha effettuato l'accesso con MFA può eseguire solo le operazioni elencate. Inoltre, possono eseguire le operazioni elencate, solo se un'altra istruzione o policy consente l'accesso a tali operazioni. Questo non consente a un utente di creare una password all'accesso, perché l'operazione `iam:ChangePassword` non deve essere consentita senza l'autorizzazione MFA.

  La versione `...IfExists` dell'operatore `Bool` garantisce che se la chiave `era:MultiFactorAuthPresent` manca, la condizione restituisce true. Questo significa che a un utente che accede a un'API con le credenziali di lungo termine, ad esempio con una chiave di accesso, viene negato l'accesso alle operazioni API non IAM.

Questa policy non consente agli utenti di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione, aggiungere l'operazione `iam:ListUsers` all'istruzione `AllowViewAccountInfo` e all'istruzione `DenyAllExceptListedIfNoMFA`. Inoltre, non consente agli utenti di cambiare la password sulla proprio pagina utente. Per consentire questa operazione, aggiungi le operazioni `iam:GetLoginProfile` e `iam:UpdateLoginProfile` all'istruzione `AllowManageOwnPasswords`. Inoltre, per consente a un utente di cambiare la password dalla propria pagina utente senza l'accesso tramite MFA, aggiungere l'operazione `iam:UpdateLoginProfile` all'istruzione `DenyAllExceptListedIfNoMFA`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewAccountInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetAccountPasswordPolicy",
                "iam:ListVirtualMFADevices"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnPasswords",
            "Effect": "Allow",
            "Action": [
                "iam:ChangePassword",
                "iam:GetUser"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
                "iam:DeleteAccessKey",
                "iam:ListAccessKeys",
                "iam:UpdateAccessKey",
                "iam:GetAccessKeyLastUsed"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnSigningCertificates",
            "Effect": "Allow",
            "Action": [
                "iam:DeleteSigningCertificate",
                "iam:ListSigningCertificates",
                "iam:UpdateSigningCertificate",
                "iam:UploadSigningCertificate"
```

```
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnSSHPublicKeys",
            "Effect": "Allow",
            "Action": [
                "iam:DeleteSSHPublicKey",
                "iam:GetSSHPublicKey",
                "iam:ListSSHPublicKeys",
                "iam:UpdateSSHPublicKey",
                "iam:UploadSSHPublicKey"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnGitCredentials",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceSpecificCredential",
                "iam:DeleteServiceSpecificCredential",
                "iam:ListServiceSpecificCredentials",
                "iam:ResetServiceSpecificCredential",
                "iam:UpdateServiceSpecificCredential"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnVirtualMFADevice",
            "Effect": "Allow",
            "Action": [
                "iam:CreateVirtualMFADevice"
            ],
            "Resource": "arn:aws:iam::*:mfa/*"
        },
        {
            "Sid": "AllowManageOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:EnableMFADevice",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
```

```
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "DenyAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:CreateVirtualMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:GetMFADevice",
                "iam:ListMFADevices",
                "iam:ListVirtualMFADevices",
                "iam:ResyncMFADevice",
                "sts:GetSessionToken"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
    ]
}
```

AWS: consente l'accesso specifico tramite MFA entro date specifiche

Questo esempio mostra come creare una policy basata sull'identità che utilizzi più condizioni che vengono valutate in base a un operatore AND logico. Consente l'accesso completo al servizio denominato `SERVICE-NAME-1` e l'accesso alle operazioni `ACTION-NAME-A` e `ACTION-NAME-B` nel servizio denominato `SERVICE-NAME-2`. Queste operazioni sono consentite solo quando l'utente è autenticato tramite l'autenticazione a più fattori (MFA). L'accesso è limitato alle operazioni effettuate tra il 1 luglio 2017 e il 31 dicembre 2017 (UTC), inclusi. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per ulteriori informazioni sull'utilizzo di condizioni multiple all'interno di un blocco `Condition` di una policy IAM, consultare Valori multipli in una condizione

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "service-prefix-1:*",
            "service-prefix-2:action-name-a",
            "service-prefix-2:action-name-b"
        ],
        "Resource": "*",
        "Condition": {
            "Bool": {"aws:MultiFactorAuthPresent": true},
            "DateGreaterThan": {"aws:CurrentTime": "2017-07-01T00:00:00Z"},
            "DateLessThan": {"aws:CurrentTime": "2017-12-31T23:59:59Z"}
        }
    }
}
```

AWS: consente agli utenti IAM di gestire le proprie credenziali nella pagina Credenziali di sicurezza

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di gestire le proprie credenziali nella pagina Credenziali di sicurezza. Questa Console di gestione AWS pagina mostra informazioni sull'account come l'ID account e l'ID utente canonico. Gli utenti possono anche visualizzare e modificare le password, le chiavi di accesso, i certificati X.509, le chiavi SSH e le credenziali Git. Questa policy di esempio include le autorizzazioni necessarie per visualizzare e modificare le informazioni sulla pagina eccetto il dispositivo MFA dell'utente. Per consentire agli utenti di gestire le proprie credenziali con MFA, consulta AWS: consente agli utenti IAM autenticati con MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Per ulteriori informazioni su come gli utenti possono accedere alla pagina Credenziali di sicurezza, consulta Come gli utenti IAM possono cambiare le proprie password (console).

Che cosa fa questa policy?

- L'istruzione `AllowViewAccountInfo` consente all'utente di visualizzare le informazioni a livello di account. Queste autorizzazioni devono essere nella propria istruzione perché non supportano o

non devono specificare l'ARN di una risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`. Questa istruzione include le seguenti operazioni che consentono all'utente di visualizzare informazioni specifiche:

  - `GetAccountPasswordPolicy`: visualizza i requisiti della password dell'account cambiando la propria password utente IAM.
  - `GetAccountSummary`: visualizza l'ID account e l'ID utente canonico dell'account.
- L'istruzione `AllowManageOwnPasswords` consente all'utente di modificare la propria password. Questa istruzione include anche l'operazione `GetUser`, obbligatoria per visualizzare la maggior parte delle informazioni nella pagina My security credentials (Le mie credenziali di sicurezza).
- L'istruzione `AllowManageOwnAccessKeys` consente all'utente di creare, aggiornare ed eliminare le proprie chiavi di accesso. L'utente può anche ottenere informazioni su quando è stata utilizzata l'ultima volta la chiave di accesso specificata.
- L'istruzione `AllowManageOwnSigningCertificates` consente all'utente di caricare, aggiornare ed eliminare i propri certificati di firma.
- L'istruzione `AllowManageOwnSSHPublicKeys` consente all'utente di caricare, aggiornare ed eliminare le proprie chiavi pubbliche SSH per CodeCommit.
- L'istruzione `AllowManageOwnGitCredentials` consente all'utente di creare, aggiornare ed eliminare le proprie credenziali Git per CodeCommit.

Questa policy non consente agli utenti di visualizzare o gestire i propri dispositivi MFA. Inoltre, non sono in grado di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione, aggiungere l'operazione `iam:ListUsers` all'istruzione `AllowViewAccountInfo`. Inoltre, non consente agli utenti di cambiare la password sulla proprio pagina utente. Per consentire questa operazione, aggiungere le operazioni `iam:CreateLoginProfile`, `iam:DeleteLoginProfile`, `iam:GetLoginProfile` e `iam:UpdateLoginProfile` all'istruzione `AllowManageOwnPasswords`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewAccountInfo",
            "Effect": "Allow",
            "Action": [
```

```
            "iam:GetAccountPasswordPolicy",
            "iam:GetAccountSummary"
        ],
        "Resource": "*"
    },
    {
        "Sid": "AllowManageOwnPasswords",
        "Effect": "Allow",
        "Action": [
            "iam:ChangePassword",
            "iam:GetUser"
        ],
        "Resource": "arn:aws:iam::*:user/${aws:username}"
    },
    {
        "Sid": "AllowManageOwnAccessKeys",
        "Effect": "Allow",
        "Action": [
            "iam:CreateAccessKey",
            "iam:DeleteAccessKey",
            "iam:ListAccessKeys",
            "iam:UpdateAccessKey",
            "iam:GetAccessKeyLastUsed"
        ],
        "Resource": "arn:aws:iam::*:user/${aws:username}"
    },
    {
        "Sid": "AllowManageOwnSigningCertificates",
        "Effect": "Allow",
        "Action": [
            "iam:DeleteSigningCertificate",
            "iam:ListSigningCertificates",
            "iam:UpdateSigningCertificate",
            "iam:UploadSigningCertificate"
        ],
        "Resource": "arn:aws:iam::*:user/${aws:username}"
    },
    {
        "Sid": "AllowManageOwnSSHPublicKeys",
        "Effect": "Allow",
        "Action": [
            "iam:DeleteSSHPublicKey",
            "iam:GetSSHPublicKey",
            "iam:ListSSHPublicKeys",
```

```
                "iam:UpdateSSHPublicKey",
                "iam:UploadSSHPublicKey"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnGitCredentials",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceSpecificCredential",
                "iam:DeleteServiceSpecificCredential",
                "iam:ListServiceSpecificCredentials",
                "iam:ResetServiceSpecificCredential",
                "iam:UpdateServiceSpecificCredential"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

AWS: consente agli utenti IAM autenticati con MFA di gestire il proprio dispositivo MFA nella pagina Credenziali di sicurezza

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM autenticati tramite l'autenticazione a più fattori (MFA) di gestire il proprio dispositivo MFA sulla pagina Credenziali di sicurezza. Questa Console di gestione AWS pagina mostra le informazioni sull'account e sull'utente, ma l'utente può solo visualizzare e modificare il proprio dispositivo MFA. Per consentire agli utenti di gestire le proprie credenziali con MFA, consulta AWS: consente agli utenti IAM autenticati con MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

> **Note**
>
> Se un utente IAM con questa policy non è autenticato tramite MFA, questa policy nega l'accesso a tutte le AWS azioni tranne quelle necessarie per l'autenticazione tramite MFA. Per utilizzare l' AWS API AWS CLI and, gli utenti IAM devono prima recuperare il proprio token MFA utilizzando AWS STS GetSessionTokenl'operazione e quindi utilizzare quel token per autenticare l'operazione desiderata. Altre policy, ad esempio le policy basate sulle risorse o altre policy basate sull'identità, possono consentire operazioni in altri servizi. Questa policy negherà tale accesso se l'utente IAM non dispone dell'autenticazione MFA.

Per ulteriori informazioni su come gli utenti possono accedere alla pagina Credenziali di sicurezza, consulta Come gli utenti IAM possono cambiare le proprie password (console).

Che cosa fa questa policy?

- L'istruzione `AllowViewAccountInfo` consente all'utente di visualizzare i dettagli di un dispositivo MFA virtuale abilitato per l'utente. Questa autorizzazione deve essere nella propria dichiarazione perché non specifica un ARN della risorsa. È necessario invece specificare `"Resource" : "*"`.
- L'istruzione `AllowManageOwnVirtualMFADevice` consente all'utente di creare il proprio dispositivo virtuale MFA. L'ARN della risorsa in questa istruzione consente all'utente di creare un dispositivo MFA con qualsiasi nome, ma le altre istruzioni nella policy consentono all'utente soltanto di collegare il dispositivo all'utente correntemente registrato.
- L'istruzione `AllowManageOwnUserMFA` consente all'utente di visualizzare o gestire il proprio dispositivo MFA virtuale, U2F o hardware. L'ARN della risorsa in questa istruzione consente di accedere solo all'utente IAM dell'utente stesso. Gli utenti non possono visualizzare o gestire il dispositivo MFA per altri utenti.
- L'`DenyAllExceptListedIfNoMFA`istruzione nega l'accesso a tutte le azioni in tutti i AWS servizi, ad eccezione di alcune azioni elencate, ma solo se l'utente non ha effettuato l'accesso con MFA. L'istruzione utilizza una combinazione di `"Deny"` e `"NotAction"` per negare esplicitamente l'accesso a tutte le operazioni che non sono nell'elenco. Gli elementi elencati non sono negati o consentiti da questa istruzione. Tuttavia, le azioni sono consentite da altre istruzioni della policy. Per ulteriori informazioni sulla logica di questa istruzione, vedere NotAction with Deny. Se l'utente ha eseguito l'accesso con l'autenticazione MFA, il test `Condition` dà esito negativo e questa istruzione non produce effetti. In questo caso, altre policy o dichiarazioni per l'utente determinano le autorizzazioni dell'utente.

  Questa istruzione garantisce che quando l'utente non ha effettuato l'accesso con MFA può eseguire solo le operazioni elencate. Inoltre, possono eseguire le operazioni elencate, solo se un'altra istruzione o policy consente l'accesso a tali operazioni.

  La versione `...IfExists` dell'operatore `Bool` garantisce che se la chiave `aws:MultiFactorAuthPresent` manca, la condizione restituisce true. Questo significa che a un utente che accede a un'operazione API con le credenziali di lungo termine, come una chiave di accesso, viene negato l'accesso alle operazioni API non IAM.

Questa policy non consente agli utenti di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione,

aggiungere l'operazione `iam:ListUsers` all'istruzione `AllowViewAccountInfo` e all'istruzione `DenyAllExceptListedIfNoMFA`.

> **Warning**
>
> Non aggiungere l'autorizzazione per l'eliminazione di un dispositivo MFA senza autenticazione MFA. Gli utenti con questa policy potrebbero tentare di autoassegnarsi un dispositivo MFA virtuale e ricevere un errore in cui si specifica che non sono autorizzati a eseguire `iam:DeleteVirtualMFADevice`. In questo caso, non aggiungere tale autorizzazione all'istruzione `DenyAllExceptListedIfNoMFA`. Agli utenti che non si autenticano tramite MFA non deve mai essere consentito di eliminare il dispositivo MFA. Gli utenti potrebbero visualizzare questo errore se hanno in precedenza iniziato ad assegnare un dispositivo MFA virtuale all'utente e annullato il processo. Per risolvere questo problema, tu o un altro amministratore dovete eliminare il dispositivo MFA virtuale esistente dell'utente utilizzando l'API AWS CLI o AWS . Per ulteriori informazioni, consulta Non sono autorizzato a eseguire: iam: DeleteVirtualMFADevice.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewAccountInfo",
            "Effect": "Allow",
            "Action": "iam:ListVirtualMFADevices",
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnVirtualMFADevice",
            "Effect": "Allow",
            "Action": [
                "iam:CreateVirtualMFADevice"
            ],
            "Resource": "arn:aws:iam::*:mfa/*"
        },
        {
            "Sid": "AllowManageOwnUserMFA",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:GetMFADevice",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "DenyAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:CreateVirtualMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ListVirtualMFADevices",
                "iam:ResyncMFADevice",
                "sts:GetSessionToken"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {"aws:MultiFactorAuthPresent": "false"}
            }
        }
    ]
}
```

AWS: consente agli utenti IAM di modificare la password della console sulla pagina Credenziali di sicurezza

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta agli utenti IAM di modificare la propria Console di gestione AWS password nella pagina Credenziali di sicurezza. Questa Console di gestione AWS pagina mostra le informazioni sull'account e sull'utente, ma l'utente può accedere solo alla propria password. Per consentire agli utenti di gestire le proprie credenziali con MFA, consulta AWS: consente agli utenti IAM autenticati con MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza. Per consentire agli utenti di gestire le proprie credenziali senza MFA, consulta AWS: consente agli utenti IAM di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Per ulteriori informazioni su come gli utenti possono accedere alla pagina Credenziali di sicurezza, consulta Come gli utenti IAM possono cambiare le proprie password (console).

Che cosa fa questa policy?

- L'istruzione `ViewAccountPasswordRequirements` consente all'utente di visualizzare i requisiti della password dell'account cambiando la propria password utente IAM.
- L'istruzione `ChangeOwnPassword` consente all'utente di modificare la propria password. Questa istruzione include anche l'operazione `GetUser`, obbligatoria per visualizzare la maggior parte delle informazioni nella pagina My security credentials (Le mie credenziali di sicurezza).

Questa policy non consente agli utenti di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione, aggiungere l'operazione `iam:ListUsers` all'istruzione `ViewAccountPasswordRequirements`. Inoltre, non consente agli utenti di cambiare la password sulla proprio pagina utente. Per consentire questa operazione, aggiungi le operazioni `iam:GetLoginProfile` e `iam:UpdateLoginProfile` all'istruzione `ChangeOwnPassword`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewAccountPasswordRequirements",
            "Effect": "Allow",
            "Action": "iam:GetAccountPasswordPolicy",
            "Resource": "*"
        },
        {
            "Sid": "ChangeOwnPassword",
            "Effect": "Allow",
            "Action": [
                "iam:GetUser",
                "iam:ChangePassword"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
```

```
}
```

AWS: consente agli utenti IAM di gestire la propria password, le chiavi di accesso e le chiavi pubbliche SSH nella pagina Credenziali di sicurezza

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di gestire la propria password, le chiavi di accesso e i certificati X.509 nella pagina Credenziali di sicurezza. Questa pagina della Console di gestione AWS mostra informazioni sull'account, come l'ID account e l'ID utente canonico. Gli utenti possono anche visualizzare e modificare le password, le chiavi di accesso, i dispositivi MFA, i certificati X.509, le chiavi SSH e le credenziali Git. Questa policy di esempio include le autorizzazioni necessarie per visualizzare e modificare solo le password, le chiavi di accesso e il certificato X.509. Per consentire agli utenti di gestire le proprie credenziali con MFA, consulta AWS: consente agli utenti IAM autenticati con MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza. Per consentire agli utenti di gestire le proprie credenziali senza MFA, consulta AWS: consente agli utenti IAM di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Per ulteriori informazioni su come gli utenti possono accedere alla pagina Credenziali di sicurezza, consulta Come gli utenti IAM possono cambiare le proprie password (console).

Che cosa fa questa policy?

- L'istruzione `AllowViewAccountInfo` consente all'utente di visualizzare le informazioni a livello di account. Queste autorizzazioni devono essere nella propria istruzione perché non supportano o non devono specificare l'ARN di una risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`. Questa istruzione include le seguenti operazioni che consentono all'utente di visualizzare informazioni specifiche:
  - `GetAccountPasswordPolicy`: visualizza i requisiti della password dell'account cambiando la propria password utente IAM.
  - `GetAccountSummary`: visualizza l'ID account e l'ID utente canonico dell'account.
- L'istruzione `AllowManageOwnPasswords` consente all'utente di modificare la propria password. Questa istruzione include anche l'operazione `GetUser`, obbligatoria per visualizzare la maggior parte delle informazioni nella pagina My security credentials (Le mie credenziali di sicurezza).
- L'istruzione `AllowManageOwnAccessKeys` consente all'utente di creare, aggiornare ed eliminare le proprie chiavi di accesso. L'utente può anche ottenere informazioni su quando è stata utilizzata l'ultima volta la chiave di accesso specificata.

- L'istruzione `AllowManageOwnSSHPublicKeys` consente all'utente di caricare, aggiornare ed eliminare le proprie chiavi pubbliche SSH per CodeCommit.

Questa policy non consente agli utenti di visualizzare o gestire i propri dispositivi MFA. Inoltre, non sono in grado di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione, aggiungere l'operazione `iam:ListUsers` all'istruzione `AllowViewAccountInfo`. Inoltre, non consente agli utenti di cambiare la password sulla proprio pagina utente. Per consentire questa operazione, aggiungi le operazioni `iam:GetLoginProfile` e `iam:UpdateLoginProfile` all'istruzione `AllowManageOwnPasswords`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewAccountInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetAccountPasswordPolicy",
                "iam:GetAccountSummary"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnPasswords",
            "Effect": "Allow",
            "Action": [
                "iam:ChangePassword",
                "iam:GetUser"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
                "iam:DeleteAccessKey",
                "iam:ListAccessKeys",
```

```
                "iam:UpdateAccessKey",
                "iam:GetAccessKeyLastUsed"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnSSHPublicKeys",
            "Effect": "Allow",
            "Action": [
                "iam:DeleteSSHPublicKey",
                "iam:GetSSHPublicKey",
                "iam:ListSSHPublicKeys",
                "iam:UpdateSSHPublicKey",
                "iam:UploadSSHPublicKey"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

AWS: nega l'accesso in AWS base alla regione richiesta

Questo esempio illustra come creare una policy basata sull'identità che neghi l'accesso a qualsiasi operazione esterna alle regioni specificate utilizzando la chiave di condizione `aws:RequestedRegion`, fatta eccezione per le operazioni nei servizi specificati tramite `NotAction`. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Questa policy utilizza l'elemento `NotAction` con l'effetto `Deny`, che rifiuta esplicitamente l'accesso a tutte le operazioni che non sono elencate nella dichiarazione. Le azioni relative a IAM CloudFront, Route 53 e Supporto ai servizi non devono essere negate, poiché si tratta di servizi AWS globali molto diffusi con un unico endpoint che si trova fisicamente nella `us-east-1` regione. Poiché tutte le richieste a questi servizi vengono effettuate alla regione `us-east-1`, le richieste vengono rifiutate senza l'elemento `NotAction`. Modifica questo elemento per includere operazioni per altri servizi globali AWS che utilizzi, ad esempio `budgets`, `globalaccelerator`, `importexport`, `organizations` o `waf`. Alcuni altri servizi globali, come Amazon Q Developer nelle applicazioni di chat AWS Device Farm, sono servizi globali con endpoint che si trovano fisicamente nella `us-west-2` regione. Per ulteriori informazioni su tutti i servizi che dispongono di un singolo

endpoint globale, consulta Regioni ed endpoint AWS nella Riferimenti generali di AWS. Per ulteriori informazioni sull'utilizzo dell'elemento `NotAction` con l'effetto `Deny`, consulta Elementi delle policy JSON IAM: NotAction.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DenyAllOutsideRequestedRegions",
            "Effect": "Deny",
            "NotAction": [
                "cloudfront:*",
                "iam:*",
                "organizations:*",
                "route53:*",
                "support:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:RequestedRegion": [
                        "eu-central-1",
                        "eu-west-1",
                        "eu-west-2",
                        "eu-west-3"
                    ]
                }
            }
        }
    ]
}
```

AWS: nega l'accesso in AWS base all'IP di origine

Questo esempio mostra come è possibile creare una policy basata sull'identità che neghi l'accesso a tutte le AWS azioni dell'account quando la richiesta proviene da soggetti esterni all'intervallo IP specificato. La policy è utile quando gli indirizzi IP per la tua azienda sono all'interno di determinati intervalli. In questo esempio, la richiesta verrà rifiutata a meno che non provenga dall'intervallo CIDR 192.0.2.0/24 o 203.0.113.0/24. La politica non nega le richieste effettuate dai AWS servizi che utilizzano l'indirizzo IP Inoltro delle sessioni di accesso del richiedente originale.

Fare attenzione a utilizzare condizioni negative nella stessa dichiarazione di policy come `"Effect": "Deny"`. In questo caso, le operazioni specificate nella dichiarazione di policy vengono negate in modo esplicito in tutte le condizioni, ad eccezione di quelle specificate.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

Quando altre policy consentono operazioni, le entità possono effettuare richieste all'interno dell'intervallo di indirizzi IP. Un AWS servizio può anche effettuare richieste utilizzando le credenziali del principale. Quando un'entità effettua una richiesta al di fuori dell'intervallo IP, la richiesta viene negata.

Per ulteriori informazioni sull'utilizzo della chiave di condizione `aws:SourceIp`, incluse le informazioni su quando `aws:SourceIp` potrebbe non funzionare nelle policy, consulta AWS chiavi di contesto delle condizioni globali.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
            "NotIpAddress": {
                "aws:SourceIp": [
```

```
                    "192.0.2.0/24",
                    "203.0.113.0/24"
                ]
            }
        }
    }
}
```

AWS: nega l'accesso alle risorse Amazon S3 al di fuori del tuo account tranne AWS Data Exchange

Questo esempio mostra come potresti creare una politica basata sull'identità che neghi l'accesso a tutte le risorse AWS che non appartengono al tuo account, ad eccezione delle risorse necessarie per il normale funzionamento. AWS Data Exchange Per utilizzare questa politica, sostituisci la politica dell'*italicized placeholder text*esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

È possibile creare una politica simile per limitare l'accesso alle risorse all'interno di un'organizzazione o di un'unità organizzativa, contabilizzando al contempo le risorse di AWS Data Exchange proprietà utilizzando i tasti di condizione `aws:ResourceOrgPaths` e`aws:ResourceOrgID`.

Se lo utilizzi AWS Data Exchange nel tuo ambiente, il servizio crea e interagisce con risorse come i bucket Amazon S3 di proprietà dell'account del servizio. Ad esempio, AWS Data Exchange invia richieste ai bucket Amazon S3 di proprietà del AWS Data Exchange servizio per conto del principale IAM (utente o ruolo) invocando il. AWS Data Exchange APIs In tal caso, l'utilizzo `aws:ResourceAccount`aws:ResourceOrgPaths, o `aws:ResourceOrgID` nell'ambito di una policy, senza tenere conto delle risorse di AWS Data Exchange proprietà, nega l'accesso ai bucket di proprietà dell'account di servizio.

- L'istruzione `DenyAllAwsResourcesOutsideAccountExceptS3` utilizza l'elemento `NotAction` con l'effetto Deny che nega esplicitamente l'accesso a tutte le operazioni non elencate nell'istruzione e che non appartengono all'account elencato. L'elemento `NotAction` indica le eccezioni a questa istruzione. Queste azioni fanno eccezione a questa dichiarazione perché se le azioni vengono eseguite su risorse create da AWS Data Exchange, la policy le nega.
- L'istruzione `DenyAllS3ResourcesOutsideAccountExceptDataExchange` utilizza una combinazione delle condizioni `ResourceAccount` e `CalledVia` per negare l'accesso alle tre operazioni di Amazon S3 escluse nell'istruzione precedente. L'istruzione nega le operazioni se le risorse non appartengono all'account elencato e se il servizio chiamante

non è AWS Data Exchange. L'istruzione non nega le operazioni se la risorsa appartiene all'account elencato o l'operazione viene eseguita dal principale del servizio elencato, `dataexchange.amazonaws.com`.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza l'effetto `Deny`, che neghi esplicitamente l'accesso a tutte le risorse elencate nell'istruzione che non appartengono all'account elencato. Utilizza questa policy in combinazione con altre policy che consentono l'accesso a risorse specifiche.

L'esempio seguente mostra come configurare la policy per consentire l'accesso ai bucket Amazon S3 richiesti.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAllAwsReourcesOutsideAccountExceptAmazonS3",
      "Effect": "Deny",
      "NotAction": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "111122223333"
          ]
        }
      }
    },
    {
      "Sid": "DenyAllS3ResourcesOutsideAccountExceptDataExchange",
      "Effect": "Deny",
```

```
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "111122223333"
          ]
        },
        "ForAllValues:StringNotEquals": {
          "aws:CalledVia": [
            "dataexchange.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

AWS Data Pipeline: nega l'accesso alle DataPipeline pipeline che un utente non ha creato

Questo esempio mostra come creare una policy basata sull'identità che neghi l'accesso alle pipeline che non sono state create da un utente. Se il valore del campo `PipelineCreator` corrisponde al nome dell'utente IAM, le operazioni specificate non saranno rifiutate. Questa policy concede le autorizzazioni necessarie per completare questa azione in modo programmatico dall'API o. AWS AWS CLI

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ExplicitDenyIfNotTheOwner",
            "Effect": "Deny",
            "Action": [
                "datapipeline:ActivatePipeline",
                "datapipeline:AddTags",
                "datapipeline:DeactivatePipeline",
                "datapipeline:DeletePipeline",
                "datapipeline:DescribeObjects",
                "datapipeline:EvaluateExpression",
                "datapipeline:GetPipelineDefinition",
                "datapipeline:PollForTask",
                "datapipeline:PutPipelineDefinition",
                "datapipeline:QueryObjects",
                "datapipeline:RemoveTags",
                "datapipeline:ReportTaskProgress",
                "datapipeline:ReportTaskRunnerHeartbeat",
                "datapipeline:SetStatus",
                "datapipeline:SetTaskStatus",
                "datapipeline:ValidatePipelineDefinition"
            ],
            "Resource": ["*"],
            "Condition": {
                "StringNotEquals": {"datapipeline:PipelineCreator":
 "${aws:userid}"}
            }
        }
    ]
}
```

Amazon DynamoDB: consente l'accesso a una tabella specifica

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo alla tabella `MyTable` di DynamoDB. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica,

sostituisci la politica *italicized placeholder text* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Important**
>
> Questa policy consente tutte le operazioni che possono essere eseguite su una tabella DynamoDB. Per esaminare queste operazioni, consulta Autorizzazioni API di DynamoDB: riferimento a operazioni, risorse e condizioni nella Guida per gli sviluppatori di Amazon DynamoDB. Puoi fornire le stesse autorizzazioni elencando ogni singola azione. Tuttavia, se utilizzi il carattere jolly (`*`) nell'elemento `Action`, ad esempio `"dynamodb:List*"`, non sarà necessario aggiornare la policy se DynamoDB aggiunge una nuova operazione Elenco.

Questa policy consente le operazioni solo sulle tabelle DynamoDB con il nome specificato. Per consentire ai tuoi utenti di `Read` accedere a tutto in DynamoDB, puoi anche allegare AmazonDynamoDBReadOnlyAccess AWS la policy gestita.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListAndDescribe",
            "Effect": "Allow",
            "Action": [
                "dynamodb:List*",
                "dynamodb:DescribeReservedCapacity*",
                "dynamodb:DescribeLimits",
                "dynamodb:DescribeTimeToLive"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SpecificTable",
            "Effect": "Allow",
            "Action": [
                "dynamodb:BatchGet*",
                "dynamodb:DescribeStream",
                "dynamodb:DescribeTable",
```

```
                "dynamodb:Get*",
                "dynamodb:Query",
                "dynamodb:Scan",
                "dynamodb:BatchWrite*",
                "dynamodb:CreateTable",
                "dynamodb:Delete*",
                "dynamodb:Update*",
                "dynamodb:PutItem"
            ],
            "Resource": "arn:aws:dynamodb:*:*:table/MyTable"
        }
    ]
}
```

Amazon DynamoDB: consente l'accesso ad attributi specifici

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso ad attributi DynamoDB specifici. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *italicized placeholder text* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Il requisito `dynamodb:Select` impedisce all'operazione API di restituire qualsiasi attributo per cui non si abbia il permesso, come ad esempio da una proiezione di indice. Per ulteriori informazioni sulle chiavi di condizione DynamoDB, consulta Specifica delle condizioni: uso delle chiavi di condizione nella Guida per gli sviluppatori di Amazon DynamoDB. Per ulteriori informazioni sulle condizioni multiple o sulle chiavi di condizioni multiple all'interno di un blocco di policy IAM `Condition`, consultare la pagina Valori multipli in una condizione

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:GetItem",
                "dynamodb:BatchGetItem",
                "dynamodb:Query",
```

```
                "dynamodb:PutItem",
                "dynamodb:UpdateItem",
                "dynamodb:DeleteItem",
                "dynamodb:BatchWriteItem"
            ],
            "Resource": ["arn:aws:dynamodb:*:*:table/table-name"],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:Attributes": [
                        "column-name-1",
                        "column-name-2",
                        "column-name-3"
                    ]
                },
                "StringEqualsIfExists": {"dynamodb:Select":
 "SPECIFIC_ATTRIBUTES"}
            }
        }
    ]
}
```

Amazon DynamoDB: consente l'accesso a livello di elemento a DynamoDB in base a un ID Amazon Cognito

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso a livello di elemento alla tabella DynamoDB `MyTable` in base all'ID utente di un pool di identità Amazon Cognito. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per utilizzare questa policy, devi strutturare la tabella DynamoDB in modo che l'ID utente del pool di identità Amazon Cognito costituisca la chiave di partizione. Per ulteriori informazioni, consulta Creazione di una tabella nella Guida per gli sviluppatori di Amazon DynamoDB.

Per ulteriori informazioni sulle chiavi di condizione DynamoDB, consulta Specifica delle condizioni: uso delle chiavi di condizione nella Guida per gli sviluppatori di Amazon DynamoDB.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:UpdateItem"
            ],
            "Resource": ["arn:aws:dynamodb:*:*:table/MyTable"],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:LeadingKeys": ["${cognito-
identity.amazonaws.com:sub}"]
                }
            }
        }
    ]
}
```

Amazon EC2: Collegare o distaccare volumi Amazon EBS alle istanze EC2 in base ai tag

Questo esempio mostra come creare una policy basata sull'identità che consenta ai proprietari di volumi EBS di collegare o scollegare i propri volumi EBS, definiti utilizzando il tag `VolumeUser`, alle istanze EC2 contrassegnate con tag come istanze di sviluppo (`Department=Development`). Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per ulteriori informazioni su come creare policy IAM per controllare l'accesso a risorse Amazon EC2, consulta Controllo dell'accesso alle risorse Amazon EC2 nella Guida per l'utente di Amazon EC2.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/Department": "Development"}
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/VolumeUser": "${aws:username}"}
            }
        }
    ]
}
```

Amazon EC2: consente l'avvio di istanze EC2 in una sottorete specifica, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta di elencare informazioni per tutti gli oggetti EC2 e di avviare istanze EC2 in una specifica sottorete. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci quella dell'*italicized placeholder text*esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:GetConsole*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/subnet-subnet-id",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        }
    ]
}
```

Amazon EC2: consente la gestione dei gruppi di sicurezza EC2 con una specifica coppia chiave-valore tag, in modo programmatico e nella console

In questo esempio viene illustrato come creare una policy basata sull'identità che conceda agli utenti l'autorizzazione a intraprendere determinate operazioni per i gruppi di sicurezza con lo stesso tag. Questa policy concede le autorizzazioni per visualizzare i gruppi di sicurezza nella console Amazon EC2, per aggiungere e rimuovere le regole in entrata e in uscita e per elencare e modificare le descrizioni delle regole per i gruppi di sicurezza esistenti con il tag `Department=Test`. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con

le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeSecurityGroups",
         "ec2:DescribeSecurityGroupRules",
         "ec2:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:AuthorizeSecurityGroupIngress",
         "ec2:RevokeSecurityGroupIngress",
         "ec2:AuthorizeSecurityGroupEgress",
         "ec2:RevokeSecurityGroupEgress",
         "ec2:ModifySecurityGroupRules",
         "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
         "ec2:UpdateSecurityGroupRuleDescriptionsEgress"
      ],
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:security-group/*"
      ],
      "Condition": {
         "StringEquals": {
            "aws:ResourceTag/Department": "Test"
         }
      }
     },
     {
      "Effect": "Allow",
      "Action": [
         "ec2:ModifySecurityGroupRules"
      ],
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:security-group-rule/*"
```

```
      ]
     }
   ]
}
```

Amazon EC2: consente l'avvio o l'arresto di istanze EC2 che un utente ha contrassegnato, a livello programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta a un utente IAM di avviare o arrestare le istanze EC2, ma solo se il 0`wner` del tag dell'istanza ha il valore del nome utente di quell'utente. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "${aws:username}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        }
    ]
}
```

EC2: avvia o interrompe le istanze in base ai tag

Questo esempio mostra come creare una policy basata sull'identità che consenta di avviare o arrestare le istanze con la coppia chiave-valore di tag `Project = DataAnalytics`, ma solo ai principali con la coppia chiave-valore di tag `Department = Data`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

La condizione nella policy restituisce true se entrambe le parti della condizione sono vere. L'istanza deve avere il tag `Project=DataAnalytics`. Inoltre, il principale IAM (utente o ruolo) da cui proviene la richiesta deve avere il tag `Department=Data`.

> **Note**
>
> Come best practice, collega policy con la chiave di condizione `aws:PrincipalTag` a gruppi IAM, nel caso in cui non tutti gli utenti dispongano del tag specificato.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "StartStopIfTags",
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Project": "DataAnalytics",
                    "aws:PrincipalTag/Department": "Data"
                }
            }
        }
```

```
    ]
}
```

EC2: avvio o arresto di istanze in base alla corrispondenza dei tag della risorsa e del principale

Questo esempio mostra come creare una policy basata sull'identità che consenta a un principale di avviare o interrompere un'istanza Amazon EC2 quando il tag della risorsa dell'istanza e il tag del principale hanno lo stesso valore per la chiave di tag `CostCenter`. Questa politica concede le autorizzazioni necessarie per completare questa azione in modo programmatico dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Note**
>
> Come best practice, collega policy con la chiave di condizione `aws:PrincipalTag` a gruppi IAM, nel caso in cui non tutti gli utenti dispongano del tag specificato.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "ec2:startInstances",
            "ec2:stopInstances"
        ],
        "Resource": "*",
        "Condition": {"StringEquals":
            {"aws:ResourceTag/CostCenter": "${aws:PrincipalTag/CostCenter}"}}
    }
}
```

Amazon EC2: consente l'accesso completo a EC2 entro una regione specifica, a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo a EC2 in una regione specifica. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *italicized placeholder text* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy. Per un elenco dei codici di regione, consulta Regioni disponibili nella Guida per l'utente di Amazon EC2.

In alternativa, puoi utilizzare la chiave di condizione `aws:RequestedRegion`, supportata da tutte le operazioni API di Amazon EC2. Per ulteriori informazioni, consulta Esempio: limitazione dell'accesso a una regione specifica nella Guida per l'utente di Amazon EC2.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": "ec2:*",
            "Resource": "*",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-2"
                }
            }
        }
    ]
}
```

Amazon EC2: consente l'avvio o l'arresto di un'istanza EC2 e la modifica di un gruppo di sicurezza, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta l'avvio o l'arresto di un'istanza EC2 specifica e la modifica di un determinato gruppo di sicurezza. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare

questa politica, sostituisci la politica *italicized placeholder text* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSecurityGroupReferences",
        "ec2:DescribeStaleSecurityGroups"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/i-instance-id",
        "arn:aws:ec2:*:*:security-group/sg-security-group-id"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Amazon EC2: richiede MFA (GetSessionToken) per operazioni EC2 specifiche

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo a tutte le operazioni AWS API in Amazon EC2. Tuttavia, rifiuta esplicitamente l'accesso alle operazioni API `StopInstances` e `TerminateInstances` se l'utente non viene autenticato

utilizzando l'autenticazione a più fattori (MFA, Multi-Factor Authentication). Per eseguire questa operazione a livello di programmazione, l'utente deve includere valori `SerialNumber` e `TokenCode` opzionali durante la chiamata all'operazione `GetSessionToken`. Questa operazione restituisce credenziali temporanee autenticate utilizzando MFA. Per saperne di più, consulta. GetSessionToken Richiesta di credenziali per gli utenti in ambienti non attendibili

Che cosa fa questa policy?

- L'istruzione `AllowAllActionsForEC2` consente tutte le operazioni Amazon EC2.
- La dichiarazione `DenyStopAndTerminateWhenMFAIsNotPresent` rifiuta le operazioni `TerminateInstances` e `StopInstances` quando manca il contesto MFA. Ciò significa che le operazioni vengono rifiutate quando manca il contesto dell'autenticazione a più fattori (ovvero, MFA non è stato utilizzato). Un rifiuto sostituisce il consenso.

> **Note**
>
> Il controllo della condizione per `MultiFactorAuthPresent` nella dichiarazione `Deny` non deve essere `{"Bool":{"aws:MultiFactorAuthPresent":false}}`, perché tale chiave non è presente e non può essere valutata quando MFA non viene utilizzato. Utilizzare invece il controllo `BoolIfExists` per vedere se la chiave è presente prima di controllare il valore. Per ulteriori informazioni, consulta ... IfExists operatori di condizione.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAllActionsForEC2",
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*"
        },
        {
            "Sid": "DenyStopAndTerminateWhenMFAIsNotPresent",
            "Effect": "Deny",
            "Action": [
                "ec2:StopInstances",
```

```
                "ec2:TerminateInstances"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {"aws:MultiFactorAuthPresent": false}
            }
        }
    ]
}
```

Amazon EC2: limita la chiusura delle istanze EC2 a un intervallo di indirizzi IP

Questo esempio mostra come creare una policy basata sull'identità che limiti le istanze EC2, consentendo l'operazione, ma negando esplicitamente l'accesso quando la richiesta proviene da un indirizzo IP esterno all'intervallo specificato. La policy è utile quando gli indirizzi IP per la tua azienda sono all'interno di determinati intervalli. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Se questa politica viene utilizzata in combinazione con altre politiche che consentono l'`ec2:TerminateInstances`azione (come la politica EC2 FullAccess AWS gestita da Amazon), l'accesso viene negato. Questo perché una dichiarazione di rifiuto esplicita prevale su una dichiarazione di consenso. Per ulteriori informazioni, consulta In che modo AWS la logica del codice di applicazione valuta le richieste per consentire o negare l'accesso.

> **Important**
>
> La chiave `aws:SourceIp` condizionale nega l'accesso a un utente Servizio AWS, ad esempio AWS CloudFormation, che effettua chiamate per conto dell'utente. Per ulteriori informazioni su come utilizzare la chiave di condizione `aws:SourceIp`, consultare AWS chiavi di contesto delle condizioni globali.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:TerminateInstances"],
            "Resource": ["*"]
        },
        {
            "Effect": "Deny",
            "Action": ["ec2:TerminateInstances"],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            },
            "Resource": ["*"]
        }
    ]
}
```

IAM: accesso all'API del simulatore di policy

Questo esempio mostra come creare una policy basata sull'identità che consenta l'utilizzo dell'API del simulatore di policy per le policy collegate a un utente, un gruppo o un ruolo nell' Account AWS corrente. Questa policy consente inoltre l'accesso per simulare le policy meno sensibili passate all'API come stringhe. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetContextKeysForCustomPolicy",
                "iam:GetContextKeysForPrincipalPolicy",
                "iam:SimulateCustomPolicy",
                "iam:SimulatePrincipalPolicy"
```

```
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Per consentire a un utente di accedere alla console del simulatore di policy per simulare le policy associate a un utente, gruppo o ruolo nel sistema corrente, consulta. Account AWSIAM: accesso alla console del simulatore di policy

IAM: accesso alla console del simulatore di policy

Questo esempio mostra come creare una policy basata sull'identità che consenta l'utilizzo della console del simulatore di policy per le policy collegate a un utente, un gruppo o un ruolo nell' Account AWS corrente.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "access-analyzer:ValidatePolicy",
                "iam:GetContextKeysForCustomPolicy",
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:GetUser",
                "iam:GetUserPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListAttachedRolePolicies",
                "iam:ListAttachedUserPolicies",
                "iam:ListGroups",
                "iam:ListGroupPolicies",
```

```
                "iam:ListGroupsForUser",
                "iam:ListPolicies",
                "iam:ListRolePolicies",
                "iam:ListRoles",
                "iam:ListUserPolicies",
                "iam:ListUsers",
                "iam:SimulateCustomPolicy",
                "iam:SimulatePrincipalPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

IAM: assumere ruoli che dispongono di un tag specifico

Questo esempio mostra come creare una policy basata sull'identità che consenta a un utente IAM di assumere ruoli con la coppia chiave-valore di tag `Project = ExampleCorpABC`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Se un ruolo con questo tag esiste nello stesso account dell'utente, l'utente può assumere tale ruolo. Se un ruolo con questo tag esiste in un account diverso da quello dell'utente, sono richieste autorizzazioni aggiuntive. La policy di attendibilità del ruolo tra account deve anche consentire all'utente o a tutti i membri dell'account dell'utente di assumere il ruolo. Per ulteriori informazioni sull'utilizzo di ruoli per l'accesso tra account, consulta Accesso per un utente IAM in un altro Account AWS che possiedi.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AssumeTaggedRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {"iam:ResourceTag/Project": "ExampleCorpABC"}
            }
        }
    ]
}
```

IAM: consente e rifiuta l'accesso a più servizi a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo a diversi servizi e l'accesso autonomo limitato in IAM. Rifiuta inoltre l'accesso al bucket `logs` di Amazon S3 o all'istanza `i-1234567890abcdef0` Amazon EC2. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Warning**
>
> Questa policy consente l'accesso completo a tutte le operazioni e risorse in più servizi. Questa policy deve essere applicata solo ad amministratori fidati.

Puoi usare questa policy come un limite delle autorizzazioni per definire il numero massimo di autorizzazioni che una policy basata su identità può concedere a un utente IAM. Per ulteriori informazioni, consulta Delega di responsabilità ad altri mediante i limiti delle autorizzazioni. Quando la policy viene usata come un limite delle autorizzazioni per un utente, le dichiarazioni definiscono i seguenti limiti:

- L'istruzione `AllowServices` consente l'accesso completo ai servizi AWS specificati. Ciò significa che le operazioni dell'utente in questi servizi sono limitate solo dalle policy di autorizzazioni collegate all'utente.
- La dichiarazione `AllowIAMConsoleForCredentials` consente l'accesso per elencare tutti gli utenti IAM. Questo accesso è necessario per navigare nella pagina Users (Utenti) nella Console di gestione AWS. Inoltre, consente di visualizzare i requisiti associati alle password per l'account, operazione necessaria per permettere all'utente di modificare la sua password.

- L'istruzione `AllowManageOwnPasswordAndAccessKeys` consente agli utenti di gestire solo le proprie chiavi di accesso programmatiche e password della console. Questo è importante perché se un'altra policy offre a un utente l'accesso IAM completo, tale utente può modificare le sue autorizzazioni o quelle di altri utenti. Questa istruzione impedisce che ciò si verifichi.
- L'istruzione `DenyS3Logs` nega esplicitamente l'accesso al bucket `logs`. Questa policy applica limitazioni aziendali all'utente.
- L'istruzione `DenyEC2Production` nega esplicitamente l'accesso all'istanza `i-1234567890abcdef0`.

Questa policy non consente l'accesso ad altri servizi o operazioni. Quando la politica viene utilizzata come limite di autorizzazioni per un utente, anche se altre politiche allegate all'utente consentono tali azioni, AWS rifiuta la richiesta.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowServices",
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "cloudwatch:*",
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowIAMConsoleForCredentials",
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers",
                "iam:GetAccountPasswordPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnPasswordAndAccessKeys",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:*AccessKey*",
                "iam:ChangePassword",
                "iam:GetUser",
                "iam:*LoginProfile*"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "DenyS3Logs",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::logs",
                "arn:aws:s3:::logs/*"
            ]
        },
        {
            "Sid": "DenyEC2Production",
            "Effect": "Deny",
            "Action": "ec2:*",
            "Resource": "arn:aws:ec2:*:*:instance/i-1234567890abcdef0"
        }
    ]
}
```

IAM: aggiunta di un tag specifico a un utente con un determinato tag

Questo esempio mostra come creare una policy basata sull'identità che consenta di aggiungere la chiave di tag `Department` con i valori di tag `Marketing`, `Development` o `QualityAssurance` a un utente IAM. Tale utente deve già includere la coppia chiave-valore di tag `JobFunction = Manager`. È possibile usare questa policy per richiedere che un responsabile appartenga solo a uno dei tre reparti. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

L'istruzione `ListTagsForAllUsers` consente di visualizzare i tag per tutti gli utenti nell'account.

La prima condizione nell'istruzione `TagManagerWithSpecificDepartment` utilizza l'operatore di condizione `StringEquals`. La condizione restituisce true se entrambe le parti della condizione sono

vere. L'utente che necessita di tag deve già disporre del tag `JobFunction=Manager`. La richiesta deve includere la chiave di tag `Department` con uno dei valori di tag elencati.

La seconda condizione utilizza l'operatore di condizione `ForAllValues:StringEquals`. La condizione restituisce true se tutte le chiavi di tag nella richiesta corrispondono alla chiave nella policy. Ciò significa che l'unica chiave di tag nella richiesta deve essere `Department`. Per ulteriori informazioni sull'utilizzo di `ForAllValues`, consultare Operatori dell'insieme per le chiavi di contesto multivalore.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListTagsForAllUsers",
            "Effect": "Allow",
            "Action": [
                "iam:ListUserTags",
                "iam:ListUsers"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TagManagerWithSpecificDepartment",
            "Effect": "Allow",
            "Action": "iam:TagUser",
            "Resource": "*",
            "Condition": {"StringEquals": {
                "iam:ResourceTag/JobFunction": "Manager",
                "aws:RequestTag/Department": [
                    "Marketing",
                    "Development",
                    "QualityAssurance"
                    ]
                },
                "ForAllValues:StringEquals": {"aws:TagKeys": "Department"}
            }
        }
    ]
}
```

IAM: aggiunta di un determinato tag con valori specifici

Questo esempio mostra come creare una policy basata sull'identità che consenta di aggiungere solo la chiave di tag `CostCenter` e il valore di tag `A-123` o il valore di tag `B-456` a qualsiasi ruolo o utente IAM. Puoi utilizzare questa policy per limitare il tagging a una chiave di tag e a un set di valori di tag specifici. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

L'istruzione `ConsoleDisplay` consente di visualizzare i tag per tutti gli utenti e i ruoli nell'account.

La prima condizione nell'istruzione `AddTag` utilizza l'operatore di condizione `StringEquals`. La condizione restituisce true se la richiesta include la chiave di tag `CostCenter` con uno dei valori di tag elencati.

La seconda condizione utilizza l'operatore di condizione `ForAllValues:StringEquals`. La condizione restituisce true se tutte le chiavi di tag nella richiesta corrispondono alla chiave nella policy. Ciò significa che l'unica chiave di tag nella richiesta deve essere `CostCenter`. Per ulteriori informazioni sull'utilizzo di `ForAllValues`, consultare Operatori dell'insieme per le chiavi di contesto multivalore.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ConsoleDisplay",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:GetUser",
                "iam:ListRoles",
                "iam:ListRoleTags",
                "iam:ListUsers",
                "iam:ListUserTags"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "AddTag",
            "Effect": "Allow",
            "Action": [
                "iam:TagUser",
                "iam:TagRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CostCenter": [
                        "A-123",
                        "B-456"
                    ]
                },
                "ForAllValues:StringEquals": {"aws:TagKeys": "CostCenter"}
            }
        }
    ]
}
```

IAM: creazione di nuovi utenti solo con tag specifici

Questo esempio mostra come creare una policy basata sull'identità che consenta la creazione di utenti IAM, ma solo con una o entrambe le chiavi di tag `Department` e `JobFunction`. La chiave di tag `Department` deve avere il valore di tag `Development` o `QualityAssurance`. La chiave di tag `JobFunction` deve avere il valore di tag `Employee`. È possibile usare questa policy per richiedere che i nuovi utenti dispongano di una mansione e un reparto specifici. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

La prima condizione nell'istruzione utilizza l'operatore di condizione `StringEqualsIfExists`. Se un tag con la chiave `Department` o `JobFunction` è presente nella richiesta, il tag deve avere il valore specificato. Se non è presente alcuna chiave, questa condizione viene valutata come true. La condizione viene valutata come false solo se una delle chiavi di condizione specificate è presente nella richiesta, ma ha un valore diverso da quelli consentiti. Per ulteriori informazioni sull'utilizzo di `IfExists`, consultare ... IfExists operatori di condizione.

La seconda condizione utilizza l'operatore di condizione `ForAllValues:StringEquals`. La condizione restituisce true se si verifica una corrispondenza tra ognuna delle chiavi di tag specificate nella richiesta e almeno un valore nella policy. Ciò significa che tutti i tag nella richiesta devono essere in questo elenco. Tuttavia, la richiesta può includere solo uno dei tag nell'elenco. Ad esempio, puoi creare un utente IAM con il solo tag `Department=QualityAssurance`. Tuttavia, non puoi creare un utente IAM con il tag `JobFunction=Employee` e il tag `Project=core`. Per ulteriori informazioni sull'utilizzo di `ForAllValues`, consultare Operatori dell'insieme per le chiavi di contesto multivalore.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "TagUsersWithOnlyTheseTags",
            "Effect": "Allow",
            "Action": [
                "iam:CreateUser",
                "iam:TagUser"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIfExists": {
                    "aws:RequestTag/Department": [
                        "Development",
                        "QualityAssurance"
                    ],
                    "aws:RequestTag/JobFunction": "Employee"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "Department",
                        "JobFunction"
                    ]
                }
            }
        }
    ]
}
```

IAM: generazione e recupero di report di credenziali IAM

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta agli utenti di generare e scaricare un report che elenca tutti gli utenti IAM del loro gruppo. Account AWS Il report include lo stato delle credenziali dell'utente, incluse le password, le chiavi di accesso, i dispositivi MFA e i certificati di firma. Questa policy concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI

Per ulteriori informazioni sui report delle credenziali, consultare la pagina Genera report sulle credenziali per i tuoi Account AWS.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GenerateCredentialReport",
            "iam:GetCredentialReport"
        ],
        "Resource": "*"
    }
}
```

IAM: consente di gestire l'appartenenza di un gruppo a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta di aggiornare l'appartenenza al gruppo denominato `MarketingTeam`. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Che cosa fa questa policy?

- La `ViewGroups` dichiarazione consente all'utente di elencare tutti gli utenti e i gruppi nella Console di gestione AWS. Inoltre, consente all'utente di visualizzare informazioni di base sugli utenti nell'account. Queste autorizzazioni devono essere nella propria istruzione perché non supportano o

non devono specificare l'ARN di una risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`.

- La dichiarazione `ViewEditThisGroup` consente all'utente di visualizzare le informazioni sul gruppo `MarketingTeam` e aggiungere o rimuovere utenti da tale gruppo.

Questa policy non consente all'utente di visualizzare o modificare le autorizzazioni degli utenti o del gruppo `MarketingTeam`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewGroups",
            "Effect": "Allow",
            "Action": [
                "iam:ListGroups",
                "iam:ListUsers",
                "iam:GetUser",
                "iam:ListGroupsForUser"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewEditThisGroup",
            "Effect": "Allow",
            "Action": [
                "iam:AddUserToGroup",
                "iam:RemoveUserFromGroup",
                "iam:GetGroup"
            ],
            "Resource": "arn:aws:iam::*:group/MarketingTeam"
        }
    ]
}
```

IAM: gestione di un tag specifico

Questo esempio mostra come creare una policy basata sull'identità che consenta di aggiungere e rimuovere il tag IAM con la chiave di tag `Department` dalle entità IAM (utenti e ruoli). Questa policy non limita il valore del tag `Department`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:TagUser",
            "iam:TagRole",
            "iam:UntagUser",
            "iam:UntagRole"

        ],
        "Resource": "*",
        "Condition": {"ForAllValues:StringEquals": {"aws:TagKeys": "Department"}}
    }
}
```

IAM: passa un ruolo IAM a un AWS servizio specifico

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta di passare qualsiasi ruolo del servizio IAM al servizio Amazon. CloudWatch Questa politica concede le autorizzazioni necessarie per completare questa azione in modo programmatico dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Un ruolo di servizio è un ruolo IAM che specifica un AWS servizio come principale che può assumere il ruolo. Questo consente al servizio di assumere il ruolo e accedere a risorse in altri servizi a tuo nome. Per consentire CloudWatch ad Amazon di assumere il ruolo che conferisci, devi specificare il

responsabile del `cloudwatch.amazonaws.com` servizio come responsabile nella politica di fiducia del tuo ruolo. Il principale del servizio è definito dal servizio. Per ulteriori informazioni sul principale del servizio per un servizio, consultare la documentazione per quel servizio. Per alcuni servizi, consulta AWS servizi supportati da IAM e cerca i servizi che hanno Sì nella colonna Ruolo collegato ai servizi. Scegliere Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio. Cerca `amazonaws.com` per visualizzare il principale del servizio.

Per ulteriori informazioni sul passaggio di un ruolo del servizio al servizio, consulta Concedi a un utente le autorizzazioni per passare un ruolo a un AWS service.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {"iam:PassedToService":
 "cloudwatch.amazonaws.com"}
            }
        }
    ]
}
```

IAM: consente l'accesso in sola lettura alla console IAM senza la creazione di report

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di eseguire qualsiasi operazione IAM che inizia con la stringa `Get` o `List`. Quando gli utenti utilizzano la console, la console effettua richieste a IAM per elencare gruppi, utenti, ruoli e policy e generare report su tali risorse.

L'asterisco funge da carattere jolly. Quando utilizzi `iam:Get*` in una policy, le autorizzazioni risultanti includono tutte le operazioni IAM che iniziano con `Get`, ad esempio `GetUser` e `GetRole`. I caratteri jolly sono utili quando nuovi tipi di entità vengono aggiunti a IAM in futuro. In tal caso, le

autorizzazioni concesse dalla policy consentono automaticamente all'utente di elencare e ottenere i dettagli su queste nuove entità.

Questa policy non può essere utilizzata per generare report o dettagli dell'ultimo accesso al servizio. Per una policy diversa che lo consenta, consulta IAM: consente l'accesso in sola lettura alla console IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:Get*",
            "iam:List*"
        ],
        "Resource": "*"
    }
}
```

IAM: consente l'accesso in sola lettura alla console IAM

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di eseguire qualsiasi operazione IAM che inizia con la stringa `Get`, `List` o `Generate`. Quando gli utenti utilizzano la console IAM, la console effettua richieste per elencare gruppi, utenti, ruoli e policy e generare report su tali risorse.

L'asterisco funge da carattere jolly. Quando utilizzi `iam:Get*` in una policy, le autorizzazioni risultanti includono tutte le operazioni IAM che iniziano con `Get`, ad esempio `GetUser` e `GetRole`. L'uso di un carattere jolly è utile, in particolare se in futuro vengono aggiunti nuovi tipi di entità a IAM. In tal caso, le autorizzazioni concesse dalla policy consentono automaticamente all'utente di elencare e ottenere i dettagli su queste nuove entità.

Utilizza questa policy per l'accesso alla console che include le autorizzazioni per generare report o i dettagli dell'ultimo accesso al servizio. Per una policy diversa che non consenta la generazione di operazioni, consulta IAM: consente l'accesso in sola lettura alla console IAM senza la creazione di report.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:Get*",
            "iam:List*",
            "iam:Generate*"
        ],
        "Resource": "*"
    }
}
```

IAM: consente a utenti IAM specifici di gestire un gruppo a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta a specifici utenti IAM di gestire il gruppo `AllUsers`. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Che cosa fa questa policy?

- L'istruzione `AllowAllUsersToListAllGroups` consente di elencare tutti i gruppi. Questa operazione è necessaria per l'accesso alla console. Questa autorizzazione deve trovarsi nella propria dichiarazione perché non supporta un ARN della risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`.
- L'istruzione `AllowAllUsersToViewAndManageThisGroup` consente tutte le operazioni del gruppo che possono essere eseguite sul tipo di risorsa del gruppo. Non consente l'operazione `ListGroupsForUser`, che può essere eseguita su un tipo di risorsa dell'utente e non un tipo di risorsa del gruppo. Per ulteriori informazioni sui tipi di risorsa che è possibile specificare per un'operazione IAM, consulta Operazioni, risorse e chiavi di condizione per AWS Identity and Access Management.
- L'`LimitGroupManagementAccessToSpecificUsers`istruzione nega agli utenti con i nomi specificati l'accesso alle azioni di gruppo di scrittura e gestione delle autorizzazioni.

Quando un utente specificato nella policy tenta di apportare modifiche al gruppo, questa istruzione non rifiuta la richiesta. Questa richiesta è consentita dall'istruzione `AllowAllUsersToViewAndManageThisGroup`. Se altri utenti tentano di eseguire queste operazioni, la richiesta viene rifiutata. Puoi visualizzare le operazioni IAM che vengono definite con i livelli di accesso Scrittura o Gestione delle autorizzazioni durante la creazione di questa policy nella console IAM. A tale scopo, passare dalla scheda JSON alla scheda Visual editor. Per ulteriori informazioni sui livelli di accesso, vedere Azioni, risorse e chiavi di condizione per. AWS Identity and Access Management

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAllUsersToListAllGroups",
            "Effect": "Allow",
            "Action": "iam:ListGroups",
            "Resource": "*"
        },
        {
            "Sid": "AllowAllUsersToViewAndManageThisGroup",
            "Effect": "Allow",
            "Action": "iam:*Group*",
            "Resource": "arn:aws:iam::*:group/AllUsers"
        },
        {
            "Sid": "LimitGroupManagementAccessToSpecificUsers",
            "Effect": "Deny",
            "Action": [
                "iam:AddUserToGroup",
                "iam:CreateGroup",
                "iam:RemoveUserFromGroup",
                "iam:DeleteGroup",
                "iam:AttachGroupPolicy",
                "iam:UpdateGroup",
                "iam:DetachGroupPolicy",
                "iam:DeleteGroupPolicy",
                "iam:PutGroupPolicy"
            ],
            "Resource": "arn:aws:iam::*:group/AllUsers",
```

```
            "Condition": {
                "StringNotEquals": {
                    "aws:username": [
                        "srodriguez",
                        "mjackson",
                        "adesai"
                    ]
                }
            }
        }
    ]
}
```

IAM: consente l'impostazione dei requisiti della password dell'account a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta a un utente di visualizzare e aggiornare i requisiti della password dell'account. I requisiti della password specificano i requisiti di complessità e i periodi di rotazione obbligatori per le password dei membri dell'account. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

Per Scopri come impostare la policy dei requisiti della password dell'account per l'account, consulta Configurare una policy delle password di un account per gli utenti IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GetAccountPasswordPolicy",
            "iam:UpdateAccountPasswordPolicy"
        ],
        "Resource": "*"
    }
}
```

IAM: accesso all'API simulatore di policy basata sul percorso degli utenti

Questo esempio mostra come creare una policy basata sull'identità che consenta l'utilizzo dell'API del simulatore di policy solo da parte degli utenti con il percorso `Department/Development`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetContextKeysForPrincipalPolicy",
                "iam:SimulatePrincipalPolicy"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:user/Department/Development/*"
        }
    ]
}
```

> **Note**
>
> Per creare una policy che consente di utilizzare la console del simulatore di policy per gli utenti con percorso `Department/Development`, consultare IAM: accesso alla console del simulatore di policy in base al percorso utente.

IAM: accesso alla console del simulatore di policy in base al percorso utente

Questo esempio mostra come creare una policy basata sull'identità che consenta l'utilizzo della console del simulatore di policy solo per gli utenti con il percorso `Department/Development`. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con

le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetPolicy",
                "iam:GetUserPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:GetUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListGroupsForUser",
                "iam:ListUserPolicies",
                "iam:ListUsers"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:user/Department/Development/*"
        }
    ]
}
```

IAM: consente agli utenti IAM di gestire in modo autonomo un dispositivo MFA

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di gestire in modo automatico il proprio dispositivo di autenticazione a più fattori (MFA). Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI

> **Note**
>
> Se un utente IAM con questa policy non è autenticato tramite MFA, questa policy nega l'accesso a tutte le AWS azioni tranne quelle necessarie per l'autenticazione tramite MFA. Se aggiungi queste autorizzazioni per un utente che ha effettuato l'accesso AWS, potrebbe dover disconnettersi e riconnettersi per visualizzare le modifiche.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListActions",
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers",
                "iam:ListVirtualMFADevices"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUserToCreateVirtualMFADevice",
            "Effect": "Allow",
            "Action": [
                "iam:CreateVirtualMFADevice"
            ],
            "Resource": "arn:aws:iam::*:mfa/*"
        },
        {
            "Sid": "AllowUserToManageTheirOwnMFA",
            "Effect": "Allow",
            "Action": [
                "iam:EnableMFADevice",
                "iam:GetMFADevice",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
```

```
        {
            "Sid": "AllowUserToDeactivateTheirOwnMFAOnlyWhenUsingMFA",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice"
            ],
            "Resource": [
                "arn:aws:iam::*:user/${aws:username}"
            ],
            "Condition": {
                "Bool": {
                    "aws:MultiFactorAuthPresent": "true"
                }
            }
        },
        {
            "Sid": "BlockMostAccessUnlessSignedInWithMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:CreateVirtualMFADevice",
                "iam:EnableMFADevice",
                "iam:ListMFADevices",
                "iam:ListUsers",
                "iam:ListVirtualMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
    ]
}
```

IAM: consente agli utenti IAM di aggiornare le credenziali a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di aggiornare le proprie chiavi di accesso, i certificati di firma, le credenziali specifiche del servizio e le

password. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers",
                "iam:GetAccountPasswordPolicy"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:*AccessKey*",
                "iam:ChangePassword",
                "iam:GetUser",
                "iam:*ServiceSpecificCredential*",
                "iam:*SigningCertificate*"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        }
    ]
}
```

Per ulteriori informazioni su come un utente può modificare la propria password nella console, consultare the section called “Come un utente IAM può modificare la propria password”.

IAM: visualizza le informazioni sull'ultimo accesso al servizio per una AWS Organizations policy

Questo esempio mostra come creare una policy basata sull’identità che consenta di visualizzare le informazioni sull’ultimo accesso per una determinata policy di AWS Organizations . Questa policy consente di recuperare dati per la policy di controllo del servizio (SCP) con l'ID `p-policy123`. La persona che genera e visualizza il report deve essere autenticata utilizzando le credenziali

dell'account di AWS Organizations gestione. Questa politica consente al richiedente di recuperare i dati di qualsiasi AWS Organizations entità della propria organizzazione. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *italicized placeholder text* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per informazioni importanti sui dati sull'ultimo accesso al servizio, incluse le autorizzazioni richieste, la risoluzione dei problemi e le regioni supportate, consulta Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOrgsReadOnlyAndIamGetReport",
            "Effect": "Allow",
            "Action": [
                "iam:GetOrganizationsAccessReport",
                "organizations:Describe*",
                "organizations:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowGenerateReportOnlyForThePolicy",
            "Effect": "Allow",
            "Action": "iam:GenerateOrganizationsAccessReport",
            "Resource": "*",
            "Condition": {
                "StringEquals": {"iam:OrganizationsPolicyId": "p-policy123"}
            }
        }
    ]
}
```

IAM: limita le policy gestite che possono essere applicate a un utente, un gruppo o un ruolo IAM.

Questo esempio mostra come è possibile creare una policy basata sull'identità che limiti le policy gestite dai clienti e quelle AWS gestite che possono essere applicate a un utente, gruppo o ruolo IAM. Questa policy concede le autorizzazioni necessarie per completare questa azione in modo programmatico dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *italicized placeholder text* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:AttachUserPolicy",
            "iam:DetachUserPolicy"
        ],
        "Resource": "*",
        "Condition": {
            "ArnEquals": {
                "iam:PolicyARN": [
                    "arn:aws:iam::*:policy/policy-name-1",
                    "arn:aws:iam::*:policy/policy-name-2"
                ]
            }
        }
    }
}
```

AWS: nega l'accesso alle risorse esterne al tuo account ad eccezione delle politiche IAM AWS gestite

L'utilizzo di `aws:ResourceAccount` nelle policy basate sull'identità può influire sulla capacità dell'utente o del ruolo di utilizzare alcuni servizi che richiedono l'interazione con le risorse negli account di proprietà di un servizio.

Puoi creare una policy con un'eccezione per consentire le policy IAM AWS gestite. Un account gestito dal servizio esterno alle tue politiche AWS Organizations IAM gestite. Esistono quattro azioni IAM che elencano e AWS recuperano le politiche gestite. Utilizza queste operazioni nell'elemento `NotAction` dell'istruzione `AllowAccessToResourcesInSpecificAccountsAndSpecificService1` nella policy.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToResourcesInSpecificAccountsAndSpecificService1",
      "Effect": "Deny",
      "NotAction": [
        "iam:GetPolicy",
        "iam:GetPolicyVersion",
        "iam:ListEntitiesForPolicy",
        "iam:ListPolicies"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "111122223333"
          ]
        }
      }
    }
  ]
}
```

AWS Lambda: consente a una funzione Lambda di accedere a una tabella Amazon DynamoDB

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso in lettura e scrittura a una tabella Amazon DynamoDB specifica. La politica consente anche di scrivere file di registro CloudWatch nei registri. Per utilizzare questa politica, sostituisci la politica *italicized*

placeholder text nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per utilizzare questa policy, collega la policy a un ruolo del servizio Lambda. Un ruolo del servizio è un ruolo che viene creato nell'account per consentire a un servizio di eseguire operazioni a tuo nome. Tale ruolo di servizio deve essere incluso AWS Lambda come principale nella politica di fiducia. Per informazioni dettagliate su come utilizzare questa policy, consulta Come creare una policy AWS IAM per concedere AWS Lambda l'accesso a una tabella Amazon DynamoDB nel AWS Security Blog.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteTable",
            "Effect": "Allow",
            "Action": [
                "dynamodb:BatchGetItem",
                "dynamodb:GetItem",
                "dynamodb:Query",
                "dynamodb:Scan",
                "dynamodb:BatchWriteItem",
                "dynamodb:PutItem",
                "dynamodb:UpdateItem"
            ],
            "Resource": "arn:aws:dynamodb:*:*:table/SampleTable"
        },
        {
            "Sid": "GetStreamRecords",
            "Effect": "Allow",
            "Action": "dynamodb:GetRecords",
            "Resource": "arn:aws:dynamodb:*:*:table/SampleTable/stream/* "
        },
        {
            "Sid": "WriteLogStreamsAndGroups",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "CreateLogGroup",
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "*"
        }
    ]
}
```

Amazon RDS: consente l'accesso completo al database RDS in una regione specifica

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo al database RDS in una regione specifica. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *italicized placeholder text* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rds:*",
            "Resource": ["arn:aws:rds:us-east-1:*:*"]
        },
        {
            "Effect": "Allow",
            "Action": ["rds:Describe*"],
            "Resource": ["*"]
        }
    ]
}
```

Amazon RDS: consente il ripristino dei database RDS, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta di ripristinare i database RDS. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "rds:CreateDBParameterGroup",
                "rds:CreateDBSnapshot",
                "rds:DeleteDBSnapshot",
                "rds:Describe*",
                "rds:DownloadDBLogFilePortion",
                "rds:List*",
                "rds:ModifyDBInstance",
                "rds:ModifyDBParameterGroup",
                "rds:ModifyOptionGroup",
                "rds:RebootDBInstance",
                "rds:RestoreDBInstanceFromDBSnapshot",
                "rds:RestoreDBInstanceToPointInTime"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon RDS: consente ai proprietari di tag l'accesso completo alle risorse RDS da loro contrassegnate con un tag

Questo esempio mostra come creare una policy basata sull'identità che conceda ai proprietari dei tag l'accesso completo alle risorse RDS che hanno contrassegnato con tag. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "rds:Describe*",
                "rds:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "rds:DeleteDBInstance",
                "rds:RebootDBInstance",
                "rds:ModifyDBInstance"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:db-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyOptionGroup",
                "rds:DeleteOptionGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:og-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyDBParameterGroup",
                "rds:ResetDBParameterGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
```

```
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:pg-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:AuthorizeDBSecurityGroupIngress",
                "rds:RevokeDBSecurityGroupIngress",
                "rds:DeleteDBSecurityGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:secgrp-tag/Owner":
"${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:DeleteDBSnapshot",
                "rds:RestoreDBInstanceFromDBSnapshot"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:snapshot-tag/Owner":
"${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyDBSubnetGroup",
                "rds:DeleteDBSubnetGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:subgrp-tag/Owner":
"${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyEventSubscription",
```

```
                "rds:AddSourceIdentifierToSubscription",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:DeleteEventSubscription"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:es-tag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Amazon S3: consente agli utenti di Amazon Cognito di accedere a oggetti nel relativo bucket

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti Amazon Cognito di accedere a oggetti in un determinato bucket Amazon S3. Questa policy consente l'accesso solo agli oggetti con un nome che include `cognito`, il nome dell'applicazione e l'ID del principale federato, rappresentati dalla variabile ${cognito-identity.amazonaws.com:sub}. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Note**
>
> Il valore 'sub' utilizzato nella chiave dell'oggetto non è il valore secondario dell'utente nel pool di utenti, è l'ID identità associato all'utente nel pool di identità.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ListYourObjects",
      "Effect": "Allow",
```

```
      "Action": "s3:ListBucket",
      "Resource": [
        "arn:aws:s3:::bucket-name"
      ],
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "cognito/application-name/${cognito-identity.amazonaws.com:sub}/*"
          ]
        }
      }
    },
    {
      "Sid": "ReadWriteDeleteYourObjects",
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name/cognito/application-name/${cognito-
identity.amazonaws.com:sub}/*"
      ]
    }
  ]
}
```

Amazon Cognito fornisce autenticazione, autorizzazione e gestione degli utenti per le app Web e per dispositivi mobili. Gli utenti possono accedere direttamente con un nome utente e una password, oppure tramite terze parti , ad esempio Facebook, Amazon o Google.

I due componenti principali di Amazon Cognito sono i bacini d'utenza e i pool di identità. I bacini d'utenza sono directory utente che forniscono opzioni di registrazione e di accesso agli utenti delle tue app. I pool di identità consentono di concedere agli utenti l'accesso ad altri AWS servizi. È possibile usare i pool di identità e i bacini d'utenza separatamente o insieme.

Per ulteriori informazioni su Amazon Cognito, consulta la Guida per l'utente di Amazon Cognito.

Amazon S3: consente agli utenti federati di accedere alla propria directory home Amazon S3, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta ai principali federati di accedere all'oggetto bucket nella loro directory home in S3. La directory home è un bucket che include una cartella home e le cartelle per i singoli principali federati. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

La variabile `${aws:userid}` in questa policy restituisce `role-id:specified-name`. La parte `role-id` dell'ID principale federato è un identificatore univoco assegnato al ruolo del principale federato durante la creazione. Per ulteriori informazioni, consulta Identificatori univoci. `specified-name`È il RoleSessionName parametro passato alla `AssumeRoleWithWebIdentity` richiesta quando il principale federato ha assunto il proprio ruolo.

È possibile visualizzare l'ID del ruolo utilizzando il AWS CLI comando`aws iam get-role --role-name` *`specified-name`*. Ad esempio, supponiamo di specificare il nome descrittivo `John` e che la CLI restituisca l'ID ruolo `AROAXXT2NJT7D3SIQN7Z6`. In questo caso, l'ID principale federato è `AROAXXT2NJT7D3SIQN7Z6:John`. Questa policy quindi consente al principale federato John di accedere ai bucket Amazon S3 con il prefisso `AROAXXT2NJT7D3SIQN7Z6:John`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "S3ConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "",
                        "home/",
                        "home/${aws:userid}/*"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/home/${aws:userid}",
                "arn:aws:s3:::amzn-s3-demo-bucket/home/${aws:userid}/*"
            ]
        }
    ]
}
```

Amazon S3: accesso al bucket S3, ma bucket di produzione rifiutato senza MFA recente

Questo esempio mostra come creare una policy basata sull'identità che consenta a un amministratore Amazon S3 di accedere a qualsiasi bucket, inclusi l'aggiornamento, l'aggiunta e l'eliminazione di oggetti. Tuttavia, rifiuta esplicitamente l'accesso al bucket `amzn-s3-demo-bucket-production` se l'utente non ha effettuato l'accesso utilizzando l'autenticazione multi-fattore (MFA) negli ultimi 30 minuti. Questo criterio concede le autorizzazioni necessarie per eseguire questa azione nella console o a livello di codice utilizzando l'API o. AWS CLI AWS Per utilizzare questa politica, sostituisci la politica dell'*`italicized placeholder text`*esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Questa policy non consente mai l'accesso a livello di programmazione al bucket `amzn-s3-demo-bucket` utilizzando le chiavi di accesso degli utenti a lungo termine. Questa operazione viene

eseguita utilizzando la chiave di condizione `aws:MultiFactorAuthAge` con l'operatore di condizione `NumericGreaterThanIfExists`. Questa condizione di policy restituisce `true` se MFA non è presente o se l'età di MFA è superiore a 30 minuti. In tali situazioni, l'accesso è negato. Per accedere al `amzn-s3-demo-bucket-production` bucket a livello di codice, l'amministratore S3 deve utilizzare credenziali temporanee generate negli ultimi 30 minuti utilizzando l'operazione API. GetSessionToken

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListAllS3Buckets",
            "Effect": "Allow",
            "Action": ["s3:ListAllMyBuckets"],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowBucketLevelActions",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowBucketObjectActions",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:s3:::*/*"
        },
        {
            "Sid": "RequireMFAForProductionBucket",
            "Effect": "Deny",
```

```
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket-production/*",
                "arn:aws:s3:::amzn-s3-demo-bucket-production"
            ],
            "Condition": {
                "NumericGreaterThanIfExists": {"aws:MultiFactorAuthAge": "1800"}
            }
        }
    ]
}
```

Amazon S3: consente agli utenti IAM di accedere alla propria directory home S3, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di accedere al proprio oggetto del bucket nella directory home in S3. La home directory è un bucket che include una cartella home e le cartelle per i singoli utenti. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Questa policy non funziona quando si utilizzano i ruoli IAM perché la variabile `aws:username` non è disponibile quando si utilizzano i ruoli IAM. Per informazioni dettagliate sui valori delle chiavi principali, consulta Valori della chiave dell'entità principale.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "S3ConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPublicAccessBlock",
```

```
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "",
                        "home/",
                        "home/${aws:username}/*"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/home/${aws:username}",
                "arn:aws:s3:::amzn-s3-demo-bucket/home/${aws:username}/*"
            ]
        }
    ]
}
```

Amazon S3: limitazione della gestione a un bucket S3 specifico

Questo esempio mostra come creare una policy basata sull'identità che limiti la gestione di un bucket Amazon S3 a quel determinato bucket. Questa policy concede l'autorizzazione a eseguire tutte le operazioni di Amazon S3, ma nega l'accesso a ogni Servizio AWS tranne Amazon S3. Guarda l'esempio seguente. In base a questa policy, puoi accedere solo alle operazioni di Amazon S3 che è possibile eseguire su un bucket S3 o una risorsa oggetto S3. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *italicized placeholder text* nell'esempio

con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Se questa politica viene utilizzata in combinazione con altre politiche (come le politiche EC2 FullAccess AWS gestite da AmazonS3 o FullAccess Amazon) che consentono azioni negate da questa politica, l'accesso viene negato. Questo perché una dichiarazione di rifiuto esplicita prevale su una dichiarazione di consenso. Per ulteriori informazioni, consulta In che modo AWS la logica del codice di applicazione valuta le richieste per consentire o negare l'accesso.

> **Warning**
>
> `NotAction` e `NotResource` sono elementi di policy avanzate da utilizzare con attenzione. Questa policy rifiuta l'accesso a qualsiasi servizio AWS a eccezione di Amazon S3. Se colleghi questa policy a un utente, qualsiasi altra policy che concede le autorizzazioni ad altri servizi viene ignorata e l'accesso viene negato.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
            "Effect": "Deny",
            "NotAction": "s3:*",
            "NotResource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

Concedere l'accesso in lettura e scrittura agli oggetti di un bucket Amazon S3

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso `Read` e `Write` agli oggetti in un bucket Amazon S3 specifico. Questa policy concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

L'operazione `s3:*Object` usa un carattere jolly nel nome dell'operazione. L'istruzione `AllObjectActions` consente le operazioni `GetObject`, `DeleteObject`, `PutObject` e qualsiasi altra operazione Amazon S3 che termina con la parola "Object".

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": ["s3:ListBucket"],
            "Resource": ["arn:aws:s3:::bucket-name"]
        },
        {
            "Sid": "AllObjectActions",
            "Effect": "Allow",
            "Action": "s3:*Object",
            "Resource": ["arn:aws:s3:::bucket-name/*"]
        }
    ]
}
```

> **Note**
>
> Per consentire l'accesso `Read` e `Write` a un oggetto di un bucket Amazon S3 e includere anche altre autorizzazioni per l'accesso alla console, consulta Amazon S3: consente l'accesso in lettura e scrittura agli oggetti in un bucket S3, in modo programmatico e nella console.

Amazon S3: consente l'accesso in lettura e scrittura agli oggetti in un bucket S3, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso `Read` e `Write` agli oggetti di un bucket S3 specifico. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa politica, sostituisci quella *`italicized placeholder text`* dell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

L'operazione `s3:*Object` usa un carattere jolly nel nome dell'operazione. L'istruzione `AllObjectActions` consente le operazioni `GetObject`, `DeleteObject`, `PutObject` e qualsiasi altra operazione Amazon S3 che termina con la parola "Object".

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "S3ConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket"]
        },
        {
            "Sid": "AllObjectActions",
            "Effect": "Allow",
```

```
            "Action": "s3:*Object",
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/*"]
        }
    ]
}
```

Gestire le policy IAM

IAM fornisce gli strumenti per creare e gestire tutti i tipi di policy IAM (policy gestite e policy in linea). Per aggiungere le autorizzazioni a un'identità IAM (utente, gruppo o ruolo IAM), crea una policy, convalida la policy, quindi collega la policy all'identità. È possibile anche collegare più policy a un'entità e ogni policy può contenere più autorizzazioni.

Argomenti

- Risorse aggiuntive
- Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente
- Convalida delle policy IAM
- Test delle policy IAM con il simulatore di policy IAM
- Aggiunta e rimozione di autorizzazioni per identità IAM
- Controllo delle versioni delle policy IAM
- Modificare le policy IAM
- Eliminare le policy IAM
- Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso

Risorse aggiuntive

Le seguenti risorse possono aiutarti a saperne di più sulle AWS politiche.

- Per ulteriori informazioni sui diversi tipi di policy IAM, consulta Politiche e autorizzazioni in AWS Identity and Access Management.
- Per informazioni generali sull'uso delle policy con IAM, consulta Gestione degli accessi per AWS risorse.
- Per informazioni su come utilizzare il Sistema di analisi degli accessi IAM per generare una policy IAM basata sull'attività di accesso per un'entità, consulta Generazione di policy per Sistema di analisi degli accessi IAM.

- Per ulteriori informazioni su come vengono valutate le autorizzazioni quando vengono applicate più policy per una determinata identità IAM, consulta Logica di valutazione delle policy.
- Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Definire le autorizzazioni IAM personalizzate con policy gestite dal cliente

Le politiche definiscono le autorizzazioni per le identità o le risorse in. AWS Puoi creare policy gestite dai clienti in IAM utilizzando l' Console di gestione AWS API AWS CLI, o AWS . Le policy gestite dal cliente sono policy autonome gestite dall'utente nel proprio Account AWS. Puoi quindi collegare le politiche alle identità (utenti, gruppi e ruoli) nel tuo Account AWS.

Una policy basata su identità è una policy collegata a un'identità in IAM. Le politiche basate sull'identità possono includere politiche gestite, politiche AWS gestite dai clienti e politiche in linea. AWS le politiche gestite vengono create e gestite da AWS ed è possibile utilizzarle ma non gestirle. Una policy in linea è una policy che viene creata e integrata direttamente in un gruppo di utenti, utente o ruolo IAM. Le policy in linea non possono essere riutilizzate su altre identità o gestite al di fuori dell'identità in cui esistono. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

In genere è preferibile utilizzare politiche gestite dal cliente anziché politiche in linea o politiche AWS gestite. AWS le politiche gestite in genere forniscono ampie autorizzazioni amministrative o di sola lettura. Per garantire la massima sicurezza, concedere un privilegio minimo, ovvero concedere solo le autorizzazioni necessarie per eseguire attività di processi specifici.

Quando crei o modifichi le policy IAM, AWS puoi eseguire automaticamente la convalida delle policy per aiutarti a creare una policy efficace con il minimo privilegio in mente. Nel Console di gestione AWS, IAM identifica gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce controlli aggiuntivi sulle policy con consigli per aiutarti a perfezionare ulteriormente le tue policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

Puoi utilizzare l' AWS API Console di gestione AWS AWS CLI, o per creare policy gestite dai clienti in IAM. Per ulteriori informazioni sull'utilizzo dei CloudFormation modelli per aggiungere o aggiornare le politiche, consulta il riferimento ai tipi di AWS Identity and Access Management risorse nella Guida per l'CloudFormation utente.

Argomenti

- Creare policy IAM (console)
- Creare policy IAM (AWS CLI)
- Crea policy IAM (AWS API)

Creare policy IAM (console)

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzarli Console di gestione AWS per creare policy gestite dai clienti in IAM. Le policy gestite dal cliente sono policy autonome gestite dall'utente nel proprio Account AWS. Puoi quindi allegare le politiche alle identità (utenti, gruppi e ruoli) nel tuo Account AWS.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Argomenti

- Creazione di policy IAM
- Creazione di policy utilizzando l'editor JSON
- Creazione di policy con l'editor visivo
- L'importazione di policy gestite esistenti

Creazione di policy IAM

È possibile creare una politica gestita dai clienti Console di gestione AWS utilizzando uno dei seguenti metodi:

- JSON: incolla e personalizza un esempio di policy basata sull'identità.
- Editor visivo: è possibile creare una nuova policy da zero nell'editor visivo. Se si utilizza l'editor visivo, non è necessario comprendere la sintassi JSON.
- Importa: importa e personalizza una policy gestita dall'account. È possibile importare una politica AWS gestita o una politica gestita dai clienti creata in precedenza.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Creazione di policy utilizzando l'editor JSON

Puoi digitare o incollare le policy in JSON scegliendo l'opzione JSON. Questo metodo è utile per copiare una policy di esempio da utilizzare nell'account. In alternativa, è possibile digitare il proprio documento di policy JSON nell'editor JSON. È inoltre possibile utilizzare l'opzione JSON per passare tra l'editor visivo e JSON e confrontare le visualizzazioni.

Quando si crea o si modifica una policy nell'editor JSON, IAM esegue la convalida delle policy per facilitare la creazione di una policy efficace. IAM identifica gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti utili per perfezionare ulteriormente la policy.

Un documento di policy JSON consiste in una o più istruzioni. Ogni istruzione deve contenere tutte le operazioni che condividono lo stesso risultato (`Allow` o `Deny`) e supportare le stesse risorse e condizioni. Se un'operazione richiede di specificare tutte le risorse (`"*"`) e un'altra operazione supporta l'Amazon Resource Name (ARN) di una risorsa specifica, devono essere in due diverse istruzioni JSON. Per informazioni dettagliate sui formati ARN, consulta Amazon Resource Name (ARN) nella Guida Riferimenti generali di AWS . Per informazioni generali sulle policy IAM, consulta Politiche e autorizzazioni in AWS Identity and Access Management. Per informazioni sul linguaggio delle policy IAM, consulta Riferimento alla policy JSON IAM.

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Digitare o incollare un documento di policy JSON. Per maggiori dettagli sul linguaggio della policy IAM, consulta Riferimento alla policy JSON IAM.
6. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe

ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

7. (Facoltativo) Quando crei o modifichi una policy in Console di gestione AWS, puoi generare un modello di policy JSON o YAML da utilizzare nei modelli. CloudFormation

   Per fare ciò, nell'editor delle politiche scegli Azioni, quindi scegli Genera modello. CloudFormation Per ulteriori informazioni, CloudFormation consulta il riferimento al tipo di AWS Identity and Access Management risorsa nella Guida AWS CloudFormation per l'utente.
8. Una volta terminata l'aggiunta delle autorizzazioni alla policy, scegli Successivo.
9. Nella pagina Verifica e crea, digita i valori per Nome policy e Descrizione (facoltativa) per la policy che si sta creando. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
10. (Facoltativo) Aggiungere metadati alla policy collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.
11. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato una policy, è possibile collegarlo ai gruppi, utenti o ruoli. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

Creazione di policy con l'editor visivo

L'editor visivo nella console IAM fornisce informazioni utili sulla creazione di una policy senza dover scrivere una sintassi JSON. Per visualizzare un esempio dell'editor visivo per creare una policy, consultare the section called “Controllo dell'accesso alle identità”.

Per utilizzare l'editor visivo per creare una policy.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy.
4. Nella sezione Policy editor, trova la sezione Seleziona un servizio, quindi scegli un AWS servizio. È possibile utilizzare la casella di ricerca in alto per limitare i risultati nell'elenco di servizi. È possibile selezionare solo un servizio nel blocco di autorizzazione di un editor visivo. Per

concedere l'accesso a più di un servizio, aggiungi più blocchi di autorizzazioni selezionando Aggiungi altre autorizzazioni.

5. In Operazioni consentite, scegli le operazioni da aggiungere alla policy. È possibile selezionare operazioni nei modi seguenti:

   - Selezionare la casella di controllo per tutte le azioni.
   - Scegliere aggiungi azioni per digitare il nome di un'azione specifica. È possibile utilizzare i caratteri jolly (*) per specificare più operazioni.
   - Selezionare uno dei gruppi di livelli di accesso per scegliere tutte le azioni per il livello di accesso, ad esempio Lettura, Scrittura o Elenco.
   - Espandere ciascuno dei gruppi Access level (Livello di accesso) per selezionare singole operazioni.

   Come impostazione predefinita, la policy che si sta creando utilizza le operazioni selezionate. Per rifiutare invece le operazioni scelte, selezionare Switch to deny permissions (Passa a rifiuto autorizzazioni). Poiché IAM rifiuta per impostazione predefinita, si consiglia come best practice di sicurezza di consentire le autorizzazioni solo alle operazioni e alle risorse necessarie per un utente. È necessario creare un'istruzione JSON per negare le autorizzazioni solo se si desidera sostituire un'autorizzazione separatamente consentita da un'altra istruzione o policy. Si consiglia di limitare al minimo il numero di autorizzazioni di rifiuto perché possono aumentare la difficoltà di risoluzione dei problemi relative alle autorizzazioni.

6. Per Risorse, se il servizio e le azioni selezionati nei passaggi precedenti non supportano la scelta di risorse specifiche, tutte le risorse sono consentite e non è possibile modificare questa sezione.

   Se si selezionano una o più operazioni che supportano le autorizzazioni a livello di risorsa, l'editor visivo elenca tali risorse. È possibile selezionare Risorse per specificare le risorse per la policy.

   È possibile specificare le risorse nei seguenti modi:

   - Scegli Aggiungi ARNs per specificare le risorse in base ai rispettivi Amazon Resource Names (ARN). È possibile utilizzare l'editor o l'elenco ARNs ARN visivo manualmente. Per maggiori informazioni sulla sintassi ARN, consulta Amazon Resource Name (ARN) nella Guida Riferimenti generali di AWS . Per informazioni sull'utilizzo ARNs nell'`Resource`elemento di una policy, vedereElementi della policy JSON IAM: Risorsa.

- Scegli Qualsiasi in questo account accanto a una risorsa per concedere autorizzazioni a qualsiasi risorsa di quel tipo.
- Seleziona Tutto per selezionare tutte le risorse per quel servizio.

7. (Facoltativo) Scegli Condizioni di richiesta - opzionale per aggiungere condizioni alla policy che si sta creando. Le condizioni limitano l'effetto di una dichiarazione di policy JSON. Ad esempio, puoi specificare che un utente può eseguire le operazioni sulle risorse solo quando la richiesta dell'utente viene effettuata entro un determinato intervallo di tempo. È inoltre possibile utilizzare le condizioni comuni per limitare se un utente deve essere autenticato utilizzando un dispositivo multi-factor authentication (MFA). In alternativa, è possibile richiedere che la richiesta provenga da un determinato intervallo di indirizzi IP. Per un elenco di tutte le chiavi di contesto che è possibile utilizzare in una condizione di policy, vedere Azioni, risorse e chiavi di condizione per AWS i servizi nel Service Authorization Reference.

   È possibile selezionare le condizioni nei modi seguenti:

   - Utilizzare le caselle di controllo per selezionare le condizioni di utilizzo comune.
   - Seleziona Aggiungi altra condizione per specificare altre condizioni. Selezionare Condition Key (Chiave condizione), Qualifier (Qualificatore) e Operator (Operatore) della condizione e digitare un Value (Valore). Per aggiungere più di un valore, seleziona Aggiungi. È possibile valutare i valori come se fossero connessi da un operatore logico "OR". Una volta terminato, scegli Aggiungi condizione.

   Per aggiungere più di una condizione, scegli di nuovo Aggiungi altra condizione. Ripetere come necessario. Ogni condizione si applica solo a questo blocco di autorizzazione di un editor visivo. Tutte le condizioni devono essere vere per il blocco di autorizzazioni per essere considerato una corrispondenza. In altre parole, considerare le condizioni da connettere con un operatore logico "AND".

   Per ulteriori informazioni sull'elemento Condition (Condizione), consultare Elementi della policy JSON IAM: Condizione in Riferimento alla policy JSON IAM.

8. Per aggiungere più blocchi di autorizzazioni, seleziona Aggiungi ulteriori autorizzazioni. Per ogni blocco, ripetere le fasi da 2 a 5.

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe

ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

9. (Facoltativo) Quando si crea o si modifica una politica in Console di gestione AWS, è possibile generare un modello di policy JSON o YAML da utilizzare nei modelli. CloudFormation

   Per fare ciò, nell'editor delle politiche scegli Azioni, quindi scegli Genera modello. CloudFormation Per ulteriori informazioni, CloudFormation consulta il riferimento al tipo di AWS Identity and Access Management risorsa nella Guida AWS CloudFormation per l'utente.
10. Una volta terminata l'aggiunta delle autorizzazioni alla policy, scegli Successivo.
11. Nella pagina Verifica e crea, digita i valori per Nome policy e Descrizione (facoltativa) per la policy che si sta creando. Rivedi il campo Autorizzazioni definite in questa policy per accertarti di disporre delle autorizzazioni previste.
12. (Facoltativo) Aggiungere metadati alla policy collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tag per AWS Identity and Access Management risorse.
13. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato una policy, è possibile collegarlo ai gruppi, utenti o ruoli. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

L'importazione di policy gestite esistenti

Un modo semplice per creare una nuova policy è di importare una policy gestita esistente all'interno dell'account che dispone di almeno alcune delle autorizzazioni di cui si ha bisogno. È possibile personalizzare la policy per farla corrispondere ai nuovi requisiti.

Non è possibile importare una policy inline. Per informazioni sulle differenze tra policy gestite e policy inline, consultare Policy gestite e policy inline.

Per importare una policy gestita esistente nell'editor visivo

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy.

4. Nella sezione Editor di policy, scegli l'opzione Visivo, quindi sul lato destro della pagina, scegli Operazioni e poi Importa policy.
5. Nella finestra Importa policy gestite, seleziona le policy gestite che meglio corrispondono alla policy da includere nella nuova policy. Per limitare i risultati nell'elenco di servizi, è possibile utilizzare la casella di ricerca in alto.
6. Scegli Importa policy.

   Le policy importate vengono aggiunte in nuovi blocchi di autorizzazione nella parte inferiore della policy.
7. Utilizzare Visual editor (Editor visivo) o selezionare JSON per personalizzare la policy. Quindi scegli Successivo.

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.
8. Nella pagina Verifica e crea, digita i valori per Nome policy e Descrizione (facoltativa) per la policy che si sta creando. Non è possibile modificare queste impostazioni in un secondo momento. Rivedi il campo Autorizzazioni definite in questa policy, quindi scegli Crea policy per salvare il lavoro.

Importazione di una policy gestita esistente nell'editor JSON

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON, quindi sul lato destro della pagina, scegli Operazioni e poi Importa policy.
5. Nella finestra Importa policy gestite, seleziona le policy gestite che meglio corrispondono alla policy da includere nella nuova policy. Per limitare i risultati nell'elenco di servizi, è possibile utilizzare la casella di ricerca in alto.
6. Scegli Importa policy.

Le istruzioni dalle policy importate vengono aggiunte in fondo alle policy JSON.

7. Personalizza la policy in JSON. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo). Oppure, personalizza la policy nell'Editor visivo. Quindi scegli Successivo.

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

8. Nella pagina Verifica e crea, digita i valori per Nome policy e Descrizione (facoltativa) per la policy che si sta creando. Non è possibile modificare queste impostazioni in un secondo momento. Rivedi la policy Autorizzazioni definite in questa policy, quindi scegli Crea policy per salvare il lavoro.

Dopo aver creato una policy, è possibile collegarlo ai gruppi, utenti o ruoli. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

Creare policy IAM (AWS CLI)

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzarli AWS CLI per creare policy gestite dai clienti in IAM. Le policy gestite dal cliente sono policy autonome gestite dall'utente nel proprio Account AWS. Come best practice, ti consigliamo di utilizzare IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali. Attraverso la convalida delle policy è possibile risolvere eventuali errori o suggerimenti prima di collegare le policy alle identità (utenti, gruppi e ruoli) dell' Account AWS.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Creazione di policy IAM (AWS CLI)

Puoi creare una policy gestita dal cliente IAM o una policy in line utilizzando AWS Command Line Interface (AWS CLI).

Per creare una policy gestita dal cliente (AWS CLI)

Utilizza il seguente comando:

- create-policy

Come creare una policy in linea per un'identità IAM (utente, gruppo o ruolo) (AWS CLI)

Utilizzare uno dei seguenti comandi:

- put-group-policy
- put-role-policy
- put-user-policy

> **Note**
>
> Non è possibile utilizzare IAM per integrare una policy in linea per un ruolo collegato ai servizi.

Come convalidare una policy gestita dal cliente (AWS CLI)

Utilizza il seguente comando IAM Access Analyzer:

- validate-policy

Crea policy IAM (AWS API)

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzare l' AWS API per creare policy gestite dai clienti in IAM. Le policy gestite dal cliente sono policy autonome gestite dall'utente nel proprio Account AWS. Come best practice, ti consigliamo di utilizzare IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali. Attraverso la convalida delle policy è possibile risolvere eventuali errori o suggerimenti prima di collegare le policy alle identità (utenti, gruppi e ruoli) dell' Account AWS.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Creazione di politiche IAM (AWS API)

Puoi creare una policy gestita dal cliente IAM o una policy in linea utilizzando l'API AWS .

Per creare una policy gestita dai clienti (AWS API)

Chiamare l'operazione seguente:

- CreatePolicy

Per creare una policy in linea per un'identità IAM (gruppo, utente o ruolo) (AWS API)

Chiamare una delle seguenti operazioni:

- PutGroupPolicy
- PutRolePolicy
- PutUserPolicy

> **Note**
>
> Non è possibile utilizzare IAM per integrare una policy in linea per un ruolo collegato ai servizi.

Per convalidare una politica gestita dal cliente (API)AWS

Chiama la seguente operazione IAM Access Analyzer:

- ValidatePolicy

Convalida delle policy IAM

Una policy è un documento JSON che utilizza la sintassi delle policy IAM. Quando colleghi una policy a un'entità IAM, come un utente, un gruppo o un ruolo, la policy concede le autorizzazioni a tale entità.

Quando crei o modifichi le policy di controllo degli accessi IAM utilizzando Console di gestione AWS, le esamina AWS automaticamente per garantire che siano conformi alla grammatica delle policy IAM. Se AWS determina che una policy non rispetta la sintassi, verrà richiesto di correggere la policy.

IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente la policy. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer. Per visualizzare un elenco di avvisi, errori e suggerimenti restituiti da IAM Access Analyzer, consulta Riferimento ai controlli delle policy IAM Access Analyzer.

Ambito della convalida

AWS controlla la sintassi e la grammatica della policy JSON. Verifica inoltre che la formattazione sia corretta e che i nomi delle azioni e i tasti di condizione ARNs siano corretti.

Accesso alla convalida delle policy

Le policy vengono convalidate automaticamente quando si crea una policy JSON o si modifica una policy esistente nella Console di gestione AWS. Se la sintassi della policy non è valida, riceverai una notifica e dovrai correggere il problema prima di poter continuare. I risultati della convalida delle policy di IAM Access Analyzer vengono restituiti automaticamente nella cartella Console di gestione AWS se disponi delle autorizzazioni per. `access-analyzer:ValidatePolicy` Puoi anche convalidare le politiche utilizzando l'API o. AWS AWS CLI

Policy esistenti

È possibile che le policy esistenti non siano valide perché sono state create o salvate per l'ultima volta prima degli ultimi aggiornamenti del motore di policy. Come best practice, ti consigliamo di utilizzare IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali. Consigliamo di aprire le policy esistenti e rivedere i risultati della convalida della policy generati. Non è possibile modificare e salvare le policy esistenti senza correggere eventuali errori di sintassi della policy.

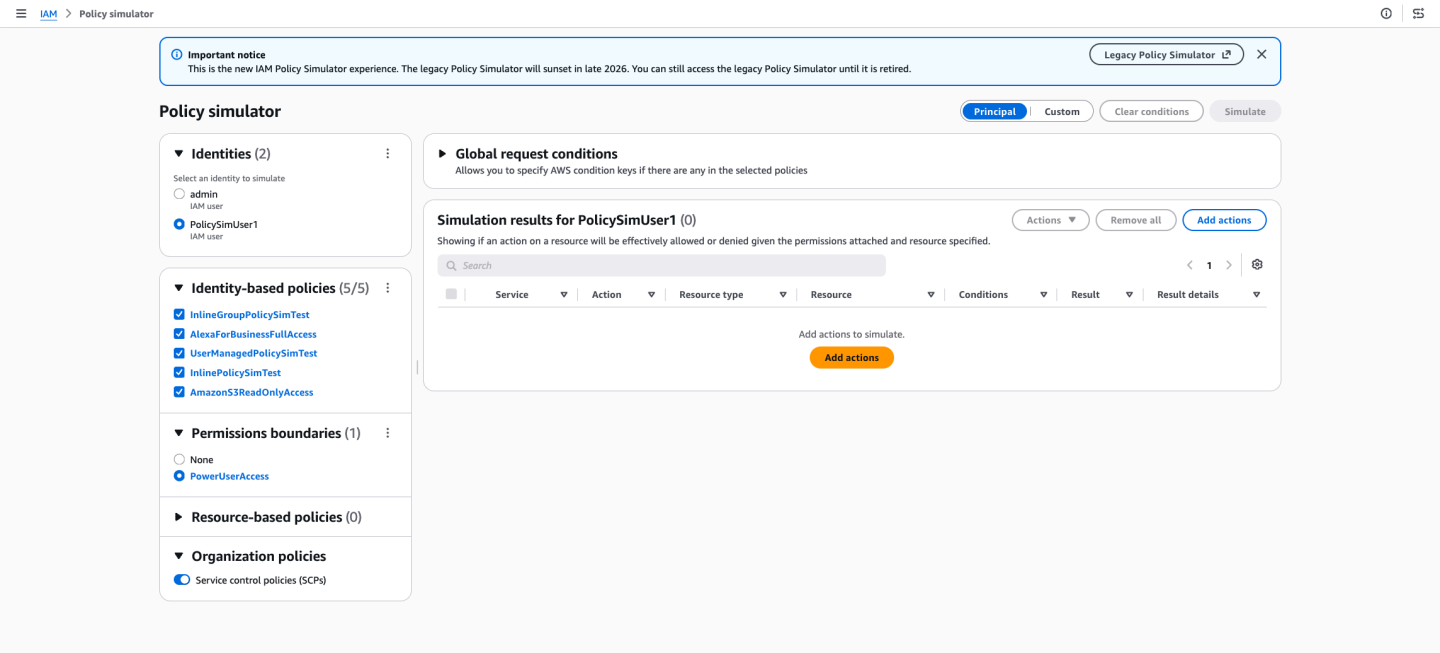
Test delle policy IAM con il simulatore di policy IAM

Con il simulatore di policy IAM, puoi testare le policy basate sull'identità, i limiti delle autorizzazioni IAM, le policy di controllo dei servizi (SCP) e le policy basate sulle risorse fornite. Simuli il modo in cui AWS valuteresti una richiesta rispetto a tali politiche, senza inviare una richiesta reale a nessun servizio. AWS Ogni simulazione restituisce un risultato di autorizzazione binario per ogni azione e risorsa testata: l'azione è consentita o negata. Quando il risultato è un'autorizzazione o un rifiuto esplicito, il simulatore mostra anche quale politica ha prodotto tale risultato, in modo da poter comprendere e risolvere la causa.

Per ulteriori informazioni su come è perché utilizzare le policy IAM, consulta Politiche e autorizzazioni in AWS Identity and Access Management.

> **Important**
>
> I risultati del simulatore di policy possono differire dall'ambiente reale in uso. AWS Ti consigliamo di confrontare le tue politiche con il tuo AWS ambiente reale dopo averle testate con il simulatore di policy per confermare di avere i risultati desiderati. Per ulteriori informazioni, consulta Come funziona il simulatore di policy IAM.

La schermata seguente mostra il simulatore di policy IAM, in cui è possibile selezionare un'identità e le relative politiche e scegliere le azioni e le risorse da testare prima di eseguire la simulazione.



Argomenti

- Modalità del simulatore di policy
- Cosa puoi fare con il simulatore di policy
- Come funziona il simulatore di policy IAM
- Come funziona la valutazione delle condizioni chiave nel simulatore di policy IAM
- Autorizzazioni per l'utilizzo del simulatore di policy
- Come simulare le politiche

Modalità del simulatore di policy

Il simulatore di politiche ha due modalità. Scegli la modalità corrispondente se desideri testare le politiche già esistenti nel tuo account o le politiche che stai ancora elaborando.

Modalità principale

In modalità Principal, si testano le policy già associate a un utente, ruolo o gruppo IAM esistente. Si sceglie un'identità per la simulazione e il simulatore valuta le politiche associate a tale identità. È inoltre possibile includere o escludere politiche di identità personalizzate o limiti di autorizzazione alla simulazione. Queste aggiunte vengono simulate come se fossero collegate al principale, senza essere effettivamente collegate in IAM, quindi il tuo account rimane invariato.

Modalità personalizzata

In modalità personalizzata, si testano una o più politiche che si scrivono o si incollano. Ciò è utile per controllare una policy prima di allegarla. Le politiche personalizzate vengono simulate da sole e non vengono salvate nel tuo account.

Per istruzioni dettagliate e dettagliate sulla selezione di una modalità e sull'esecuzione di una simulazione, consulta. Come simulare le politiche

Cosa puoi fare con il simulatore di policy

Ecco alcune operazioni comuni che puoi fare con il simulatore di policy. Una guida dettagliata e una distinzione più dettagliata tra i comportamenti delle API e delle console sono disponibili in. Come simulare le politiche

- Prova le policy basate sull'identità, indipendentemente dal fatto che siano già collegate a un utente, gruppo o ruolo IAM, oppure nuove policy che scrivi o incolli. Per un'identità associata con più di una policy, puoi testare tutte le policy insieme o selezionare singole policy e vedere quali azioni sono consentite o negate per risorse specifiche. Le nuove policy vengono valutate come se fossero associate a un'identità, senza effettivamente collegarle al tuo account; vengono utilizzate solo nella simulazione e non vengono salvate nel tuo account.
- Verifica l'effetto dei controlli restrittivi sull'accesso. Puoi simulare un limite di autorizzazioni alla volta. Se sei membro di un'organizzazione in AWS Organizations, puoi anche testare l'impatto delle politiche di controllo dei servizi (SCP) sulle tue Account AWS politiche basate sull'identità. Il simulatore ti dice se un SCP consente o nega un'azione, ma per motivi di sicurezza non fa emergere le dichiarazioni corrispondenti di un SCP come avviene per altri tipi di policy.

- Verifica l'effetto di una policy basata sulle risorse che fornisci. Tranne che nella console, il simulatore non recupera la policy di una risorsa per te. Per includere una policy basata sulle risorse in una simulazione, fornisci la policy insieme alla risorsa.
- Testa servizi, azioni e risorse specifici in condizioni reali e identifica cosa determina il risultato. Ad esempio, puoi verificare se una policy consente le `ListAllMyBuckets` `DeleteBucket` azioni in Amazon S3 su un bucket specifico e fornire chiavi di contesto come un indirizzo IP o una data a cui si fa riferimento negli `Condition` elementi delle politiche da testare. `CreateBucket` Per ogni risultato, il simulatore identifica quale istruzione specifica consente o nega l'accesso; le dichiarazioni corrispondenti possono provenire da politiche basate sull'identità e da politiche basate sulle risorse fornite dall'utente.

Come funziona il simulatore di policy IAM

Quando si esegue una simulazione, il simulatore di policy valuta le politiche nell'ambito della richiesta, eventuali SCP nell'ambito e gli input forniti, come azioni, risorse e valori chiave di contesto. In modalità personalizzata, fornisci direttamente le politiche. In modalità Principal, il simulatore utilizza le politiche allegate all'identità selezionata ed è possibile includere o escludere più politiche dalla simulazione. Il risultato è un risultato binario per ogni azione e risorsa: consentita o negata.

Il simulatore di policy si differenzia dall' AWS ambiente live nei seguenti modi:

- Il simulatore di policy non effettua una richiesta di AWS servizio effettiva, quindi puoi testare in sicurezza le richieste che altrimenti potrebbero apportare modifiche indesiderate al tuo ambiente live. AWS Non utilizza valori chiave di contesto reali provenienti dalla produzione e, poiché le azioni non vengono effettivamente eseguite, non restituisce alcuna risposta dal servizio. L'unico risultato è se ogni azione richiesta sarà consentita o negata.
- Se si modifica una policy nel simulatore di policy, la modifica ha effetto solo sulla simulazione. La politica corrispondente nella tua Account AWS rimane invariata.
- Il simulatore di policy valuta gli SCP, incluse le chiavi di condizione e l'ambito delle risorse nelle dichiarazioni di rifiuto, ma non supporta le politiche di controllo delle risorse (RCP). I risultati della simulazione possono comunque differire dal comportamento effettivo per alcune configurazioni avanzate, come le policy degli endpoint VPC, il concatenamento dei ruoli o le politiche basate su più risorse su una singola risorsa.
- Cross-account le simulazioni restituiscono decisioni relative al tipo di policy (politica di identità e politica delle risorse) in. `EvalDecisionDetails`

Come funziona la valutazione delle condizioni chiave nel simulatore di policy IAM

Molte policy utilizzano `Condition` elementi che dipendono da chiavi di condizione, come `aws:RequestedRegion` o`aws:MultiFactorAuthPresent`. Quando si esegue una simulazione, il simulatore di policy ottiene i valori per le chiavi di condizione da due fonti: valori che il simulatore compila automaticamente e valori forniti come input di simulazione. Se un valore non soddisfa una condizione, l'azione viene negata.

Contesto che il simulatore compila automaticamente

Il simulatore compila le seguenti chiavi di contesto principali e organizzative per la valutazione:

- Contesto principale:`aws:PrincipalAccount`,,`aws:PrincipalId`,,`aws:PrincipalType`, `aws:Type` `aws:UserId` `aws:UserName`
- Contesto organizzativo:`aws:PrincipalOrgID`,`aws:PrincipalOrgMasterAccountId`, `aws:PrincipalOrgPaths`

Poiché queste chiavi sono compilate automaticamente, gli SCP e le altre politiche che utilizzano condizioni principali o organizzative vengono valutate correttamente. Fornisci i valori per tutte le altre chiavi di condizione a cui fanno riferimento le tue politiche. Esamina gli `Condition` elementi delle politiche che stai testando e fornisci i valori per tutte le chiavi non presenti nell'elenco precedente.

Chiavi di condizione globali e assistenza

Per le politiche basate sull'identità e le politiche basate sulle risorse, il simulatore non determina quali servizi supportano una determinata chiave di condizione globale per l'autorizzazione. Ad esempio, il simulatore non identifica che un servizio non supporta. `aws:TagKeys` Al contrario, le politiche di controllo dei servizi (SCP) vengono valutate con una logica sensibile ai servizi, incluse chiavi di condizione globali come e. `aws:RequestedRegion` `aws:PrincipalAccount` Per ulteriori informazioni su queste chiavi di condizione, consulta AWS chiavi di contesto delle condizioni globali.

Chiavi di condizione nelle politiche di controllo dei servizi (SCP)

Il simulatore valuta le chiavi di condizione e l'ambito delle risorse negli SCP. Il modo in cui fornisci i valori per quelle chiavi dipende da come esegui la simulazione:

- Con l' AWS API AWS CLI or, puoi fornire valori per le chiavi di condizione a cui fanno riferimento gli SCP.

- Nella console IAM, puoi impostare solo i valori per le chiavi di condizione che appaiono nelle politiche basate sull'identità, nei limiti delle autorizzazioni o nelle politiche basate sulle risorse. Se una chiave di condizione è referenziata solo da un SCP, non è possibile impostarne un valore nella console. Tuttavia, se la stessa chiave di condizione appare anche in una di queste altre politiche, il valore impostato viene utilizzato nell'intera valutazione, incluso l'SCP.

Per motivi di sicurezza, la valutazione SCP non restituisce valori di contesto mancanti. Riceverai comunque la decisione di consentire o negare la richiesta.

Autorizzazioni per l'utilizzo del simulatore di policy

Le autorizzazioni necessarie dipendono dal fatto che utilizzi l' AWS API AWS CLI or o la console IAM e dalla modalità utilizzata.

Console

È necessaria l'autorizzazione per chiamare `iam:GetContextKeysForCustomPolicy` per utilizzare il simulatore di policy IAM sulla console.

Le autorizzazioni aggiuntive necessarie dipendono dalla modalità. In modalità Principal hai bisogno di autorizzazioni di lettura aggiuntive per elencare le identità del tuo account e leggere le politiche allegate. Per ridurre gli importi da concedere, utilizza la modalità Personalizzata se hai solo bisogno di testare le policy che hai incollato.

Modalità personalizzata

Per utilizzare la modalità personalizzata, in cui si incollano le politiche che si desidera testare, è necessario anche `iam:SimulateCustomPolicy`

Modalità principale

Per utilizzare la modalità Principal, in cui la console carica e visualizza le identità e le politiche dell'account, è necessaria l'autorizzazione comune più quanto segue. Questo è il set più grande, perché la console deve scoprire le identità e leggere le politiche allegate.

Simula:

- `iam:SimulatePrincipalPolicy`

Identità delle inserzioni:

- `iam:ListUsers`, `iam:ListRoles`, `iam:ListGroups`
- `iam:ListGroupsForUser`

Elenco delle politiche allegate:

- `iam:ListPolicies`, `iam:ListUserPolicies`, `iam:ListRolePolicies`, `iam:ListGroupPolicies`
- `iam:ListAttachedUserPolicies`, `iam:ListAttachedRolePolicies`, `iam:ListAttachedGroupPolicies`

Ottenere i documenti relativi alle politiche:

- `iam:GetPolicyVersion`, `iam:GetUserPolicy`, `iam:GetRolePolicy`, `iam:GetGroupPolicy`

Ottenere i limiti delle autorizzazioni:

- `iam:GetUser`, `iam:GetRole`

Autorizzazioni aggiuntive per la modalità Principal per simulare politiche basate sulle risorse

Per includere la politica di una risorsa in una simulazione in modalità Principal, aggiungi l'autorizzazione di lettura per quel tipo di risorsa:

- `s3:GetBucketPolicy`, `sns:GetTopicAttributes`, `sqs:GetQueueUrl`, `sqs:GetQueueAttributes`, `glacier:GetVaultAccessPolicy`

Ad esempio, le politiche che concedono l'accesso al simulatore, vederethe section called "Politiche di esempio: AWS Identity and Access Management (IAM)".

AWS CLI and AWS API

Per l' AWS API AWS CLI e, è necessaria solo l'autorizzazione per chiamare l'azione di simulazione per il tipo di policy che si desidera testare. Se usi `iam:GetContextKeysForCustomPolicy` e `iam:GetContextKeysForPrincipalPolicy` per scoprire prima le chiavi di contesto, concedi anche le autorizzazioni per queste azioni.

Per motivi di sicurezza, le azioni sono suddivise nelle due modalità seguenti. Concedi solo le autorizzazioni che corrispondono a ciò che desideri simulare.

Per simulare politiche che passi direttamente all'API come stringhe (politiche personalizzate)

- `iam:SimulateCustomPolicy`
- `iam:GetContextKeysForCustomPolicy`(solo se si utilizza la fase di individuazione dei tasti di scelta rapida)

Queste azioni consentono di testare politiche non ancora associate a un utente, gruppo di utenti o ruolo. Le policy vengono passate come stringhe e vengono utilizzate solo nella simulazione.

Per simulare le politiche associate a un utente, gruppo di utenti, ruolo o risorsa IAM

- `iam:SimulatePrincipalPolicy`
- `iam:GetContextKeysForPrincipalPolicy`(solo se si utilizza la fase di individuazione della chiave di contesto)

Poiché queste azioni possono rivelare le autorizzazioni assegnate ad altre entità IAM, valuta la possibilità di limitare chi può chiamarle. Ad esempio, per concedere a un utente di nome Bob il permesso di simulare una policy assegnata a un utente di nome Alice, concedi a Bob l'accesso alla risorsa. `arn:aws:iam::777788889999:user/alice`

Per visualizzare un esempio di policy che consente di utilizzare l'API del simulatore di policy per le policy collegate e non collegate nella versione corrente, vedere. Account AWSIAM: accesso all'API del simulatore di policy Per visualizzare un esempio di policy rivolto agli utenti con un percorso specifico, vedere. IAM: accesso all'API simulatore di policy basata sul percorso degli utenti

Come simulare le politiche

Puoi eseguire simulazioni nella console IAM o con l'API AWS CLI and AWS . Per una panoramica di ciò che il simulatore valuta, delle due modalità e di come raggiunge un risultato, consulta. Test delle policy IAM con il simulatore di policy IAM Per le autorizzazioni necessarie, consulta. Autorizzazioni per l'utilizzo del simulatore di policy

Argomenti

- Simula le politiche (console)
- Simula le politiche (AWS CLI and AWS API)

Simula le politiche (console)

È possibile eseguire simulazioni dal simulatore di policy nella console IAM. https://console.aws.amazon.com/iam/ Nel pannello di navigazione, scegli Policy Simulator.

> **Note**
>
> La console autonoma del simulatore di policy su non https://policysim.aws.amazon.com/ viene più gestita. Usa il simulatore di policy nella console IAM.

Modalità principale

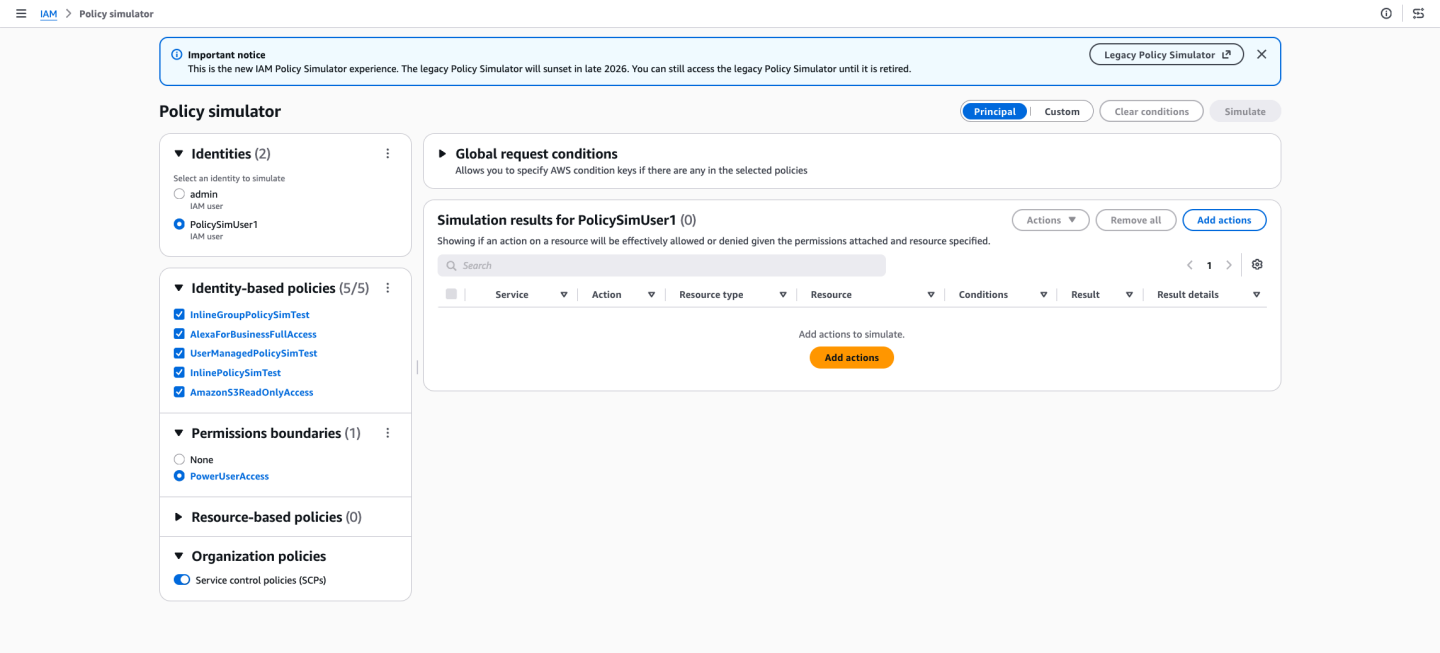
Seleziona un'identità

Per prima cosa si inizia selezionando un'identità. Puoi selezionare utenti, ruoli o gruppi IAM in quell'account. Il pannello laterale popolerà le politiche allegate a queste identità dopo la selezione.

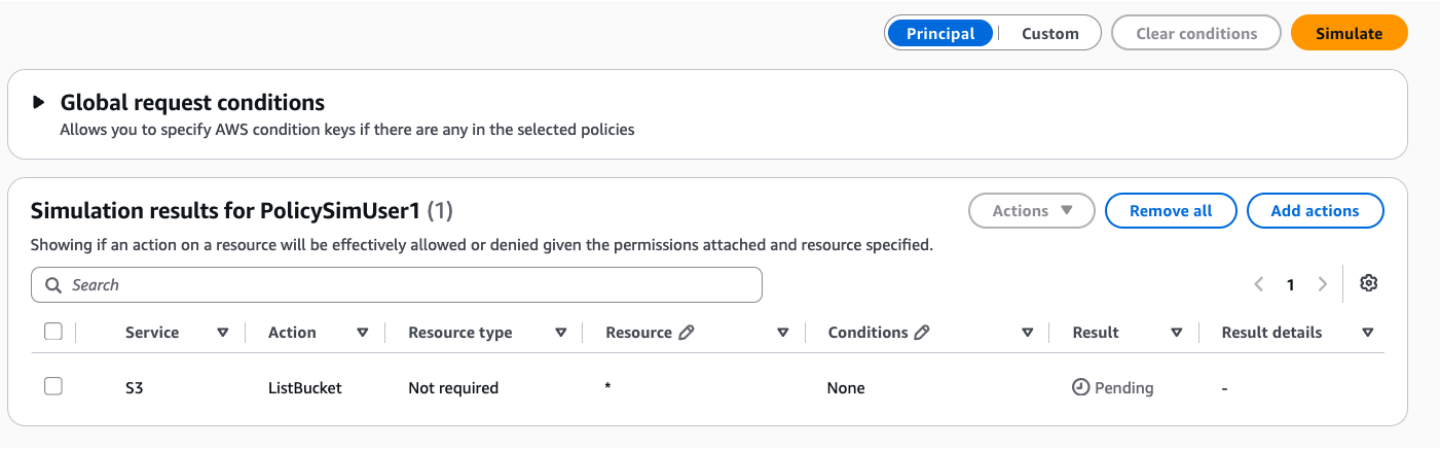
Includi o escludi le politiche

Il pannello laterale elenca le politiche oggetto della simulazione, incluse le politiche in linea, AWS gestite e gestite dal cliente e gli eventuali limiti delle autorizzazioni. Puoi escludere una policy per vedere come cambia il risultato senza di essa, quindi includerla nuovamente. L'esclusione di una politica che ha concesso l'accesso modifica il risultato in un rifiuto implicito.

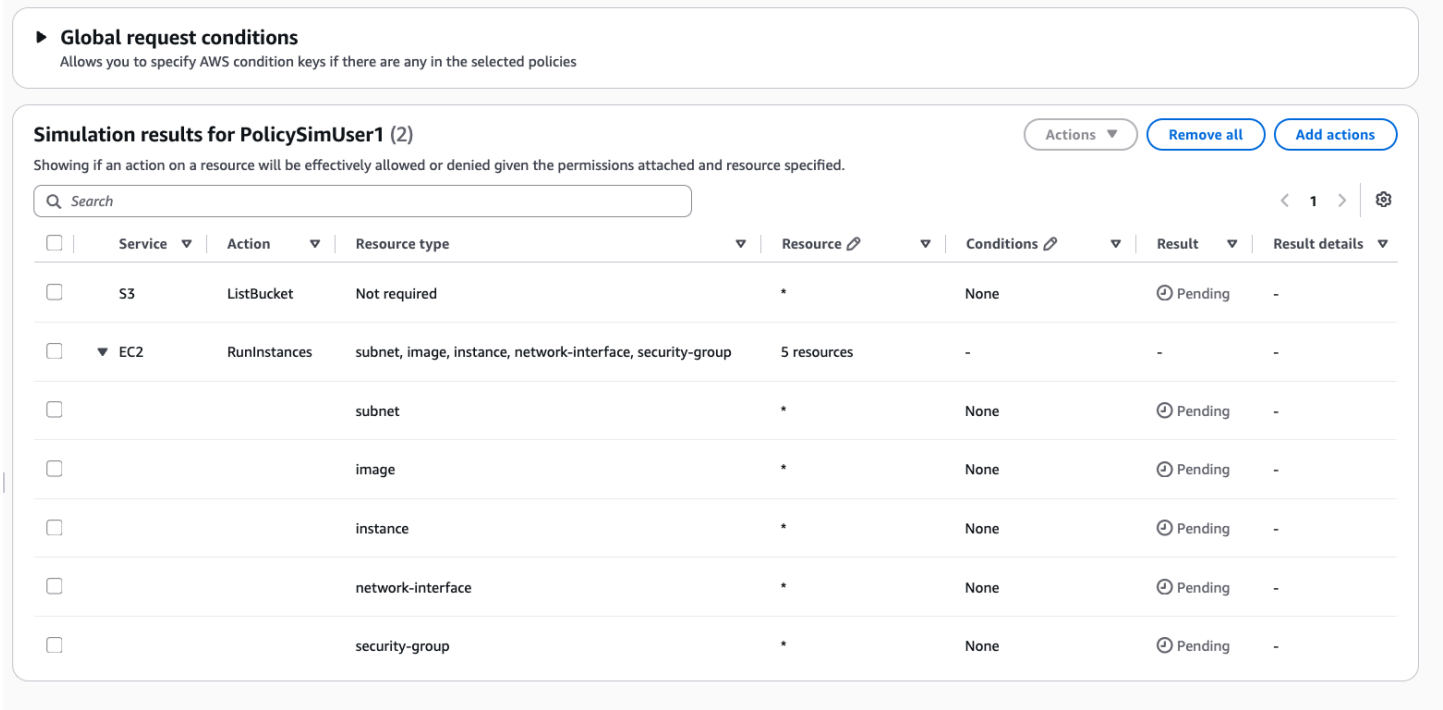
Aggiungi azioni e risorse



Dopo aver scelto identità e policy nel pannello di sinistra, aggiungi le azioni che desideri testare. Per un'azione che supporti una selezione di risorse, puoi eseguire il test su tutte le risorse o inserire uno o più ARN di risorse specifici. Quando più azioni utilizzano lo stesso tipo di risorsa, è possibile applicare una risorsa a tutte contemporaneamente oppure limitarla a una singola azione.



Alcune azioni richiedono un gruppo di tipi di risorse correlati prima che il simulatore possa valutarle. Ad esempio, durante la simulazione`ec2:RunInstances`, la console richiede di specificare insieme un'immagine, un'istanza e un gruppo di sicurezza e una sottorete o un volume a seconda dello scenario.



Fornisci i valori delle chiavi di condizione

Se le politiche testate includono chiavi di condizione, il simulatore elenca le chiavi pertinenti e consente di impostare i valori prima di eseguire la simulazione. Le chiavi mostrate riflettono le politiche attualmente selezionate, quindi selezionando o deselezionando una policy si aggiungono o si rimuovono le relative chiavi. Se un valore non soddisfa una condizione, l'azione viene negata.

Il simulatore fornisce i valori per le chiavi di condizione in due punti. Le condizioni di richiesta globali si applicano all'intera richiesta simulata. Queste sono le chiavi che iniziano con`aws:`, ad esempio `aws:MultiFactorAuthPresent` o`aws:PrincipalServiceNamesList`, e i loro valori vengono impostati una sola volta nell'area delle condizioni globali. La selezione o la cancellazione di una policy aggiunge o rimuove le relative chiavi globali e i valori immessi vengono mantenuti quando si modifica l'ambito delle politiche. Service-specific le condizioni si applicano a una singola azione. Ogni riga dell'azione mostra il numero di condizioni a cui fa riferimento e si apre la riga per impostare le chiavi specifiche del servizio per quell'azione, ad esempio `cloudwatch:namespace` for `cloudwatch:PutMetricData` o `ram:PermissionResourceType` for. `ram:ListPermissionAssociations` Una riga di azione può includere sia una chiave globale che una chiave specifica del servizio e, se un valore non soddisfa una condizione, tale azione viene negata.

Il simulatore di policy IAM compila al posto tuo alcune chiavi di contesto principali e organizzative, mentre tu fornisci i valori per il resto. Per l'elenco delle chiavi che vengono compilate

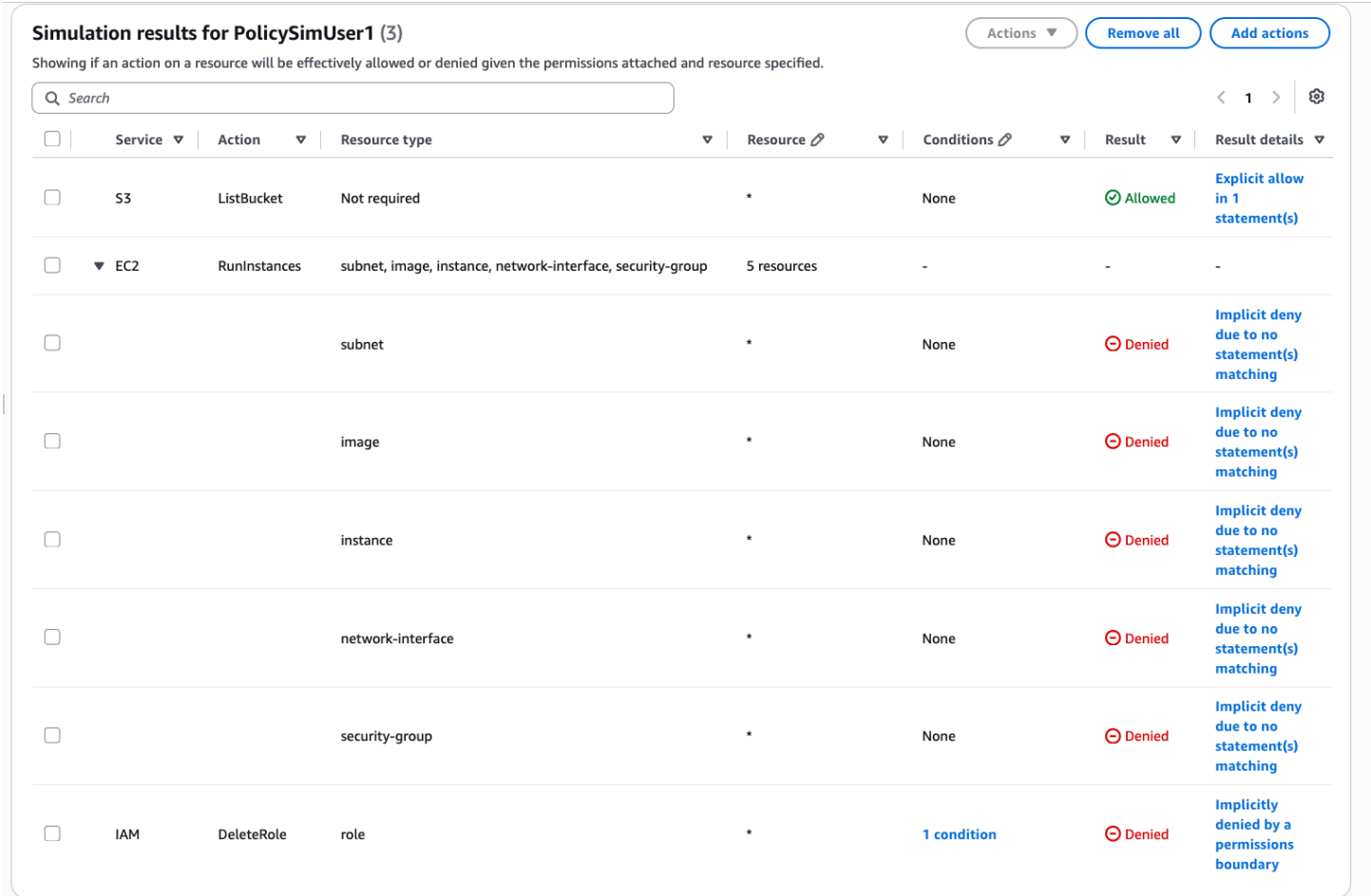
automaticamente e ulteriori informazioni su come vengono valutate le chiavi di condizione, consulta. Come funziona la valutazione delle condizioni chiave nel simulatore di policy IAM

Per informazioni su come utilizzare le chiavi di condizione, vedereAWS chiavi di contesto delle condizioni globali.

Verifica un limite di autorizzazioni

Quando un'identità ha un limite di autorizzazioni, viene visualizzata nel pannello laterale ed è inclusa per impostazione predefinita. Puoi anche escludere il limite ed eseguire nuovamente la simulazione, se vuoi scoprire se il limite è il motivo per cui un'azione viene negata.

Ad esempio, supponiamo di simulare un'identità `iam:DeleteRole` il cui confine non lo consente. Il risultato è un rifiuto attribuito al limite delle autorizzazioni. Se poi aggiungi una policy che consente l'azione ed esclude il limite, il risultato diventa un permesso. Ciò conferma che il limite era l'unico controllo che bloccava l'azione.



Includi politiche di controllo del servizio

Se il tuo account è membro di un'organizzazione, puoi includere le politiche di controllo del servizio (SCP) nella simulazione dal pannello laterale. Il simulatore valuta le chiavi di condizione compilate automaticamente e l'ambito delle risorse negli SCP quando determina il risultato. Quando un SCP nega esplicitamente un'azione, il risultato è un rifiuto esplicito attribuito a, e un permesso in un'altra policy non può sovrascriverlo. AWS Organizations Disattiva il controllo ed esegui nuovamente la simulazione per vedere il risultato senza i controlli dell'organizzazione. Per un elenco delle chiavi di condizione compilate automaticamente, vedere. Come funziona la valutazione delle condizioni chiave nel simulatore di policy IAM

> **Note**
>
> Per motivi di sicurezza, il simulatore non fornisce i dettagli della valutazione SCP. Le chiavi di condizione a cui fanno riferimento solo gli SCP non vengono elencate per consentire l'impostazione dei valori e le istruzioni corrispondenti restituite per un risultato non includono le istruzioni degli SCP. Le chiavi e le dichiarazioni corrispondenti provengono dalle politiche basate sull'identità allegate all'identità e da tutte le politiche fornite.

Prova una policy basata sulle risorse

Per includere una policy basata sulle risorse in una simulazione di console, inserisci un ARN specifico per la risorsa e scegli l'opzione per includere la policy della risorsa. La politica basata sulle risorse viene visualizzata nel pannello laterale e viene valutata nel risultato. Nella console, è disponibile per i bucket Amazon S3, le code Amazon SQS, gli argomenti Amazon SNS e i vault Amazon Glacier.

> **Note**
>
> La simulazione di policy basate sulle risorse non è supportata per i ruoli IAM.

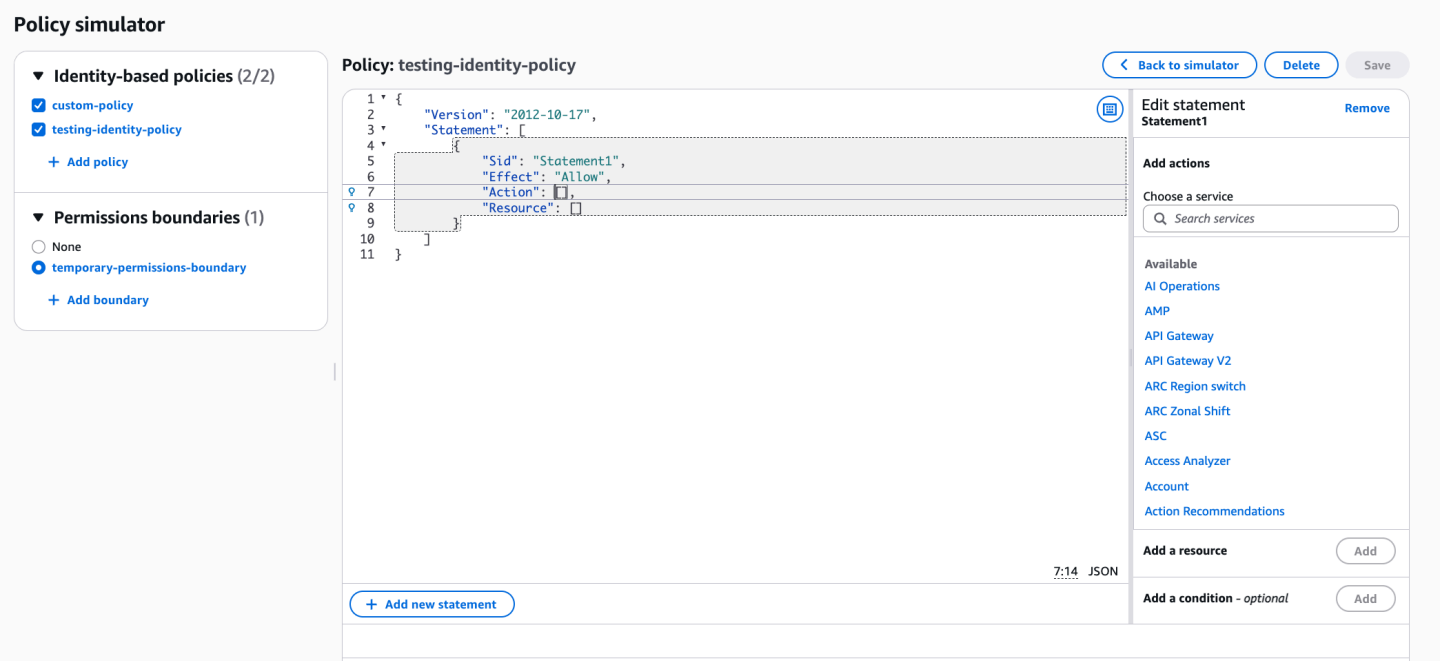
Leggi i risultati

Ogni riga della tabella dei risultati mostra se l'azione è consentita, negata implicitamente o esplicitamente negata, insieme a una breve spiegazione del risultato. Usa la spiegazione per identificare quale policy o controllo ha determinato il risultato, ad esempio un'autorizzazione esplicita, l'assenza di una dichiarazione corrispondente, un limite di autorizzazioni o un controllo dell'organizzazione.

Modalità personalizzata

Puoi utilizzare l'editor delle politiche IAM per scrivere o incollare politiche basate sull'identità e limiti di autorizzazione in modalità personalizzata. Queste politiche non sono associate a un'identità e possono essere valutate prima di applicarle. È possibile includere più politiche basate sull'identità, ma solo un limite di autorizzazioni per la simulazione.

> **Note**
>
> In modalità personalizzata, la console non supporta il trasferimento di SCP personalizzati o la simulazione di policy basate sulle risorse. Per simulare SCP personalizzati o policy basate sulle risorse, utilizza l'API o. AWS CLI AWS



Simula le politiche (AWS CLI and AWS API)

La simulazione delle policy con l' AWS API AWS CLI or richiede generalmente due passaggi:

1. Valuta le policy e restituisci l'elenco delle chiavi di contesto a cui fanno riferimento, in modo da sapere quali valori delle chiavi di contesto fornire. Questa fase è facoltativa.
2. Simula le politiche, fornendo le azioni, le risorse e i valori chiave di contesto da utilizzare durante la simulazione.

Ogni azione è associata a ciascuna risorsa e la simulazione determina se le politiche consentono o negano tale azione per quella risorsa. Puoi fornire valori per qualsiasi chiave di contesto a cui fanno riferimento le tue politiche. Se non si fornisce un valore per una chiave di contesto, la simulazione viene comunque eseguita e restituisce un rifiuto implicito e le chiavi mancanti vengono restituite in `missingContextValues` modo da sapere quali valori aggiungere. Se una chiave di contesto mancante è referenziata solo da un SCP, tale chiave non viene restituita per motivi di sicurezza, ma riceverai comunque la decisione.

Le azioni API sono suddivise in due gruppi:

- Azioni che simulano solo le politiche passate direttamente all'API come stringhe: GetContextKeysForCustomPolicy e. SimulateCustomPolicy
- Azioni che simulano le policy associate a un utente, gruppo di utenti, ruolo o risorsa IAM specificato: e. GetContextKeysForPrincipalPolicy SimulatePrincipalPolicy Poiché possono rivelare le autorizzazioni assegnate ad altre entità IAM, valuta la possibilità di limitarne l'accesso.

Queste azioni supportano anche quanto segue:

- `OrderedOrganizationPolicyInputListon` SimulateCustomPolicy consente di simulare l'effetto della gerarchia SCP di un'organizzazione senza utilizzare un responsabile attivo.
- `PolicyExclusionListon` SimulatePrincipalPolicy ti consente di testare «cosa succede se rimuovo questa politica?» scenari senza modificare le politiche associate al principale.
- La risposta ne restituisce una singola `EvaluationResult` per azione, con decisioni aggregate al livello più alto e dettagli per risorsa inseriti. `ResourceSpecificResults`

AWS CLI and AWS Guida di riferimento alle API

Per esempi di AWS CLI comandi per simulare le politiche, vedi quanto segue:

- AWS CLI: get-context-keys-for-custom-policy, get-context-keys-for-principal-policy, simulate-custom-policy, simulate-principal-policy https://docs.aws.amazon.com/cli/latest/reference/iam/get-context-keys-for-principal-policy.html https://docs.aws.amazon.com/cli/latest/reference/iam/simulate-custom-policy.html https://docs.aws.amazon.com/cli/latest/reference/iam/simulate-principal-policy.html
- AWS API:,, GetContextKeysForCustomPolicy GetContextKeysForPrincipalPolicy SimulateCustomPolicy SimulatePrincipalPolicy

Aggiunta e rimozione di autorizzazioni per identità IAM

Puoi utilizzare le policy per definire le autorizzazioni per una identità (utente, gruppo di utenti o ruolo). Puoi aggiungere e rimuovere le autorizzazioni allegando e scollegando le policy IAM per un'identità utilizzando Console di gestione AWS, the AWS Command Line Interface (AWS CLI) o l'API. AWS Puoi usare le policy anche per impostare limiti delle autorizzazioni solo per le entità (utenti o ruoli) che utilizzano gli stessi metodi. I limiti delle autorizzazioni sono una AWS funzionalità avanzata che controlla il numero massimo di autorizzazioni che un'entità può avere.

Argomenti

- Terminologia
- Visualizzazione dell'attività delle identità
- Aggiunta di autorizzazioni per identità IAM (console)
- Rimozione delle autorizzazioni per le identità IAM (console)
- Aggiunta di policy IAM (AWS CLI)
- Rimozione di policy IAM (AWS CLI)
- Aggiungere politiche IAM (AWS API)
- Rimuovere le politiche IAM (API)AWS

Terminologia

Quando associ le policy di autorizzazione alle identità (utenti IAM, gruppi IAM e ruoli IAM), la terminologia e le procedure variano a seconda che lavori con una policy gestita o con una policy in linea:

- Collega: utilizzato con le policy gestite. Una policy gestita si collega a un'identità (utente, gruppo di utenti o ruolo). Il collegamento di una policy prevede l'applicazione delle sue autorizzazioni all'identità.
- Distacca: utilizzato con le policy gestite. Una policy gestita viene scollegata da un'identità IAM (utente, gruppo di utenti o ruolo). Il distaccamento di una policy prevede la rimozione delle sue autorizzazioni dall'identità.
- Integra: utilizzato con le policy in linea. Una policy in linea viene integrata in una identità (utente, gruppo di utenti o ruolo). L'incorporamento di un policy prevede l'applicazione delle sue autorizzazioni all'identità. Poiché una policy inline è memorizzata nell'identità, è incorporata anziché collegata, anche se i risultati sono simili.

> **Note**
>
> Puoi incorporare una policy inline per un ruolo collegato al servizio solo in un servizio che dipende dal ruolo. Consulta la Documentazione di AWS relativa al servizio per scoprire se questa funzionalità è supportata.

- Elimina: utilizzato con le policy in linea. Una policy in linea viene eliminata da un'identità IAM (utente, gruppo di utenti o ruolo). L'eliminazione di una policy rimuove le sue autorizzazioni dall'identità.

> **Note**
>
> Puoi eliminare una policy inline per un ruolo collegato al servizio solo in un servizio che dipende dal ruolo. Consulta la Documentazione di AWS relativa al servizio per scoprire se questa funzionalità è supportata.

Puoi utilizzare la console o l' AWS CLI AWS API per eseguire una qualsiasi di queste azioni.

Ulteriori informazioni

- Per ulteriori informazioni sulle differenze tra policy gestite e policy inline, consulta Policy gestite e policy inline.
- Per ulteriori informazioni sui limiti delle autorizzazioni, consultare Limiti delle autorizzazioni per le entità IAM.
- Per informazioni generali sulle policy IAM, consulta Politiche e autorizzazioni in AWS Identity and Access Management.
- Per ulteriori informazioni sulla convalida delle policy IAM, consulta Convalida delle policy IAM.
- Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Visualizzazione dell'attività delle identità

Prima di modificare le autorizzazioni per un'identità (utente, gruppo di utenti o ruolo), è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori

informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Aggiunta di autorizzazioni per identità IAM (console)

Puoi utilizzare il Console di gestione AWS per aggiungere autorizzazioni a un'identità (utente, gruppo di utenti o ruolo). A tale scopo, collega le policy gestite che controllano le autorizzazioni oppure specifica una policy che funga da limite delle autorizzazioni. È inoltre possibile incorporare una policy inline.

Per usare una policy gestita come policy di autorizzazione per un'identità (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco di policy seleziona il pulsante di opzione accanto al nome della policy da collegare. Puoi utilizzare la casella di ricerca per filtrare l'elenco delle policy.
4. Scegli Operazioni e seleziona Collega.
5. Seleziona una o più identità per collegarle alla policy. Puoi usare la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver selezionato le identità, scegliere Attach policy (Collega policy).

Per usare una policy gestita per impostare un limite delle autorizzazioni (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle policy, scegliere il nome della policy da impostare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Nella pagina dei dettagli della policy, scegli la scheda Entità collegate e quindi, se necessario, apri la sezione Collega come limite delle autorizzazioni e seleziona Imposta la policy come limite delle autorizzazioni.
5. Selezionare uno o più utenti o ruoli su cui utilizzare la policy per un limite delle autorizzazioni. Puoi usare la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver selezionato i principali, scegli Imposta limite autorizzazioni.

Per incorporare una policy inline per un utente o un ruolo (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, selezionare Users (Utenti) o Roles (Ruoli).
3. Nell'elenco, scegli il nome dell'utente o il ruolo in cui incorporare una policy.
4. Scegli la scheda Autorizzazioni.
5. Scegli Aggiungi autorizzazioni, quindi seleziona Crea policy inline.

> **Note**
>
> Non è possibile integrare una policy in linea in un ruolo collegato ai servizi in IAM. Poiché il servizio collegato definisce se puoi modificare le autorizzazioni del ruolo, potresti aggiungere ulteriori policy dalla console di servizio, dall'API o dall' AWS CLI. Per visualizzare la documentazione del ruolo collegato al servizio per un servizio, consulta la pagina AWS servizi supportati da IAM e scegli Yes (Sì) nella colonna Service-Linked Role (Ruolo collegato al servizio) per il servizio.

6. Scegli tra i seguenti metodi per visualizzare i passaggi necessari per creare le tue policy:
   - L'importazione di policy gestite esistenti. È possibile importare una policy gestita all'interno del proprio account e quindi modificare la policy per personalizzarla in base a requisiti specifici. Una policy gestita può essere una policy AWS gestita o una policy gestita dal cliente che hai creato in precedenza.
   - Creazione di policy con l'editor visivo. È possibile creare una nuova policy da zero nell'editor visivo. Se si utilizza l'editor visivo, non è necessario comprendere la sintassi JSON.
   - Creazione di policy utilizzando l'editor JSON: nell'opzione dell'editor JSON, puoi utilizzare la sintassi JSON per creare una policy. È possibile scrivere un nuovo documento di policy JSON o incollare un esempio di policy.
7. Una volta creata, una policy inline viene automaticamente incorporata all'utente o al ruolo.

Come integrare una policy in linea per un gruppo di utenti (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione, seleziona Gruppi di utenti.
3. Nell'elenco, scegli il nome del gruppo di utenti in cui integrare una policy.
4. Seleziona la scheda Autorizzazioni, quindi Aggiungi autorizzazioni e Crea policy in linea.
5. Esegui una delle seguenti operazioni:
   - Seleziona l'opzione Visivo per creare la policy. Per ulteriori informazioni, consulta Creazione di policy con l'editor visivo.
   - Scegli l'opzione JSON per creare la policy. Per ulteriori informazioni, consulta Creazione di policy utilizzando l'editor JSON.
6. Al termine, scegliere Create policy (Crea policy).

Per modificare il limite delle autorizzazioni per una o più entità (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle policy, scegliere il nome della policy da impostare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Nella pagina dei dettagli della policy, scegli la scheda Entità collegate e quindi, se necessario, apri la sezione Collega come limite delle autorizzazioni. Seleziona la casella di controllo accanto agli utenti o ai ruoli i cui limiti devono essere modificati e scegli Modifica.
5. Selezionare una nuova policy per l'utilizzo di un limite delle autorizzazioni. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy. Dopo aver selezionato la policy, scegli Imposta limite autorizzazioni.

Rimozione delle autorizzazioni per le identità IAM (console)

Puoi utilizzare il Console di gestione AWS per rimuovere le autorizzazioni da un'identità (utente, gruppo di utenti o ruolo). A tale scopo, scollega le policy gestite che controllano le autorizzazioni oppure rimuovi la policy che funge da limite delle autorizzazioni. È inoltre possibile eliminare una policy inline.

Per distaccare una policy gestita utilizzata come policy di autorizzazione (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco di policy seleziona il pulsante di opzione accanto al nome della policy da scollegare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Scegli Operazioni, quindi Distacca.
5. Seleziona le identità da distaccare dalla policy. È possibile utilizzare la casella di ricerca per filtrare l'elenco di identità. Dopo aver selezionato le identità, scegliere Detach policy (Scollega policy).

Per rimuovere un limite delle autorizzazioni (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle policy, scegliere il nome della policy da impostare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Nella pagina di riepilogo della policy, scegli la scheda Entità collegate e quindi, se necessario, apri la sezione Collega come limite delle autorizzazioni e scegli le entità da cui rimuovere i limiti delle autorizzazioni. Quindi scegli Rimuovi limite.
5. Confermare la rimozione del limite e scegli Rimuovi limite.

Per eliminare una policy inline (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione scegli Gruppi di utenti, Utenti o Ruoli.
3. Nell'elenco, scegli il nome del gruppo di utenti, dell'utente o del ruolo con la policy da rimuovere.
4. Scegli la scheda Autorizzazioni.
5. Seleziona la casella di controllo accanto alla policy e scegli Rimuovi.

6. Nella finestra di dialogo di conferma seleziona Rimuovi.

Aggiunta di policy IAM (AWS CLI)

Puoi utilizzare il AWS CLI per aggiungere autorizzazioni a un'identità (utente, gruppo di utenti o ruolo). A tale scopo, collega le policy gestite che controllano le autorizzazioni oppure specifica una policy che funga da limite delle autorizzazioni. È inoltre possibile incorporare una policy inline.

Per usare una policy gestita come policy di autorizzazione per un'entità (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy gestita, eseguire i comandi seguenti:
   - Per elencare le policy gestite: aws iam list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: get-policy
2. Per collegare una policy gestita a un'identità (utente, gruppo di utenti o ruolo), utilizza uno dei seguenti comandi:
   - come sono attach-user-policy
   - era io attach-group-policy
   - era io attach-role-policy

Per usare una policy gestita per impostare un limite delle autorizzazioni (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy gestita, eseguire i comandi seguenti:
   - Per elencare le policy gestite: aws iam list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: aws iam get-policy
2. Per usare una policy gestita per impostare il limite delle autorizzazioni per un'entità (utente o ruolo), utilizzare uno dei comandi seguenti:
   - era io put-user-permissions-boundary
   - era io put-role-permissions-boundary

Per incorporare una policy inline (AWS CLI)

Per integrare una policy in linea in un'identità (utente, gruppo o ruolo che non è un ruolo collegato ai servizi), utilizza uno dei seguenti comandi:

- era io put-user-policy
- era io put-group-policy
- era io put-role-policy

Rimozione di policy IAM (AWS CLI)

È possibile utilizzare il AWS CLI per scollegare le politiche gestite che controllano le autorizzazioni o rimuovere una politica che funge da limite delle autorizzazioni. È inoltre possibile eliminare una policy inline.

Per distaccare una policy gestita utilizzata come policy di autorizzazione (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy, eseguire i comandi seguenti:
   - Per elencare le policy gestite: aws iam list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: aws iam get-policy
2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, eseguire i comandi seguenti:
   - Per elencare le identità (utenti IAM, gruppi IAM e ruoli IAM) a cui è collegata una policy gestita:
     - era io list-entities-for-policy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo), usa uno dei comandi seguenti:
     - era io list-attached-user-policies
     - era io list-attached-group-policies
     - era io list-attached-role-policies
3. Per distaccare una policy gestita da un'identità (utente, gruppo di utenti o ruolo), utilizza uno dei seguenti comandi:
   - era io detach-user-policy
   - era io detach-group-policy
   - era io detach-role-policy

Per rimuovere un limite delle autorizzazioni (AWS CLI)

1. (Facoltativo) Per visualizzare la policy gestita attualmente utilizzata per impostare il limite delle autorizzazioni per un utente o ruolo, eseguire i comandi seguenti:

- aws iam get-user
- aws iam get-role

2. (Facoltativo) Per visualizzare gli utenti o i ruoli su cui si utilizza una policy gestita per un limite delle autorizzazioni, eseguire il comando seguente:

   - era io list-entities-for-policy

3. (Facoltativo) Per visualizzare le informazioni su una policy gestita, eseguire i comandi seguenti:

   - Per elencare le policy gestite: aws iam list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: aws iam get-policy

4. Per rimuovere un limite delle autorizzazioni da un utente o ruolo, utilizzare uno dei comandi seguenti:

   - era io delete-user-permissions-boundary
   - era io delete-role-permissions-boundary

Per eliminare una policy inline (AWS CLI)

1. (Opzionale) Per elencare tutte le policy in linea che sono collegate a un'identità (utente, gruppo di utenti, ruolo), usa uno dei seguenti comandi:

   - era io list-user-policies
   - era io list-group-policies
   - era io list-role-policies

2. (Opzionale) Per recuperare un documento di policy in linea che è integrato in un'identità (utente, gruppo di utenti o ruolo), usa uno dei seguenti comandi:

   - era io get-user-policy
   - era io get-group-policy
   - era io get-role-policy

3. Per eliminare una policy in linea da un'identità (utente, gruppo di utenti o ruolo che non è un ruolo collegato ai servizi), usa uno dei seguenti comandi:

   - era io delete-user-policy
   - era io delete-group-policy

- era io delete-role-policy

Aggiungere politiche IAM (AWS API)

Puoi utilizzare l' AWS API per allegare policy gestite che controllano le autorizzazioni o specificare una policy che funga da limite di autorizzazione. È inoltre possibile incorporare una policy inline.

Per utilizzare una politica gestita come politica di autorizzazioni per un'entità (API)AWS

1. (Facoltativo) Per visualizzare le informazioni su una policy, chiamare le operazioni seguenti:
   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy
2. Per collegare una policy gestita a un'identità (utente, gruppo di utenti o ruolo), chiama una delle seguenti operazioni:
   - AttachUserPolicy
   - AttachGroupPolicy
   - AttachRolePolicy

Per utilizzare una politica gestita per impostare un limite di autorizzazioni (API)AWS

1. (Facoltativo) Per visualizzare le informazioni su una policy gestita, chiamare le operazioni seguenti:
   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy
2. Per usare una policy gestita per impostare il limite delle autorizzazioni per un'entità (utente o ruolo), chiamare una delle operazioni seguenti:
   - PutUserPermissionsBoundary
   - PutRolePermissionsBoundary

Per incorporare una policy in linea (API)AWS

Per integrare una policy in un'identità (utente, gruppo o ruolo che non è un ruolo collegato ai servizi), chiama una delle seguenti operazioni:

- PutUserPolicy
- PutGroupPolicy
- PutRolePolicy

Rimuovere le politiche IAM (API)AWS

Puoi utilizzare l' AWS API per scollegare le policy gestite che controllano le autorizzazioni o rimuovere una policy che funge da limite di autorizzazione. È inoltre possibile eliminare una policy inline.

Per scollegare una politica gestita utilizzata come politica di autorizzazioni (API)AWS

1. (Facoltativo) Per visualizzare le informazioni su una policy, chiamare le operazioni seguenti:
   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy
2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, chiamare le operazioni seguenti:
   - Per elencare le identità (utenti IAM, gruppi IAM e ruoli IAM) a cui è collegata una policy gestita:
     - ListEntitiesForPolicy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo), chiama una delle operazioni seguenti:
     - ListAttachedUserPolicies
     - ListAttachedGroupPolicies
     - ListAttachedRolePolicies
3. Per distaccare una policy gestita da un'identità (utente, gruppo di utenti o ruolo), chiama una delle seguenti operazioni:
   - DetachUserPolicy
   - DetachGroupPolicy
   - DetachRolePolicy

Per rimuovere un limite di autorizzazioni (API)AWS

1. (Facoltativo) Per visualizzare la policy gestita attualmente utilizzata per impostare il limite delle autorizzazioni per un utente o ruolo, chiamare le operazioni seguenti:

- GetUser
- GetRole

2. (Facoltativo) Per visualizzare gli utenti o i ruoli su cui si utilizza una policy gestita per un limite delle autorizzazioni, chiamare l'operazione seguente:

   - ListEntitiesForPolicy

3. (Facoltativo) Per visualizzare le informazioni su una policy gestita, chiamare le operazioni seguenti:

   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy

4. Per rimuovere un limite delle autorizzazioni da un utente o ruolo, chiamare una delle operazioni seguenti:

   - DeleteUserPermissionsBoundary
   - DeleteRolePermissionsBoundary

Per eliminare una policy in linea (API)AWS

1. (Opzionale) Per elencare tutte le policy in linea che sono collegate a un'identità (utente, gruppo di utenti, ruolo), chiama una delle seguenti operazioni:

   - ListUserPolicies
   - ListGroupPolicies
   - ListRolePolicies

2. (Opzionale) Per recuperare un documento di policy in linea che è integrato in un'identità (utente, gruppo di utenti o ruolo), chiama una delle seguenti operazioni:

   - GetUserPolicy
   - GetGroupPolicy
   - GetRolePolicy

3. Per eliminare una policy in linea da un'identità (utente, gruppo di utenti o ruolo che non è un ruolo collegato ai servizi), chiama una delle seguenti operazioni:

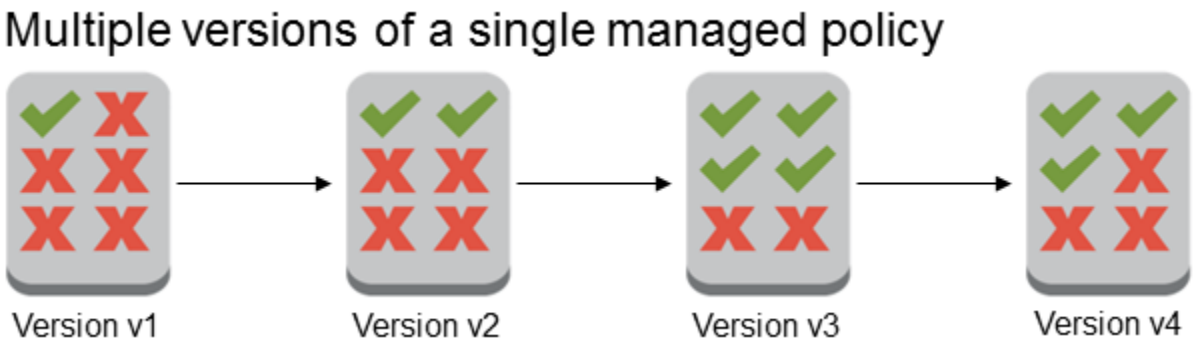
   - DeleteUserPolicy

- DeleteGroupPolicy
- DeleteRolePolicy

Controllo delle versioni delle policy IAM

Quando apporti modifiche a una policy gestita dai clienti IAM e quando AWS apporti modifiche a una policy AWS gestita, la policy modificata non sovrascrive la policy esistente. IAM crea invece una nuova versione della policy gestita. IAM memorizza fino a cinque versioni di una policy gestita dal cliente. IAM non supporta il controllo delle versioni per le policy in linea.

Il diagramma seguente illustra la funzione Versioni multiple per una policy gestita dal cliente. In questo esempio, vengono salvate le versioni 1-4. Puoi salvare fino a cinque versioni delle policy gestite in IAM. Quando si modifica una policy che creerebbe una sesta versione salvata, è possibile scegliere quale versione precedente non memorizzare più. È possibile ripristinare una qualsiasi delle altre quattro versioni salvate in qualsiasi momento.



Una versione di policy è diversa da un elemento `Version` della policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version.

È possibile utilizzare le versioni per tenere traccia delle modifiche apportate a una policy gestita. Ad esempio, è possibile effettuare una modifica a una policy gestita e quindi scoprire che la modifica ha avuto effetti non previsti. In questo caso, è possibile eseguire il rollback a una versione precedente della policy gestita impostando la versione precedente come versione predefinita.

Negli argomenti seguenti viene illustrato come è possibile utilizzare il controllo delle versioni per le policy gestite.

Argomenti

- Limiti della versione

- Utilizzare le versioni per il rollback delle modifiche
- Autorizzazioni per impostare la versione predefinita di una policy
- Impostare la versione predefinita di una policy gestita dal cliente

Limiti della versione

Una policy gestita può avere fino a cinque versioni. Se è necessario apportare modifiche a una policy gestita in più di cinque versioni della AWS Command Line Interface o dell' AWS API, è necessario prima eliminare una o più versioni esistenti. Se si utilizza la Console di gestione AWS, non è necessario eliminare una versione prima di modificare la politica. Quando si salva una sesta versione, verrà visualizzata una finestra di dialogo che richiede di eliminare una o più versioni non predefinite della policy. È possibile visualizzare il documento della policy JSON per ogni versione per facilitare la scelta. Per ulteriori informazioni su questa finestra di dialogo, consultare the section called "Modificare le policy IAM".

È possibile eliminare qualsiasi versione di policy gestita desiderata, ad eccezione della versione predefinita. Quando si elimina una versione, gli identificatori di versione per le versioni rimanenti non vengono modificati. Di conseguenza, gli identificativi di versione potrebbero non essere sequenziali. Ad esempio, se si eliminano le versioni v2 e v4 di una policy gestita e si aggiungono due nuove versioni, gli identificativi della versione rimanenti potrebbero essere v1, v3, v5, v6 e v7.

Utilizzare le versioni per il rollback delle modifiche

È possibile impostare la versione predefinita di una policy gestita dal cliente per eseguire il rollback delle modifiche. Si consideri ad esempio lo scenario riportato di seguito:

Crea una policy gestita dal cliente che consenta agli utenti di amministrare un determinato bucket Amazon S3 utilizzando la Console di gestione AWS. Al momento della creazione, la policy gestita dal cliente ha solo una versione, identificata come v1, in modo che questa versione venga automaticamente impostata come predefinita. La policy funziona come previsto.

Successivamente, si aggiorna la policy per aggiungere l'autorizzazione per amministrare un secondo bucket Amazon S3. IAM crea una nuova versione della policy, identificata come v2, che contiene le modifiche. Imposta la versione v2 come predefinita e poco dopo gli utenti segnalano che non dispongono dell'autorizzazione per utilizzare la console Amazon S3. In questo caso, è possibile ripristinare la versione v1 della policy, che funziona come previsto. Per eseguire questa operazione, è necessario impostare la versione v1 come versione predefinita. Gli utenti sono ora in grado di utilizzare la console Amazon S3 per amministrare il bucket originale.

Successivamente, dopo aver determinato l'errore nella versione v2 della policy, aggiorna nuovamente la policy per aggiungere l'autorizzazione per amministrare il secondo bucket Amazon S3. IAM crea una nuova versione della policy, identificata come v3. Si imposta quindi la versione v3 come predefinita e questa versione funziona come previsto. A questo punto, si elimina la versione v2 della policy.

Autorizzazioni per impostare la versione predefinita di una policy

Le autorizzazioni necessarie per impostare la versione predefinita di una policy corrispondono alle operazioni API di AWS per l'attività. È possibile utilizzare l'operazione API `CreatePolicyVersion` o `SetDefaultPolicyVersion` per impostare la versione predefinita di una policy. Per consentire a un utente di impostare la versione predefinita di una policy esistente, è possibile consentire l'accesso all'operazione `iam:CreatePolicyVersion` o all'operazione `iam:SetDefaultPolicyVersion`. L'operazione `iam:CreatePolicyVersion` consente di creare una nuova versione della policy e di impostarla come predefinita. L'operazione `iam:SetDefaultPolicyVersion` consente di impostare qualsiasi versione esistente della policy come predefinita.

> **Important**
>
> Per impedire a un utente di apportare modifiche alla versione predefinita di una policy, è necessario negare sia `iam:CreatePolicyVersion` sia `iam:SetDefaultPolicyVersion`.

È possibile utilizzare la policy seguente per negare a un utente l'accesso per modificare una policy gestita dal cliente esistente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "iam:CreatePolicyVersion",
                "iam:SetDefaultPolicyVersion"
            ],
            "Resource": "arn:aws:iam::*:policy/POLICY-NAME"
```

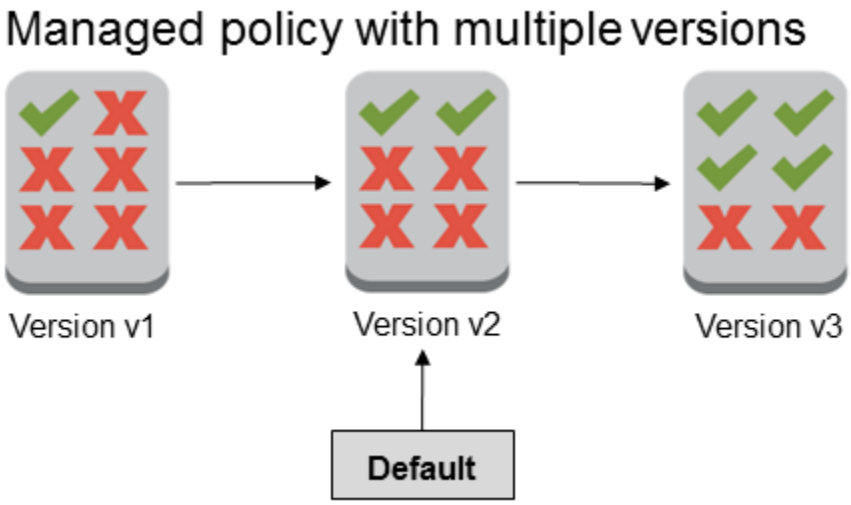
```
        }
    ]
}
```

Impostare la versione predefinita di una policy gestita dal cliente

Una delle versioni di una policy gestita viene impostata come versione predefinita. La versione predefinita della policy è la versione operativa, ovvero, è la versione valida per tutte le entità principali (utenti IAM, gruppi IAM e ruoli IAM) a cui è collegata la policy gestita.

Quando si crea una policy gestita dal cliente, la policy inizia con una singola versione identificata come v1. Per le policy gestite con una sola versione, tale versione viene automaticamente impostata come predefinita. Per le policy gestite dal cliente con più di una versione, selezionare la versione da impostare come predefinita. Per le politiche AWS gestite, la versione predefinita è impostata da AWS. I seguenti diagrammi illustrano questo concetto.





Puoi impostare la versione predefinita di una policy gestita dal cliente per applicarla a tutte le identità IAM (utente, gruppo di utenti e ruolo) a cui è collegata la policy. Non è possibile impostare la versione predefinita per una politica AWS gestita o una politica in linea.

Per impostare la versione predefinita di una policy gestita dal cliente (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle policy, selezionare il nome della policy per cui impostare la versione predefinita. Puoi utilizzare la casella di ricerca per filtrare l'elenco delle policy.
4. Selezionare la scheda Policy versions (Versioni policy). Seleziona la casella di controllo accanto alla versione da impostare come predefinita, quindi scegli Imposta come predefinita.

Per informazioni su come impostare la versione predefinita di una policy gestita dal cliente dall'API AWS Command Line Interface o dall' AWS API, consultaModificare le policy IAM (AWS CLI).

Modificare le policy IAM

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Le policy sono archiviate AWS come documenti JSON e sono allegate ai principali come policy basate sull'identità in IAM. Puoi collegare una policy basata sulle identità a un principale (o identità), ad esempio un gruppo, un utente o un ruolo IAM. Le politiche basate sull'identità includono politiche gestite, politiche AWS gestite dai clienti e politiche in linea. Puoi modificare le politiche gestite dai clienti e le politiche in linea in IAM. AWS le politiche gestite non possono essere

modificate. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

In genere è preferibile utilizzare politiche gestite dal cliente anziché politiche in linea o politiche AWS gestite. AWS le politiche gestite in genere forniscono ampie autorizzazioni amministrative o di sola lettura. Le policy in linea non possono essere riutilizzate su altre identità o gestite al di fuori dell'identità in cui esistono. Per garantire la massima sicurezza, concedere un privilegio minimo, ovvero concedere solo le autorizzazioni necessarie per eseguire attività di processi specifici.

Quando crei o modifichi le policy IAM, AWS puoi eseguire automaticamente la convalida delle policy per aiutarti a creare una policy efficace con il minimo privilegio in mente. Nel Console di gestione AWS, IAM identifica gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce controlli aggiuntivi sulle policy con consigli per aiutarti a perfezionare ulteriormente le tue policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

Puoi utilizzare l' AWS API Console di gestione AWS AWS CLI, o per modificare le policy gestite dai clienti e le policy in linea in IAM. Per ulteriori informazioni sull'utilizzo dei CloudFormation modelli per aggiungere o aggiornare le politiche, consulta il riferimento ai tipi di AWS Identity and Access Management risorse nella Guida per l'CloudFormation utente.

Argomenti

- Modificare le policy IAM (console)
- Modificare le policy IAM (AWS CLI)
- Modifica le politiche IAM (AWS API)

Modificare le policy IAM (console)

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzare il Console di gestione AWS per modificare le politiche gestite dai clienti e le politiche in linea in IAM. AWS le politiche gestite non possono essere modificate. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Per ulteriori informazioni sulla sintassi e sulla struttura della policy, consultare Politiche e autorizzazioni in AWS Identity and Access Management e Documentazione di riferimento degli elementi delle policy JSON IAM.

Prerequisiti

Prima di modificare le autorizzazioni per una policy, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Modifica di policy gestite dal cliente (console)

Puoi modificare le policy gestite dal cliente per cambiare le autorizzazioni in esse definite dalla Console di gestione AWS. Possono esistere fino a cinque versioni di una policy gestita dal cliente. È importante tenere a mente questo limite, perché se apporti modifiche a una policy gestita per più di cinque versioni, la Console di gestione AWS chiede di selezionare una versione da eliminare. Per evitare questo problema, puoi selezionare di modificare la versione predefinita oppure eliminare una versione di una policy prima di apportare modifiche. Per ulteriori informazioni sulle versioni, consultare Controllo delle versioni delle policy IAM.

Console

Per modificare una policy gestita dal cliente

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle policy, selezionare il nome della policy da modificare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Seleziona la scheda Autorizzazioni e scegli Modifica.
5. Esegui una delle seguenti operazioni:
   - Per modificare la policy senza conoscere la sintassi JSON, seleziona l'opzione Visivo. Puoi modificare servizi, operazioni, risorse o condizioni opzionali per ogni blocco di autorizzazione della policy. Inoltre, puoi importare una policy per aggiungere ulteriori autorizzazioni nella parte finale. Al termine, seleziona Successivo per continuare.
   - Seleziona la scheda JSON per modificare la policy, digitando o copiando il testo nella casella JSON. Inoltre, puoi importare una policy per aggiungere ulteriori autorizzazioni nella parte finale. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

6. Nella pagina Verifica e salva, esamina il campo Autorizzazioni definite in questa policy, quindi scegli Salva modifiche per salvare il lavoro.
7. Se esistono già un massimo di cinque versioni della policy gestita, seleziona Salva per visualizzare una finestra di dialogo. Per salvare la nuova versione, la versione non predefinita più vecchia della policy viene rimossa e sostituita con la nuova. Facoltativamente, puoi impostare la nuova versione come versione predefinita della policy.

   Scegli Salva modifiche per salvare la nuova versione della policy.

Impostazione della versione predefinita di una policy gestita dal cliente (console)

Puoi impostare una versione predefinita di una policy gestita dai clienti da Console di gestione AWS. Puoi utilizzare questa policy per stabilire una configurazione di base coerente per le autorizzazioni all'interno dell'organizzazione. Tutti i nuovi allegati della policy utilizzeranno questo insieme standardizzato di autorizzazioni.

Console

Per impostare la versione predefinita di una policy gestita dal cliente (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle policy, selezionare il nome della policy per cui impostare la versione predefinita. Puoi utilizzare la casella di ricerca per filtrare l'elenco delle policy.
4. Selezionare la scheda Policy versions (Versioni policy). Selezionare la casella di controllo a fianco della versione da impostare come predefinita e selezionare Set as default (Imposta come predefinita).

Eliminazione di una versione di una policy gestita dal cliente (console)

Potrebbe essere necessario eliminare una versione di una policy gestita dal cliente al fine di rimuovere le autorizzazioni obsolete o errate che non sono più necessarie o che presentano potenziali rischi per la sicurezza. Mantenendo solo le versioni necessarie, puoi assicurarti di rimanere entro il limite di cinque versioni di policy gestite, lasciando spazio per futuri aggiornamenti e perfezionamenti. È possibile eliminare una versione di una policy gestita dal cliente dalla Console di gestione AWS.

Console

Per eliminare una versione di una policy gestita dal cliente

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Selezionare il nome della policy gestita dal cliente di cui eliminare una versione. Puoi utilizzare la casella di ricerca per filtrare l'elenco delle policy.
4. Selezionare la scheda Policy versions (Versioni policy). Selezionare la casella di controllo accanto alla versione da eliminare. Scegli Elimina.
5. Confermare l'eliminazione della versione e selezionare Delete (Elimina).

Modifica delle policy in linea (console)

Potrebbe essere necessario modificare una policy gestita dal cliente al fine di aggiornare o perfezionare le autorizzazioni concesse, assicurandoti che rimangano in linea con i requisiti di sicurezza e le esigenze di controllo degli accessi in evoluzione dell'organizzazione. La modifica consente di adattare il documento JSON della policy, aggiungendo, modificando o rimuovendo operazioni, risorse o condizioni specifiche per mantenere il principio del privilegio minimo e adattarlo ai cambiamenti dell'ambiente o dei processi. Le policy inline possono essere modificate dalla Console di gestione AWS.

Console

Per modificare una policy inline per un utente, un gruppo di utenti o un ruolo

1. Nel pannello di navigazione scegli Gruppi, Utenti o Ruoli.

2. Scegli il nome dell'utente, del gruppo di utenti o del ruolo con la policy da modificare. Selezionare la scheda Permissions (Autorizzazioni) ed espandere la policy.
3. Per modificare una policy inline, selezionare Edit Policy (Modifica policy).
4. Esegui una delle seguenti operazioni:

   - Per modificare la policy senza conoscere la sintassi JSON, seleziona l'opzione Visivo. Puoi modificare servizi, operazioni, risorse o condizioni opzionali per ogni blocco di autorizzazione della policy. Inoltre, puoi importare una policy per aggiungere ulteriori autorizzazioni nella parte finale. Al termine, seleziona Successivo per continuare.
   - Seleziona la scheda JSON per modificare la policy, digitando o copiando il testo nella casella JSON. Inoltre, puoi importare una policy per aggiungere ulteriori autorizzazioni nella parte finale. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo). Per salvare le modifiche senza influenzare le entità collegate al momento, deselezionare la casella di controllo Save as default version (Salva come versione predefinita).

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

5. Nella pagina Rivedi consulta il riepilogo della policy e scegli Salva modifiche per salvare il lavoro.

Modificare le policy IAM (AWS CLI)

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzare il AWS Command Line Interface (AWS CLI) per modificare le politiche gestite dai clienti e le politiche in linea in IAM. AWS le politiche gestite non possono essere modificate. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Per ulteriori informazioni sulla sintassi e sulla struttura della policy, consultare Politiche e autorizzazioni in AWS Identity and Access Management e Documentazione di riferimento degli elementi delle policy JSON IAM.

Prerequisiti

Prima di modificare le autorizzazioni per una policy, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Modifica di policy gestite dal cliente (AWS CLI)

È possibile modificare una politica gestita dai clienti da AWS CLI.

> **Note**
>
> Una policy gestita può avere fino a cinque versioni. Per apportare modifiche a una policy gestita dal cliente di cui esistono già cinque versioni, devi eliminare una o più versioni esistenti.

Per modificare una policy gestita dal cliente (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy, eseguire i comandi seguenti:
   - Per elencare le policy gestite: list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: get-policy
2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, eseguire i comandi seguenti:
   - Per elencare le identità (utenti IAM, gruppi IAM e ruoli IAM) a cui è collegata una policy gestita:
     - list-entities-for-policy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo):
     - list-attached-user-policies
     - list-attached-group-policies
     - list-attached-role-policies
3. Per modificare una policy gestita dal cliente, eseguire il comando seguente:

- create-policy-version

4. (Facoltativo) Per convalidare una policy gestita dal cliente, esegui il comando IAM Access Analyzer seguente:

   - validate-policy

Impostazione della versione predefinita di una policy gestita dal cliente (AWS CLI)

È possibile impostare una versione predefinita di una politica gestita dai clienti da AWS CLI.

Per impostare la versione predefinita di una policy gestita dal cliente (AWS CLI)

1. (Facoltativo) Per elencare le policy gestite, eseguire il comando seguente:

   - list-policies

2. Per impostare la versione predefinita di una policy gestita dal cliente, eseguire il comando seguente:

   - set-default-policy-version

Eliminazione di una versione di una policy gestita dal cliente (AWS CLI)

È possibile eliminare una versione di una policy gestita dal cliente dalla AWS CLI.

Per eliminare una versione di una policy gestita dal cliente (AWS CLI)

1. (Facoltativo) Per elencare le policy gestite, eseguire il comando seguente:

   - list-policies

2. Per eliminare una policy gestita dal cliente, eseguire il comando seguente:

   - delete-policy-version

Modifica delle policy in linea (AWS CLI)

Le policy inline possono essere modificate dalla AWS CLI.

Per modificare una policy in linea (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy, eseguire i comandi seguenti:
   - Per visualizzare le policy in linea associate a un'identità (utente, gruppo di utenti o ruolo):
     - list-user-policies
     - list-role-policies
     - list-group-policies
   - Per recuperare informazioni dettagliate su una policy in linea:
     - get-user-policy
     - get-role-policy
     - get-group-policy
2. Per modificare una policy in linea, eseguire il comando seguente:
   - put-user-policy
   - put-role-policy
   - put-group-policy
3. (Facoltativo) Per convalidare una policy in linea, esegui il seguente comando del Sistema di analisi degli accessi IAM:
   - validate-policy

Modifica le politiche IAM (AWS API)

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzare l' AWS API per modificare le politiche gestite dai clienti e le politiche in linea in IAM. AWS le politiche gestite non possono essere modificate. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Per ulteriori informazioni sulla sintassi e sulla struttura della policy, consultare Politiche e autorizzazioni in AWS Identity and Access Management e Documentazione di riferimento degli elementi delle policy JSON IAM.

Prerequisiti

Prima di modificare le autorizzazioni per una policy, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Modifica delle politiche gestite dai clienti (AWS API)

Puoi modificare una politica gestita dai clienti utilizzando l' AWS API.

> **Note**
>
> Una policy gestita può avere fino a cinque versioni. Per apportare modifiche a una policy gestita dal cliente di cui esistono già cinque versioni, devi eliminare una o più versioni esistenti.

Per modificare una politica gestita dal cliente (AWS API)

1. (Facoltativo) Per visualizzare le informazioni su una policy, chiamare le operazioni seguenti:
   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy
2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, chiamare le operazioni seguenti:
   - Per elencare le identità (utenti IAM, gruppi IAM e ruoli IAM) a cui è collegata una policy gestita:
     - ListEntitiesForPolicy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo):
     - ListAttachedUserPolicies
     - ListAttachedGroupPolicies
     - ListAttachedRolePolicies
3. Per modificare una policy gestita dal cliente, richiamare la seguente operazione:
   - CreatePolicyVersion
4. (Facoltativo) Per convalidare una policy gestita dal cliente, richiama la seguente operazione IAM Access Analyzer:

- ValidatePolicy

Impostazione della versione predefinita di una politica gestita dal cliente (AWS API)

È possibile impostare una versione predefinita di una politica gestita dai clienti dall' AWS API.

Per impostare la versione predefinita di una politica gestita dal cliente (AWS API)

1. (Facoltativo) Per elencare le policy gestite, richiamare la seguente operazione:
   - ListPolicies
2. Per impostare la versione predefinita di una policy gestita dal cliente, richiamare la seguente operazione:
   - SetDefaultPolicyVersion

Eliminazione di una versione di una politica gestita dal cliente (AWS API)

Puoi eliminare una versione di una politica gestita dai clienti dall' AWS API.

Per eliminare una versione di una politica gestita dai clienti (AWS API)

1. (Facoltativo) Per elencare le policy gestite, richiamare la seguente operazione:
   - ListPolicies
2. Per eliminare una policy gestita dal cliente, chiamare l'operazione seguente:
   - DeletePolicyVersion

Modifica delle politiche in linea (AWS API)

È possibile modificare una politica in linea dall' AWS API.

Per modificare una politica in linea (API)AWS

1. (Facoltativo) Per visualizzare informazioni su una politica in linea, effettuate le seguenti operazioni:
   - Per visualizzare le policy in linea associate a un'identità (utente, gruppo di utenti o ruolo):

- ListUserPolicies
- ListRolePolicies
- ListGroupPolicies

- Per recuperare informazioni dettagliate su una policy in linea:
  - GetUserPolicy
  - GetRolePolicy
  - GetGroupPolicy

2. Per modificare una politica in linea, effettuate le seguenti operazioni:
   - PutUserPolicy
   - PutRolePolicy
   - PutGroupPolicy
3. (Facoltativo) Per convalidare una policy in linea, completa la seguente operazione del Sistema di analisi degli accessi IAM:
   - ValidatePolicy

Eliminare le policy IAM

Puoi eliminare le politiche IAM utilizzando Console di gestione AWS, the AWS Command Line Interface (AWS CLI) o l' AWS API.

> **Note**
>
> L'eliminazione delle policy IAM è definitiva. Una volta eliminata, la policy non potrà più essere ripristinata.

Per ulteriori informazioni sulla sintassi e sulla struttura della policy IAM, consulta Politiche e autorizzazioni in AWS Identity and Access Management e Documentazione di riferimento degli elementi delle policy JSON IAM.

Per ulteriori informazioni sulle differenze tra policy gestite e policy inline, consulta Policy gestite e policy inline.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Argomenti

- Eliminare le policy IAM (console)
- Eliminare le policy IAM (AWS CLI)
- Eliminare le politiche IAM (AWS API)

Eliminare le policy IAM (console)

Puoi utilizzare il Console di gestione AWS per eliminare le politiche gestite dai clienti e le politiche in linea in IAM. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

> **Note**
>
> L'eliminazione delle policy IAM è definitiva. Una volta eliminata, la policy non potrà più essere ripristinata.

Per ulteriori informazioni sulla sintassi e sulla struttura della policy IAM, consulta Politiche e autorizzazioni in AWS Identity and Access Management e Documentazione di riferimento degli elementi delle policy JSON IAM.

Per ulteriori informazioni sulle differenze tra policy gestite e policy inline, consulta Policy gestite e policy inline.

Prerequisiti

Prima di eliminare una policy, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Eliminazione di policy IAM (console)

Potrebbe essere necessario eliminare una policy gestita dal cliente quando diventa obsoleta o non è più in linea con i requisiti di sicurezza e le esigenze di controllo degli accessi dell'organizzazione.

Eliminando le policy non necessarie, si riducono i potenziali rischi per la sicurezza associati a policy obsolete o non utilizzate. Puoi eliminare una policy gestita dal cliente per rimuoverla dal tuo Account AWS. Non è possibile eliminare le policy AWS gestite.

Console

Per eliminare una policy gestita dal cliente

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Seleziona il pulsante di opzione accanto alla policy gestita dal cliente da eliminare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Scegli Azioni, quindi Elimina.
5. Segui le istruzioni per confermare che desideri eliminare la policy, quindi scegli Elimina.

Eliminazione delle policy in linea (console)

Potrebbe essere necessario eliminare una policy inline quando le autorizzazioni specifiche che concede non sono più necessarie per l'utente, il gruppo o il ruolo IAM a cui è direttamente collegata. L'eliminazione delle policy inline non necessarie aiuta a ridurre il rischio di accessi non intenzionali, soprattutto perché tali policy non possono essere riutilizzate o condivise tra più identità come le policy gestite. Puoi eliminare una policy in linea per rimuoverla dal tuo Account AWS. Non è possibile eliminare policy gestite da AWS .

Console

Per eliminare una policy in linea per un utente, un gruppo o un ruolo IAM

1. Nel pannello di navigazione scegli Gruppi di utenti, Utenti o Ruoli.
2. Scegli il nome del gruppo di utenti, dell'utente o del ruolo con la policy da eliminare. Selezionare la scheda Permissions (Autorizzazioni).
3. Seleziona le caselle di controllo accanto alle policy da eliminare, quindi scegli Rimuovi. Quindi, nella finestra di dialogo di conferma, conferma la rimozione e l'eliminazione della policy.

   - Per eliminare una policy in linea in Utenti o Ruoli, scegli Rimuovi per confermare l'eliminazione.

- Se elimini una singola policy in linea in Gruppi di utenti digita il nome della policy e scegli Elimina. Se elimini più policy in linea in Gruppi di utenti, digita il numero di policy da eliminare seguito da **`inline policies`** e scegli Elimina. Ad esempio, se elimini tre policy in linea, digita **`3 inline policies`**.

Eliminare le policy IAM (AWS CLI)

Puoi utilizzare la AWS Command Line Interface (AWS CLI) per eliminare le policy gestite dai clienti e le policy in linea in IAM. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

> **Note**
>
> L'eliminazione delle policy IAM è definitiva. Una volta eliminata, la policy non potrà più essere ripristinata.

Per ulteriori informazioni sulla sintassi e sulla struttura della policy IAM, consulta Politiche e autorizzazioni in AWS Identity and Access Management e Documentazione di riferimento degli elementi delle policy JSON IAM.

Per ulteriori informazioni sulle differenze tra policy gestite e policy inline, consulta Policy gestite e policy inline.

Prerequisiti

Prima di eliminare una policy, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Eliminazione di policy gestite dal cliente (AWS CLI)

Puoi eliminare una policy gestita dal cliente dall' AWS Command Line Interface.

Per eliminare una policy gestita dal cliente (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy, eseguire i comandi seguenti:

   - Per elencare le policy gestite: list-policies

- Per recuperare informazioni dettagliate su una policy gestita: get-policy

2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, eseguire i comandi seguenti:
   - Per visualizzare le identità (utenti IAM, gruppi IAM e ruoli IAM) a cui è collegata una policy gestita, eseguire il comando seguente:
     - list-entities-for-policy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo), esegui uno dei comandi riportati sotto:
     - list-attached-user-policies
     - list-attached-group-policies
     - list-attached-role-policies
3. Per eliminare una policy gestita dal cliente, eseguire il comando seguente:
   - delete-policy

Eliminazione delle policy in linea (AWS CLI)

Una policy in linea può essere eliminata dalla AWS CLI.

Per eliminare una policy inline (AWS CLI)

1. (Opzionale) Per elencare tutte le policy in linea che sono collegate a un'identità (utente, gruppo di utenti, ruolo), usa uno dei seguenti comandi:
   - era iam list-user-policies
   - era io list-group-policies
   - era io list-role-policies
2. (Opzionale) Per recuperare un documento di policy in linea che è integrato in un'identità (utente, gruppo di utenti o ruolo), usa uno dei seguenti comandi:
   - era io get-user-policy
   - era io get-group-policy
   - era io get-role-policy
3. Per eliminare una policy in linea da un'identità (utente, gruppo di utenti o ruolo che non è un ruolo collegato ai servizi), usa uno dei seguenti comandi:

- era io delete-user-policy
- era io delete-group-policy
- era io delete-role-policy

Eliminare le politiche IAM (AWS API)

Puoi utilizzare l' AWS API per eliminare le policy gestite dai clienti e le policy in linea in IAM. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

> **Note**
>
> L'eliminazione delle policy IAM è definitiva. Una volta eliminata, la policy non potrà più essere ripristinata.

Per ulteriori informazioni sulla sintassi e sulla struttura della policy IAM, consulta Politiche e autorizzazioni in AWS Identity and Access Management e Documentazione di riferimento degli elementi delle policy JSON IAM.

Per ulteriori informazioni sulle differenze tra policy gestite e policy inline, consulta Policy gestite e policy inline.

Prerequisiti

Prima di eliminare una policy, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Eliminazione delle politiche gestite dai clienti (AWS API)

Puoi eliminare una politica gestita dai clienti utilizzando l' AWS API.

Per eliminare una politica gestita dal cliente (AWS API)

1. (Facoltativo) Per visualizzare le informazioni su una policy, chiamare le operazioni seguenti:
   - Per elencare le politiche gestite: ListPolicies

- Per recuperare informazioni dettagliate su una politica gestita: GetPolicy

2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, chiamare le operazioni seguenti:
   - Per visualizzare le identità (utenti IAM, gruppi IAM e ruoli IAM) a cui è collegata una policy gestita, chiama la seguente operazione:
     - ListEntitiesForPolicy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo), chiama una delle operazioni seguenti:
     - ListAttachedUserPolicies
     - ListAttachedGroupPolicies
     - ListAttachedRolePolicies
3. Per eliminare una policy gestita dal cliente, chiamare l'operazione seguente:
   - DeletePolicy

Eliminazione delle politiche in linea (API)AWS

È possibile eliminare una politica in linea utilizzando l'API. AWS

Per eliminare una politica in linea (API)AWS

1. (Opzionale) Per elencare tutte le policy in linea che sono collegate a un'identità (utente, gruppo di utenti, ruolo), chiama una delle seguenti operazioni:
   - ListUserPolicies
   - ListGroupPolicies
   - ListRolePolicies
2. (Opzionale) Per recuperare un documento di policy in linea che è integrato in un'identità (utente, gruppo di utenti o ruolo), chiama una delle seguenti operazioni:
   - GetUserPolicy
   - GetGroupPolicy
   - GetRolePolicy
3. Per eliminare una policy in linea da un'identità (utente, gruppo di utenti o ruolo che non è un ruolo collegato ai servizi), chiama una delle seguenti operazioni:

- DeleteUserPolicy
- DeleteGroupPolicy
- DeleteRolePolicy

Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso

In qualità di amministratore, puoi concedere alle risorse IAM (utenti, ruoli, gruppi di utenti o policy) autorizzazioni aggiuntive rispetto a quelle indispensabili. IAM fornisce le informazioni sull'ultimo accesso per facilitare l'identificazione delle autorizzazioni inutilizzate in modo da poterle rimuovere. È possibile utilizzare le informazioni relative all'ultimo accesso per perfezionare le policy e consentire l'accesso solo ai servizi e alle operazioni utilizzati dalle identità IAM. In questo modo è più facile rispettare le best practice dei privilegi minimi. È possibile visualizzare le informazioni relative all'ultimo accesso per le identità o le policy esistenti in IAM o AWS Organizations.

È possibile monitorare continuamente le informazioni relative all'ultimo accesso con analizzatori degli accessi inutilizzati. Per ulteriori informazioni, consulta Risultati relativi agli accessi esterni e inutilizzati.

Argomenti

- Tipi di informazioni sull'ultimo accesso per IAM
- Ultimo accesso alle informazioni per AWS Organizations
- Cose da sapere sulle ultime informazioni di accesso
- Autorizzazioni richieste
- Risolvi i problemi relativi all'attività per IAM e le entità AWS Organizations
- Dove AWS tiene traccia delle ultime informazioni a cui si accede
- Visualizzare le informazioni sull'ultimo accesso per IAM
- Visualizza le informazioni sull'ultimo accesso per AWS Organizations
- Scenari di esempio per l'utilizzo delle ultime informazioni di accesso
- Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM

Tipi di informazioni sull'ultimo accesso per IAM

È possibile visualizzare due tipi di informazioni relative all'ultimo accesso per le identità IAM: informazioni sui servizi AWS consentiti e informazioni sulle operazioni consentite. Le informazioni

includono la data e l'ora in cui è stato effettuato il tentativo di accedere a un' AWS API. Per le operazioni, le informazioni relative all'ultimo accesso riportano le operazioni di gestione del servizio. Le azioni di gestione includono le azioni di creazione, eliminazione e modifica. Per ulteriori informazioni su come visualizzare le informazioni sull'ultimo accesso per IAM, consulta Visualizzare le informazioni sull'ultimo accesso per IAM.

Per esempi di scenari di impiego delle informazioni relative all'ultimo accesso per prendere decisioni sulle autorizzazioni da concedere alle identità IAM, consulta la pagina Scenari di esempio per l'utilizzo delle ultime informazioni di accesso.

Per ulteriori informazioni su come vengono fornite le informazioni per le azioni di gestione, vedere Cose da sapere sulle ultime informazioni di accesso.

Ultimo accesso alle informazioni per AWS Organizations

Se si accede utilizzando le credenziali dell'account di gestione, è possibile visualizzare le informazioni relative all'ultimo accesso al servizio per un' AWS Organizations entità o una politica dell'organizzazione. AWS Organizations le entità includono la radice dell'organizzazione, le unità organizzative (OUs) o gli account. Le informazioni relative all'ultimo accesso AWS Organizations includono informazioni sui servizi consentiti da una policy di controllo dei servizi (SCP). Le informazioni indicano quali principali (utente root, ruolo o utente IAM) di un'organizzazione o un account hanno tentato l'ultimo accesso al servizio e quando. Per ulteriori informazioni sul rapporto e su come visualizzare le informazioni relative all'ultimo accesso AWS Organizations, vedereVisualizza le informazioni sull'ultimo accesso per AWS Organizations.

Ad esempio, scenari per l'utilizzo delle informazioni dell'ultimo accesso per prendere decisioni sulle autorizzazioni da concedere alle AWS Organizations entità, vediScenari di esempio per l'utilizzo delle ultime informazioni di accesso.

Cose da sapere sulle ultime informazioni di accesso

Prima di utilizzare le informazioni dell'ultimo accesso contenute in un report per modificare le autorizzazioni per un'identità o un' AWS Organizations entità IAM, esamina i seguenti dettagli sulle informazioni.

- Periodo di monitoraggio: l'attività recente viene visualizzata nella console IAM entro quattro ore. Il periodo di monitoraggio per le informazioni sul servizio dura almeno 400 giorni, a seconda di quando il servizio ha avviato il monitoraggio delle informazioni sulle operazioni. Il periodo di monitoraggio delle informazioni sulle operazioni Amazon S3 è iniziato il 12 aprile 2020. Il periodo di monitoraggio per le operazioni di Amazon EC2, IAM e Lambda è iniziato il 7 aprile 2021. Il periodo

di monitoraggio per tutti gli altri servizi è iniziato il 23 maggio 2023. Per un elenco dei servizi per i quali sono disponibili le informazioni relative all'ultimo accesso a un'operazione, consulta la pagina Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM. Per ulteriori informazioni sulle regioni per le quali sono disponibili le informazioni relative all'ultimo accesso a un'operazione, consulta la pagina Dove AWS tiene traccia delle ultime informazioni a cui si accede.

- Tentativi segnalati: i dati dell'ultimo accesso al servizio includono tutti i tentativi di accesso a un' AWS API, non solo i tentativi riusciti. Ciò include tutti i tentativi effettuati utilizzando l' Console di gestione AWS AWS API tramite uno qualsiasi degli strumenti a riga di comando o uno qualsiasi degli strumenti a riga di comando. SDKs Una voce non prevista nell'ultimo accesso ai dati del servizio non significa che l'account è stato compromesso, poiché la richiesta potrebbe essere stata rifiutata. Fai riferimento ai tuoi CloudTrail log come fonte autorevole per informazioni su tutte le chiamate API e sull'esito positivo o negativo dell'accesso.
- PassRole— L'`iam:PassRole`azione non viene tracciata e non è inclusa nelle informazioni relative all'ultimo accesso dell'azione IAM.
- Informazioni sull'ultimo accesso all'operazione: le informazioni sull'ultimo accesso a un'operazione sono disponibili per le operazioni di gestione del servizio alle quali le identità IAM hanno eseguito l'accesso. Consulta l'elenco di servizi e delle relative operazioni per scoprire a quale operazione si riferiscono i report relativi all'ultimo accesso.

> **Note**
>
> Le informazioni sull'ultima operazione alla quale è stato effettuato l'accesso non sono disponibili per gli eventi dei piani dati.

- Eventi di gestione: IAM fornisce informazioni sulle azioni per gli eventi di gestione dei servizi registrati da. CloudTrail Talvolta, gli eventi di gestione di CloudTrail sono chiamati anche operazioni del piano di controllo o eventi del piano di controllo. Gli eventi di gestione forniscono visibilità sulle operazioni amministrative eseguite sulle risorse dell'azienda. Account AWS Per ulteriori informazioni sugli eventi di gestione in CloudTrail, vedere Registrazione degli eventi di gestione nella Guida per l'AWS CloudTrail utente.
- Proprietario del report: solo il principale che genera un report può visualizzare i dettagli del report. Ciò significa che quando si visualizzano le informazioni contenute in Console di gestione AWS, potrebbe essere necessario attendere che vengano generate e caricate. Se utilizzi l' AWS API AWS CLI or per ottenere i dettagli del report, le tue credenziali devono corrispondere alle credenziali del responsabile che ha generato il rapporto. Se utilizzi credenziali temporanee per un ruolo o un utente AWS STS federato, devi generare e recuperare il report durante la stessa

sessione. Per ulteriori informazioni sulle entità principal di sessione del ruolo assunto, consulta AWS Elementi della policy JSON: Principale.

- Risorse IAM: le informazioni relative all'ultimo accesso per IAM includono le risorse IAM (ruoli, utenti, gruppi IAM e policy) presenti nell'account. Le ultime informazioni a cui si accede AWS Organizations includono i principali (utenti IAM, ruoli IAM o Utente root dell'account AWS) nell'entità specificata. AWS Organizations Le informazioni relative all'ultimo accesso non includono i tentativi non autenticati.
- Tipi di policy IAM: le informazioni relative all'ultimo accesso per IAM includono i servizi consentiti dalle policy di un'identità IAM. Si tratta delle policy collegate a un ruolo o a un utente direttamente o tramite un gruppo. L'accesso consentito da altri tipi di policy non è incluso nel report. I tipi di policy esclusi includono policy basate sulle risorse, liste di controllo degli accessi, limiti delle autorizzazioni IAM e policy di sessione. AWS Organizations SCPs Le autorizzazioni fornite dai ruoli collegati ai servizi sono definite dal servizio a cui sono collegati e non possono essere modificati in IAM. Per ulteriori informazioni sui ruoli collegati ai servizi, vedere Creare un ruolo collegato ai servizi. Per informazioni sulla valutazione dei diversi tipi di criteri per consentire o negare l'accesso, vedere Logica di valutazione delle policy.
- AWS Organizations tipi di policy: le informazioni per AWS Organizations includono solo i servizi consentiti dalle politiche di controllo dei servizi ereditate da un' AWS Organizations entità (). SCPs SCPs sono politiche collegate a un'unità organizzativa principale, a un'unità organizzativa o a un account. L'accesso consentito da altri tipi di policy non è incluso nel report. I tipi di policy esclusi includono le policy basate sulle risorse, le liste di controllo accessi, i limiti delle autorizzazioni IAM e le policy di sessione. Per ulteriori informazioni su come i diversi tipi di policy vengono valutati per consentire o negare l'accesso, consulta Logica di valutazione delle policy.
- Specificazione di un ID di policy: quando si utilizza l' AWS API AWS CLI or per generare un report per le informazioni dell'ultimo accesso in AWS Organizations, è possibile specificare facoltativamente un ID di policy. Il report risultante include i dati per i servizi consentiti solo da tale policy. Le informazioni includono l'attività più recente dell'account nell' AWS Organizations entità specificata o nei figli dell'entità. Per ulteriori informazioni, consulta aws iam generate-organizations-access-report or GenerateOrganizationsAccessReport.
- AWS Organizations account di gestione: è necessario accedere all'account di gestione dell'organizzazione per visualizzare le informazioni relative all'ultimo accesso al servizio. Puoi scegliere di visualizzare le informazioni relative all'account di gestione utilizzando la console IAM AWS CLI, l'o l' AWS API. Il report risultante elenca tutti i AWS servizi, poiché l'account di gestione non è limitato da SCPs. Se specifichi un ID policy nella CLI o nell'API, la policy viene ignorata. Per

ogni servizio, il report include le informazioni solo per l'account di gestione. Tuttavia, i report per altre AWS Organizations entità non restituiscono informazioni sulle attività nell'account di gestione.

- AWS Organizations impostazioni: un amministratore deve abilitare SCPs nella cartella principale dell'organizzazione prima di poter generare dati per AWS Organizations.

Autorizzazioni richieste

Per visualizzare le ultime informazioni a cui si accede in Console di gestione AWS, è necessario disporre di una politica che conceda le autorizzazioni necessarie.

Autorizzazioni per le informazioni IAM

Per utilizzare la console IAM per visualizzare le informazioni sull'ultimo accesso per un utente, ruolo o policy IAM, devi disporre di una policy che includa le seguenti operazioni:

- `iam:GenerateServiceLastAccessedDetails`
- `iam:Get*`
- `iam:List*`

Queste autorizzazioni consentono a un utente di visualizzare ciò che segue:

- Gli utenti, i gruppi o i ruoli collegati a una policy gestita
- I servizi cui può accedere un utente o ruolo
- L'ultima volta che ha effettuato l'accesso al servizio
- L'ultima volta che hanno provato a utilizzare un'operazione Amazon EC2, IAM, Lambda o Amazon S3 specifica

Per utilizzare l' AWS API AWS CLI or per visualizzare le informazioni dell'ultimo accesso per IAM, devi disporre delle autorizzazioni corrispondenti all'operazione che desideri utilizzare:

- `iam:GenerateServiceLastAccessedDetails`
- `iam:GetServiceLastAccessedDetails`
- `iam:GetServiceLastAccessedDetailsWithEntities`
- `iam:ListPoliciesGrantingServiceAccess`

Questo esempio mostra come creare una policy basata sull'identità che consenta di visualizzare le informazioni sull'ultimo accesso a IAM. Inoltre, consente l'accesso in sola lettura a tutto IAM. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GenerateServiceLastAccessedDetails",
            "iam:Get*",
            "iam:List*"
        ],
        "Resource": "*"
    }
}
```

Autorizzazioni per le informazioni AWS Organizations

Per utilizzare la console IAM per visualizzare un report per il root, l'UO o le entità account in AWS Organizations, devi disporre di una policy che includa le seguenti operazioni:

- `iam:GenerateOrganizationsAccessReport`
- `iam:GetOrganizationsAccessReport`
- `organizations:DescribeAccount`
- `organizations:DescribeOrganization`
- `organizations:DescribeOrganizationalUnit`
- `organizations:DescribePolicy`
- `organizations:ListChildren`
- `organizations:ListParents`
- `organizations:ListPoliciesForTarget`
- `organizations:ListRoots`
- `organizations:ListTargetsForPolicy`

Per utilizzare l' AWS API AWS CLI o per visualizzare le informazioni relative all'ultimo accesso al servizio AWS Organizations, è necessario disporre di una politica che includa le seguenti azioni:

- `iam:GenerateOrganizationsAccessReport`
- `iam:GetOrganizationsAccessReport`
- `organizations:DescribePolicy`
- `organizations:ListChildren`
- `organizations:ListParents`
- `organizations:ListPoliciesForTarget`
- `organizations:ListRoots`
- `organizations:ListTargetsForPolicy`

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta di visualizzare le informazioni relative all'ultimo accesso al servizio. AWS Organizations Inoltre, consente l'accesso in sola lettura a tutti. AWS Organizations Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GenerateOrganizationsAccessReport",
            "iam:GetOrganizationsAccessReport",
            "organizations:Describe*",
            "organizations:List*"
        ],
        "Resource": "*"
    }
}
```

Puoi anche utilizzare la chiave iam: OrganizationsPolicyId condition per consentire la generazione di un report solo per una politica specifica AWS Organizations . Per un esempio di policy, consulta IAM: visualizza le informazioni sull'ultimo accesso al servizio per una AWS Organizations policy.

Risolvi i problemi relativi all'attività per IAM e le entità AWS Organizations

In alcuni casi, Console di gestione AWS l'ultima tabella delle informazioni a cui si accede potrebbe essere vuota. O forse la tua richiesta AWS CLI o AWS l'API restituisce un set di informazioni vuoto o un campo nullo. In questi casi, verifica i problemi seguenti:

- Per le informazioni sull'ultimo accesso, un'azione che si prevede di visualizzare potrebbe non essere restituita nell'elenco. Ciò può accadere perché l'identità IAM non dispone delle autorizzazioni per l'azione o AWS non tiene ancora traccia dell'azione per le ultime informazioni a cui si accede.
- Per un utente IAM, assicurati che disponga di almeno una policy in linea o gestita collegata, direttamente o tramite le appartenenze ai gruppi.
- Per un gruppo IAM, verifica che disponga di almeno una policy in linea o gestita collegata.
- Per un gruppo IAM, il report restituisce solo i dati sull'ultimo accesso al servizio per i membri che hanno utilizzato le policy del gruppo per accedere a un servizio. Per scoprire se un membro ha utilizzato altre policy, esamina i dati sull'ultimo accesso al servizio per tale utente.
- Per un ruolo IAM, verifica che disponga di almeno una policy in linea o gestita collegata.
- Per un'entità IAM (utente o ruolo), esamina altri tipi di policy che potrebbero influenzare le autorizzazioni di tale entità. Questi includono politiche basate sulle risorse, elenchi di controllo degli accessi, AWS Organizations politiche, limiti delle autorizzazioni IAM o politiche di sessione. Per ulteriori informazioni, consulta Tipi di policy o Valutazione delle policy per le richieste all'interno di un singolo account.
- Per una policy IAM, assicurati che la policy gestita specificata sia collegata ad almeno un utente, un gruppo con membri o un ruolo.
- Per un' AWS Organizations entità (root, OU o account), assicurati di aver effettuato l'accesso utilizzando AWS Organizations le credenziali dell'account di gestione.
- Verifica che SCPs siano abilitati nella cartella principale dell'organizzazione.
- Le informazioni sull'ultimo accesso all'azione sono disponibili solo per alcune azioni elencate in Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM.

Quando apporti modifiche, le attività impiegano almeno quattro ore per comparire nel report della console IAM. Se utilizzi l' AWS API AWS CLI o, devi generare un nuovo rapporto per visualizzare le informazioni aggiornate.

Dove AWS tiene traccia delle ultime informazioni a cui si accede

AWS raccoglie le ultime informazioni a cui si accede per le AWS regioni standard. Quando si AWS aggiungono altre regioni, tali regioni vengono aggiunte alla tabella seguente, inclusa la data di AWS inizio del monitoraggio delle informazioni in ciascuna regione.

- Informazioni sul servizio: il periodo di monitoraggio per i servizi dura almeno 400 giorni, o meno se la regione che ha avviato il monitoraggio di questa funzione negli ultimi 400 giorni.
- Informazioni sulle operazioni: il periodo di monitoraggio delle operazioni di gestione Amazon S3 è iniziato il 12 aprile 2020. Il periodo di monitoraggio delle operazioni di gestione Amazon EC2, IAM e Lambda è iniziato il 7 aprile 2021. Il periodo di monitoraggio per le operazioni di gestione di tutti gli altri servizi è iniziato il 23 maggio 2023. Se la data di monitoraggio di una regione è successiva al 23 maggio 2023, le informazioni relative all'ultimo accesso all'operazione da quella regione inizieranno da una data successiva.

Nome Regione	Regione	Data di inizio monitoraggio
Stati Uniti orientali (Ohio)	us-east-2	27 ottobre 2017
Stati Uniti orientali (Virginia settentrionale)	us-east-1	1° ottobre 2015
Stati Uniti occidentali (California settentrionale)	us-west-1	1° ottobre 2015
Stati Uniti occidentali (Oregon)	us-west-2	1° ottobre 2015
Africa (Città del Capo)	af-south-1	22 aprile 2020
Asia Pacific (Hong Kong)	ap-east-1	24 aprile 2019
Asia Pacifico (Hyderabad)	ap-south-2	22 novembre 2022
Asia Pacifico (Giacarta)	ap-southeast-3	13 dicembre 2021
Asia Pacifico (Melbourne)	ap-southeast-4	23 gennaio 2023
Asia Pacifico (Mumbai)	ap-south-1	27 giugno 2016

Nome Regione	Regione	Data di inizio monitoraggio
Asia Pacifico (Osaka-Locale)	ap-northeast-3	11 febbraio 2018
Asia Pacific (Seoul)	ap-northeast-2	6 gennaio 2016
Asia Pacifico (Singapore)	ap-southeast-1	1° ottobre 2015
Asia Pacifico (Sydney)	ap-southeast-2	1° ottobre 2015
Asia Pacifico (Tokyo)	ap-northeast-1	1° ottobre 2015
Canada (Centrale)	ca-central-1	28 ottobre 2017
Europa (Francoforte)	eu-central-1	1° ottobre 2015
Europa (Irlanda)	eu-west-1	1° ottobre 2015
Europe (London)	eu-west-2	28 ottobre 2017
Europe (Milan)	eu-south-1	28 aprile 2020
Europa (Parigi)	eu-west-3	18 dicembre 2017
Europa (Spagna)	eu-south-2	15 novembre 2022
Europa (Stoccolma)	eu-north-1	12 dicembre 2018
Europa (Zurigo)	eu-central-2	8 novembre 2022
Israele (Tel Aviv)	il-central-1	1° agosto 2023
Medio Oriente (Bahrein)	me-south-1	29 luglio 2019
Medio Oriente (Emirati Arabi Uniti)	me-central-1	30 agosto 2022
Sud America (San Paolo)	sa-east-1	11 dicembre 2015
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	1 luglio 2023

Nome Regione	Regione	Data di inizio monitoraggio
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	1 luglio 2023

Se una regione non è presente nella tabella precedente, significa che per tale regione non sono ancora disponibili i dati sull'ultimo accesso al servizio.

Una AWS regione è una raccolta di AWS risorse in un'area geografica. Le regioni sono raggruppate in partizioni. Le Regioni standard sono le Regioni che appartengono alla partizione `aws`. Per ulteriori informazioni sulle diverse partizioni, consulta il formato Amazon Resource Names (ARNs) nel Riferimenti generali di AWS. Per ulteriori informazioni sulle regioni, consulta About AWS Regions anche in. Riferimenti generali di AWS

Visualizzare le informazioni sull'ultimo accesso per IAM

Puoi visualizzare le informazioni sull'ultimo accesso per IAM utilizzando l' AWS API Console di gestione AWS AWS CLI,, o. Consulta un elenco di servizi e delle relative operazioni per i quali vengono visualizzate le informazioni relative all'ultimo accesso. Per ulteriori informazioni sulle ultime informazioni di accesso, vedere Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

È possibile visualizzare le informazioni per i seguenti tipi di risorsa in IAM. In ogni caso, i dati includono i servizi consentiti per il periodo di reporting specificato:

- Utente: visualizza l'ultima volta che l'utente ha tentato di accedere a ogni servizio consentito.
- Gruppo di utenti: visualizza informazioni sull'ultima volta che un membro del gruppo di utenti ha provato ad accedere a ogni servizio consentito. Questo report include anche il numero totale di membri che hanno tentato di accedere.
- Ruolo: visualizza l'ultima volta che qualcuno ha utilizzato il ruolo nel tentativo di accedere a ogni servizio consentito.
- Policy: visualizza le informazioni sull'ultima volta che un utente o un ruolo ha provato ad accedere a ogni servizio consentito. Questo report include anche il numero totale di entità che hanno tentato di accedere.

> **Note**
>
> Prima di visualizzare i dati di accesso per una risorsa in IAM, assicurati di comprendere il periodo di riferimento, le entità incluse nel report e i tipi di policy valutati per i tuoi dati. Per ulteriori dettagli, consultare the section called "Cose da sapere sulle ultime informazioni di accesso".

Visualizzazione delle informazioni per IAM (console)

È possibile visualizzare le informazioni sull'ultimo accesso per IAM nella scheda Ultimo accesso della console IAM.

Come visualizzare le informazioni per IAM (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Gruppi di utenti, Utenti, Ruoli o Policy.
3. Seleziona il nome di qualsiasi utente, gruppo di utenti, ruolo o policy per aprire la relativa pagina Riepilogo e seleziona la scheda Ultimo accesso. Visualizzare le seguenti informazioni, in base alla risorsa scelta:

   - Gruppo di utenti: visualizza l'elenco dei servizi a cui i membri del gruppo di utenti possono accedere. È inoltre possibile visualizzare l'ultima volta che un membro ha effettuato l'accesso al servizio, quali policy di gruppo ha utilizzato e quale membro del gruppo ha effettuato la richiesta. Scegli il nome della policy per scoprire se è una policy gestita o una policy del gruppo di utenti in linea. Scegli il nome del membro del gruppo per visualizzare tutti i membri del gruppo di utenti e il momento in cui hanno effettuato l'ultimo accesso al servizio.
   - Utente: visualizza l'elenco dei servizi a cui l'utente può accedere. È inoltre possibile visualizzare l'ultima volta che hanno effettuato l'accesso al servizio e i criteri associati attualmente all'utente. Scegli il nome della policy per sapere se si tratta di una policy gestita, di una policy utente in linea o di una policy in linea per il gruppo di utenti.
   - Ruolo: visualizzare l'elenco dei servizi cui il ruolo può accedere, il suo ultimo accesso al servizio e le policy utilizzate. Scegliere il nome della policy per scoprire se è una policy gestita o una policy del ruolo inline.
   - Policy: visualizza l'elenco dei servizi con le operazioni consentite nella policy. È inoltre possibile visualizzare l'ultima volta che il criterio è stato utilizzato per accedere al servizio e

l'entità (utente o ruolo) utilizzata dal criterio. La data dell'Ultimo accesso include anche quando viene concesso l'accesso a questa policy tramite un'altra policy. Scegliere il nome dell'entità per scoprire a quali entità è collegata questa policy e l'ultimo accesso al servizio da parte dell'entità.

4. Nella colonna Servizio della tabella, scegli il nome di uno dei servizi che include le informazioni relative all'ultimo accesso a un'operazione per visualizzare un elenco delle operazioni di gestione alle quali le entità IAM hanno provato ad accedere. È possibile visualizzare la Regione AWS e un timestamp che indica l'ultimo tentativo di eseguire l'operazione da parte di un utente.
5. La colonna Ultimo accesso viene visualizzata per i servizi e le operazioni di gestione dei servizi che includono le informazioni relative all'ultimo accesso a un'operazione. Esaminare i seguenti risultati possibili restituiti in questa colonna. Questi risultati variano a seconda che un servizio o un'azione sia consentito, sia stato effettuato l'accesso e che venga registrato l'ultimo accesso alle informazioni a cui si accede. AWS

   <number of> giorni fa

   Numero di giorni dall'utilizzo del servizio o dell'azione nel periodo di registrazione. Il periodo di monitoraggio per i servizi è degli ultimi 400 giorni. Il periodo di monitoraggio delle operazioni Amazon S3 è iniziato il 12 aprile 2020. Il periodo di monitoraggio delle azioni di Amazon EC2, IAM e Lambda è iniziato il 7 aprile 2021. Il periodo di monitoraggio per tutti gli altri servizi è iniziato il 23 maggio 2023. Per ulteriori informazioni sulle date di inizio del monitoraggio per ciascuna di esse Regione AWS, consultaDove AWS tiene traccia delle ultime informazioni a cui si accede.

   Non accessibile nel periodo di tracciabilità

   Il servizio o l'azione tracciati non sono stati utilizzati da un'entità nel periodo di registrazione.

   È possibile disporre delle autorizzazioni per un'azione che non viene visualizzata nell'elenco. Ciò può verificarsi se le informazioni di monitoraggio per l'operazione non sono attualmente incluse da AWS. Non è consigliabile prendere decisioni sulle autorizzazioni basate esclusivamente sull'assenza di informazioni di tracciamento. Si consiglia invece di utilizzare queste informazioni per informare e supportare la strategia generale di concessione di privilegi minimi. Controllare i criteri per verificare che il livello di accesso sia appropriato.

Visualizzazione delle informazioni per IAM (AWS CLI)

Puoi utilizzare il AWS CLI per recuperare informazioni sull'ultima volta che una risorsa IAM è stata utilizzata per tentare di accedere ai AWS servizi e alle azioni Amazon S3, Amazon EC2, IAM e Lambda. Una risorsa IAM può essere un utente, un gruppo di utenti, un ruolo o una policy.

Come visualizzare le informazioni per IAM (AWS CLI)

1. Generare un report. La richiesta deve includere l'ARN della risorsa IAM (utente, gruppo di utenti, ruolo o policy) per cui desideri un report. È possibile specificare il livello di granularità che si desidera generare nel report per visualizzare i dettagli di accesso per i servizi o per entrambi i servizi e le azioni. Viene restituito un `job-id` che è possibile utilizzare nelle operazioni `get-service-last-accessed-details` e `get-service-last-accessed-details-with-entities` per monitorare `job-status` finché il processo viene completato.

   - aws iam -details generate-service-last-accessed

2. Recuperare i dettagli sul report utilizzando il parametro `job-id` dal passaggio precedente.

   - aws iam get-service-last-accessed -details

   Questa operazione restituisce le seguenti informazioni, a seconda del tipo di risorsa richiesto nell'operazione `generate-service-last-accessed-details`:

   - Utente: restituisce un elenco dei servizi cui l'utente specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo dell'utente e l'ARN dell'utente.
   - Gruppo di utenti: restituisce un elenco dei servizi a cui i membri del gruppo di utenti specificato possono accedere utilizzando la policy collegata al gruppo di utenti. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo effettuato da qualsiasi membro del gruppo di utenti. Inoltre, restituisce l'ARN dell'utente e il numero totale di membri del gruppo di utenti che hanno provato ad accedere al servizio. Usa l'GetServiceLastAccessedDetailsWithEntitiesoperazione per recuperare un elenco di tutti i membri.
   - Ruolo: restituisce un elenco dei servizi cui il ruolo specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo del ruolo e l'ARN del ruolo.
   - Policy: restituisce un elenco dei servizi per i quali la policy specificata consente l'accesso. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo di accesso al

servizio da parte di un'entità (utente o ruolo), utilizzando la policy. Inoltre, restituisce l'ARN di quell'entità e il numero totale di entità che hanno tentato di accedere.

3. Scopri di più sulle entità che hanno utilizzato autorizzazioni di policy o gruppo di utenti in un tentativo di accesso a un servizio specifico. Questa operazione restituisce un elenco delle entità insieme all'ARN, l'ID, il nome, il percorso, il tipo (utente o ruolo) e l'ultimo tentativo di accesso al servizio di ogni entità. È anche possibile utilizzare questa operazione per gli utenti e i ruoli, ma restituisce informazioni solo su tale entità.

   - era iam get-service-last-accessed - details-with-entities

4. Scopri di più sulle policy basate sull'identità utilizzate da un'identità (utente, gruppo di utenti o ruolo) in un tentativo di accesso a un servizio specifico. Quando si specifica un'identità e un servizio, questa operazione restituisce un elenco delle policy di autorizzazione che l'identità può utilizzare per accedere al servizio specificato. Questa operazione fornisce lo stato attuale delle policy e non dipende dal report generato. Non restituisce neanche altri tipi di policy, come le policy basate sulle risorse, le liste di controllo accessi, le policy di AWS Organizations , i limiti delle autorizzazioni IAM o le policy di sessione. Per ulteriori informazioni, consulta Tipi di policy o Valutazione delle policy per le richieste all'interno di un singolo account.

   - aws iam list-policies-granting-service -access

Visualizzazione delle informazioni per IAM (AWS API)

Puoi utilizzare l' AWS API per recuperare informazioni sull'ultima volta che una risorsa IAM è stata utilizzata per tentare di accedere ai AWS servizi e alle azioni Amazon S3, Amazon EC2, IAM e Lambda. Una risorsa IAM può essere un utente, un gruppo di utenti, un ruolo o una policy. È possibile specificare il livello di granularità da generare nel report per visualizzare i dettagli relativi ai servizi o ai servizi e alle azioni.

Per visualizzare le informazioni per IAM (API)AWS

1. Generare un report. La richiesta deve includere l'ARN della risorsa IAM (utente, gruppo di utenti, ruolo o policy) per cui desideri un report. Viene restituito un `JobId` che è possibile utilizzare nelle operazioni `GetServiceLastAccessedDetails` e `GetServiceLastAccessedDetailsWithEntities` per monitorare il `JobStatus` finché il processo viene completato.

   - GenerateServiceLastAccessedDetails

2. Recuperare i dettagli sul report utilizzando il parametro `JobId` dal passaggio precedente.

- GetServiceLastAccessedDetails

Questa operazione restituisce le seguenti informazioni, a seconda del tipo di risorsa richiesto nell'operazione `GenerateServiceLastAccessedDetails`:

- Utente: restituisce un elenco dei servizi cui l'utente specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo dell'utente e l'ARN dell'utente.
- Gruppo di utenti: restituisce un elenco dei servizi a cui i membri del gruppo di utenti specificato possono accedere utilizzando la policy collegata al gruppo di utenti. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo effettuato da qualsiasi membro del gruppo di utenti. Inoltre, restituisce l'ARN dell'utente e il numero totale di membri del gruppo di utenti che hanno provato ad accedere al servizio. Utilizza l'GetServiceLastAccessedDetailsWithEntitiesoperazione per recuperare un elenco di tutti i membri.
- Ruolo: restituisce un elenco dei servizi cui il ruolo specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo del ruolo e l'ARN del ruolo.
- Policy: restituisce un elenco dei servizi per i quali la policy specificata consente l'accesso. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo di accesso al servizio da parte di un'entità (utente o ruolo), utilizzando la policy. Inoltre, restituisce l'ARN di quell'entità e il numero totale di entità che hanno tentato di accedere.

3. Scopri di più sulle entità che hanno utilizzato autorizzazioni di policy o gruppo di utenti in un tentativo di accesso a un servizio specifico. Questa operazione restituisce un elenco delle entità insieme all'ARN, l'ID, il nome, il percorso, il tipo (utente o ruolo) e l'ultimo tentativo di accesso al servizio di ogni entità. È anche possibile utilizzare questa operazione per gli utenti e i ruoli, ma restituisce informazioni solo su tale entità.

   - GetServiceLastAccessedDetailsWithEntities

4. Scopri di più sulle policy basate sull'identità utilizzate da un'identità (utente, gruppo di utenti o ruolo) in un tentativo di accesso a un servizio specifico. Quando si specifica un'identità e un servizio, questa operazione restituisce un elenco delle policy di autorizzazione che l'identità può utilizzare per accedere al servizio specificato. Questa operazione fornisce lo stato attuale delle policy e non dipende dal report generato. Non restituisce neanche altri tipi di policy, come le policy basate sulle risorse, le liste di controllo accessi, le policy di AWS Organizations , i limiti delle autorizzazioni IAM o le policy di sessione. Per ulteriori informazioni, consulta Tipi di policy o Valutazione delle policy per le richieste all'interno di un singolo account.

- ListPoliciesGrantingServiceAccess

Visualizza le informazioni sull'ultimo accesso per AWS Organizations

Puoi visualizzare le informazioni sull'ultimo accesso al servizio AWS Organizations utilizzando la console IAM o AWS l'API. AWS CLI Per informazioni importanti sui dati, sulle autorizzazioni necessarie, sulla risoluzione dei problemi e sulle regioni supportate, consulta Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

Quando accedi alla console IAM utilizzando le credenziali dell'account di AWS Organizations gestione, puoi visualizzare le informazioni per qualsiasi entità della tua organizzazione. AWS Organizations le entità includono la radice dell'organizzazione, le unità organizzative (OU) e gli account. Inoltre, puoi utilizzare la console IAM per visualizzare le informazioni per eventuali policy di controllo dei servizi (SCP) nell'organizzazione. IAM mostra un elenco di servizi consentiti da qualsiasi SCP applicabile all'entità. Per ogni servizio, puoi visualizzare le informazioni più recenti sull'attività dell'account per l' AWS Organizations entità scelta o per i figli dell'entità.

Quando utilizzi l' AWS API AWS CLI or con le credenziali dell'account di gestione, puoi generare un report per qualsiasi entità o policy della tua organizzazione. Un report programmatico per un'entità include un elenco di servizi consentiti da qualsiasi SCP applicabile all'entità. Per ciascun servizio, il report include l'attività più recente per gli account nell'entità AWS Organizations specificata o nella sottostruttura dell'entità.

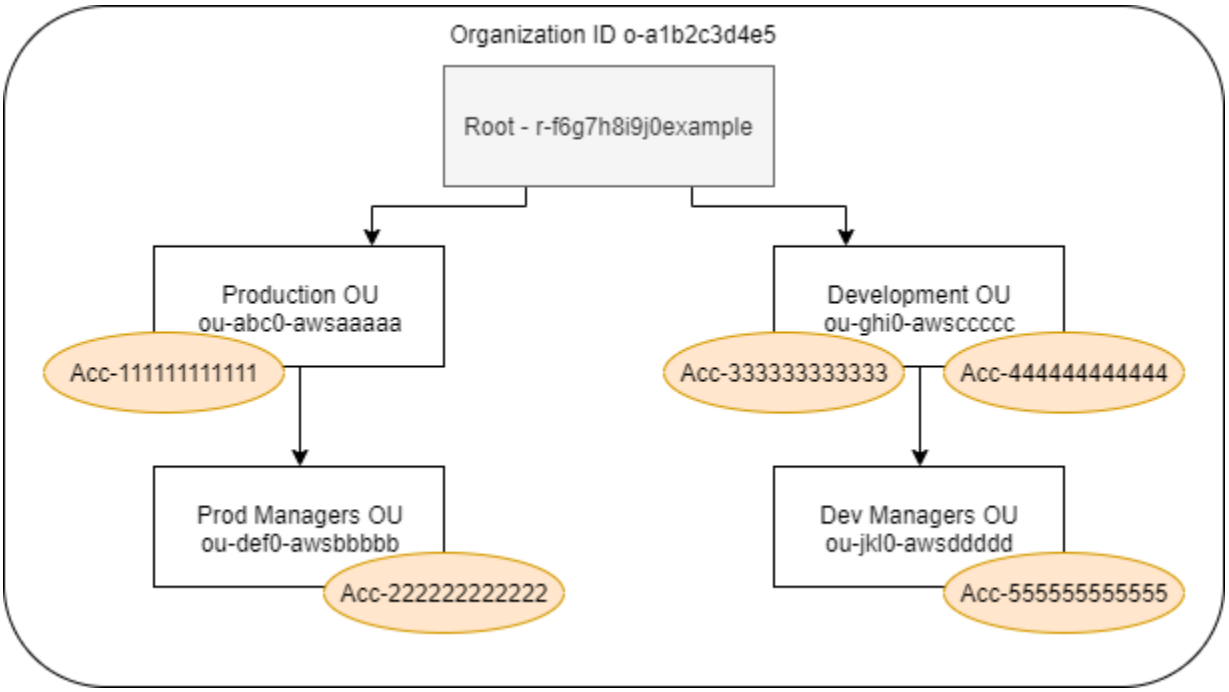
Quando si genera un report programmatico per una policy, è necessario specificare un' AWS Organizations entità. Questo report include un elenco di servizi consentiti dalla SCP specificata. Per ogni servizio, include l'attività dell'account più recente nell'entità o negli elementi figlio dell'entità a cui è concessa l'autorizzazione da tale policy. Per ulteriori informazioni, consulta aws iam generate-organizations-access-report o. GenerateOrganizationsAccessReport

Prima di visualizzare il report, assicurati di aver compreso i requisiti e i dati dell'account di gestione, il periodo del report, le entità incluse nel report e i tipi di policy valutati. Per ulteriori dettagli, consultare the section called “Cose da sapere sulle ultime informazioni di accesso”.

Comprendi il AWS Organizations percorso dell'entità

Quando si utilizza l' AWS API AWS CLI or per generare un report di AWS Organizations accesso, è necessario specificare un percorso dell'entità. Un percorso è una rappresentazione in testo della struttura di un'entità AWS Organizations .

È possibile creare un percorso di entità utilizzando la struttura nota dell'organizzazione. Ad esempio, supponiamo di avere la seguente struttura organizzativa in AWS Organizations.



Il percorso per l'unità organizzativa Dev Manager viene creato utilizzando gli ID dell'organizzazione, il root e tutte le unità organizzative presenti nel percorso fino all'unità organizzativa inclusa.

```
o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-ghi0-awsccccc/ou-jkl0-awsddddd/
```

Il percorso dell'account nell'unità organizzativa Production (Produzione) viene creato utilizzando gli ID dell'organizzazione, il root, l'unità organizzativa e il numero di account.

```
o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-abc0-awsaaaaa/111111111111/
```

> **Note**
>
> Gli ID organizzazione sono univoci a livello globale, ma gli ID delle unità organizzative e gli ID root sono univoci solo all'interno di un'organizzazione. Ciò significa che non ci sono due organizzazioni che condividono lo stesso ID organizzazione. Tuttavia, un'altra organizzazione potrebbe avere un'unità organizzativa o un root con il tuo stesso ID. Si consiglia di includere sempre l'ID organizzazione quando si specifica un'unità organizzativa o un root.

Visualizzazione delle informazioni per AWS Organizations (console)

Puoi utilizzare la console IAM per visualizzare le informazioni sull'ultimo accesso al servizio per la root, l'unità organizzativa, l'account o la policy.

Per visualizzare le informazioni relative alla radice (console)

1. Accedi alle credenziali dell'account di AWS Organizations gestione Console di gestione AWS utilizzando e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione nella sezione Access reports (Report di accesso), scegliere Attività organizzazione.
3. Nella pagina Organization activity (Attività dell'organizzazione), scegliere Root.
4. Nella scheda Details and activity (Dettagli e attività), visualizzare la sezione Service access report (Report di accesso al servizio). I dati includono un elenco di servizi consentiti dalle policy collegate direttamente alla root. I dati mostrano da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione. Per maggiori dettagli su quale principale ha avuto accesso al servizio, accedi come amministratore a tale account e visualizza le informazioni sull'ultimo accesso al servizio IAM.
5. Scegli la scheda SCP collegate per visualizzare l'elenco di policy di controllo dei servizi (SCP) collegate alla root. IAM mostra il numero di entità di destinazione a cui è collegata ogni policy. È possibile utilizzare queste informazioni per decidere quali SCP rivedere.
6. Scegliere il nome di una SCP per visualizzare tutti i servizi consentiti dalla policy. Per ogni servizio, visualizzare da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione.
7. Scegli Modifica AWS Organizations per visualizzare ulteriori dettagli e modificare l'SCP nella AWS Organizations console. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS Organizations .

Per visualizzare le informazioni relative a un'unità organizzativa o a un conto (console)

1. Accedi alle credenziali dell'account di AWS Organizations gestione Console di gestione AWS utilizzando e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione nella sezione Access reports (Report di accesso), scegliere Attività organizzazione.

3. Nella pagina Organization activity (Attività dell'organizzazione), espandere la struttura dell'organizzazione. Quindi sceglie il nome dell'unità organizzativa o qualsiasi account che si desidera visualizzare, tranne l'account di gestione.
4. Nella scheda Details and activity (Dettagli e attività), visualizzare la sezione Service access report (Report di accesso al servizio). I dati includono un elenco di servizi consentiti dalle SCP collegate all'unità organizzativa o all'account e a tutti i relativi elementi padre. I dati mostrano da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione. Per maggiori dettagli su quale principale ha avuto accesso al servizio, accedi come amministratore a tale account e visualizza le informazioni sull'ultimo accesso al servizio IAM.
5. Scegli la scheda SCP collegate per visualizzare l'elenco di policy di controllo dei servizi (SCP) collegate direttamente all'unità organizzativa o all'account. IAM mostra il numero di entità di destinazione a cui è collegata ogni policy. È possibile utilizzare queste informazioni per decidere quali SCP rivedere.
6. Scegliere il nome di una SCP per visualizzare tutti i servizi consentiti dalla policy. Per ogni servizio, visualizzare da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione.
7. Scegli Modifica AWS Organizations per visualizzare ulteriori dettagli e modificare l'SCP nella AWS Organizations console. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS Organizations .

Come visualizzare le informazioni relative all'account di gestione (console)

1. Accedi alle credenziali dell'account di AWS Organizations gestione Console di gestione AWS utilizzando e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione nella sezione Access reports (Report di accesso), scegliere Attività organizzazione.
3. Nella pagina Attività dell'organizzazione, espandi la struttura dell'organizzazione e scegli il nome dell'account di gestione.
4. Nella scheda Details and activity (Dettagli e attività), visualizzare la sezione Service access report (Report di accesso al servizio). Le informazioni includono un elenco di tutti i servizi AWS . L'account di gestione non è limitato dalle SCP. I dati mostrano se l'account ha effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione. Per maggiori dettagli su quale principale ha avuto accesso al servizio, accedi come amministratore a tale account e visualizza le informazioni sull'ultimo accesso al servizio IAM.

5. Seleziona la scheda SCP collegate per verificare che non siano presenti SCP collegate, dato che si tratta dell'account di gestione.

Per visualizzare le informazioni relative a un criterio (console)

1. Accedi alle credenziali dell'account di AWS Organizations gestione Console di gestione AWS utilizzando e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione nella sezione Access reports (Report di accesso), scegliere Policy di controllo dei servizi (SCP).
3. Nella pagina Service control policies (SCPs) (Policy di controllo dei servizi (SCP)), visualizzare un elenco delle policy nell'organizzazione. È possibile visualizzare il numero di entità di destinazione a cui è collegata ogni policy.
4. Scegliere il nome di una SCP per visualizzare tutti i servizi consentiti dalla policy. Per ogni servizio, visualizzare da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione.
5. Scegli Modifica AWS Organizations per visualizzare ulteriori dettagli e modificare l'SCP nella AWS Organizations console. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS Organizations .

Visualizzazione delle informazioni per AWS Organizations (AWS CLI)

È possibile utilizzare AWS CLI per recuperare le informazioni a cui si è effettuato l'ultimo accesso al servizio per la AWS Organizations root, l'unità organizzativa, l'account o la policy.

Per visualizzare AWS Organizations informazioni sull'ultimo accesso al servizio (AWS CLI)

1. Usa le credenziali AWS Organizations del tuo account di gestione con l'IAM e AWS Organizations le autorizzazioni richiesti e conferma che gli SCP siano abilitati per il tuo root. Per ulteriori informazioni, consulta Cose da sapere sulle ultime informazioni di accesso.
2. Generare un report. La richiesta deve includere il percorso dell' AWS Organizations entità (root, OU o account) per la quale desideri un report. È anche possibile includere un parametro `organization-policy-id` per visualizzare un report per una policy specifica. Il comando restituisce `job-id` che è quindi possibile utilizzare nel comando `get-organizations-access-report` per monitorare `job-status` fino al completamento del processo.
   - aws iam generate-organizations-access-report

3. Recuperare i dettagli sul report utilizzando il parametro `job-id` dal passaggio precedente.

   - aws iam get-organizations-access-report

   Questo comando restituisce un elenco di servizi a cui i membri dell'entità possono accedere. Per ogni servizio, il comando restituisce la data e l'ora dell'ultimo tentativo di un membro dell'account e il percorso dell'entità dell'account. Inoltre, restituisce il numero totale di servizi disponibili per l'accesso e il numero di servizi a cui non è stato effettuato l'accesso. Se è stato specificato il parametro `organizations-policy-id` facoltativo, i servizi disponibili per l'accesso sono quelli consentiti dalla policy specificata.

Visualizzazione delle informazioni per AWS Organizations (AWS API)

Puoi utilizzare l' AWS API per recuperare le informazioni di ultimo accesso al servizio relative alla tua AWS Organizations root, all'unità organizzativa, all'account o alla policy.

Per visualizzare AWS Organizations informazioni sull'ultimo accesso al servizio (AWS API)

1. Usa le credenziali AWS Organizations del tuo account di gestione con l'IAM e AWS Organizations le autorizzazioni richiesti e conferma che gli SCP siano abilitati per la tua root. Per ulteriori informazioni, consulta Cose da sapere sulle ultime informazioni di accesso.
2. Generare un report. La richiesta deve includere il percorso dell' AWS Organizations entità (root, OU o account) per la quale desideri un report. È anche possibile includere un parametro `OrganizationsPolicyId` per visualizzare un report per una policy specifica. L'operazione restituisce `JobId` che è quindi possibile utilizzare nell'operazione `GetOrganizationsAccessReport` per monitorare `JobStatus` fino al completamento del processo.

   - GenerateOrganizationsAccessReport
3. Recuperare i dettagli sul report utilizzando il parametro `JobId` dal passaggio precedente.

   - GetOrganizationsAccessReport

   Questa operazione restituisce un elenco di servizi a cui i membri dell'entità possono accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo di un membro dell'account e il percorso dell'entità dell'account. Inoltre, restituisce il numero totale di servizi disponibili per l'accesso e il numero di servizi a cui non è stato effettuato l'accesso. Se è stato

specificato il parametro `OrganizationsPolicyId` facoltativo, i servizi disponibili per l'accesso sono quelli consentiti dalla policy specificata.

Scenari di esempio per l'utilizzo delle ultime informazioni di accesso

Puoi utilizzare le informazioni di ultimo accesso per prendere decisioni sulle autorizzazioni da concedere alle tue entità o AWS Organizations entità IAM. Per ulteriori informazioni, consulta Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

> **Note**
>
> Prima di visualizzare le informazioni di accesso per un'entità o una policy in IAM or AWS Organizations, assicurati di aver compreso il periodo di riferimento, le entità segnalate e i tipi di policy valutati per i tuoi dati. Per ulteriori dettagli, consultare the section called "Cose da sapere sulle ultime informazioni di accesso".

È compito dell'amministratore determinare il giusto equilibrio tra accessibilità e privilegio minimo appropriato per l'azienda.

Utilizzo delle informazioni per ridurre le autorizzazioni per un gruppo IAM

Puoi utilizzare le informazioni sull'ultimo accesso per ridurre le autorizzazioni per un gruppo IAM in modo da includere solo i servizi necessari agli utenti. Questo metodo è una fase importante nella concessione del privilegio minimo a livello di servizio.

Ad esempio, Paulo Santos è l'amministratore responsabile della definizione delle autorizzazioni AWS utente per Example Corp. Questa società ha appena iniziato a utilizzare AWS e il team di sviluppo software non ha ancora definito quali AWS servizi utilizzerà. Paulo vuole concedere al team l'autorizzazione ad accedere solo ai servizi necessari, ma poiché non sono stati ancora definiti, ha concesso temporaneamente le autorizzazioni di power user. Quindi utilizza le ultime informazioni a cui si accede per ridurre le autorizzazioni del gruppo.

Paulo crea una policy gestita denominata `ExampleDevelopment`, utilizzando il seguente testo in formato JSON. La collega poi a un gruppo denominato `Development` e aggiunge tutti gli sviluppatori al gruppo.

> **Note**
>
> I power user di Paulo potrebbero avere bisogno di autorizzazioni `iam:CreateServiceLinkedRole` per utilizzare alcuni servizi e funzionalità. È consapevole che l'aggiunta di questa autorizzazione consente agli utenti di creare qualsiasi ruolo collegato al servizio. Accetta questo rischio per i suoi power user.

JSON

```
{

    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccessToAllServicesExceptPeopleManagement",
            "Effect": "Allow",
            "NotAction": [
                "iam:*",
                "organizations:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RequiredIamAndOrgsActions",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "organizations:DescribeOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

Paulo decide di attendere 90 giorni prima di visualizzare i dati sull'ultimo accesso al servizio per il gruppo `Development` utilizzando la Console di gestione AWS. Visualizza l'elenco dei servizi a cui i membri del gruppo hanno effettuato l'accesso. Viene a sapere che gli utenti hanno avuto accesso

a cinque servizi nell'ultima settimana: Amazon CloudWatch Logs AWS CloudTrail, Amazon EC2 e Amazon S3. AWS KMS Hanno avuto accesso ad alcuni altri servizi durante la prima valutazione AWS, ma non da allora.

Paulo decide di ridurre le autorizzazioni delle policy per includere solo quei cinque servizi e lo IAM e AWS Organizations le azioni richiesti. Modifica la policy `ExampleDevelopment` utilizzando il seguente testo in formato JSON.

> **Note**
>
> I power user di Paulo potrebbero avere bisogno di autorizzazioni `iam:CreateServiceLinkedRole` per utilizzare alcuni servizi e funzionalità. È consapevole che l'aggiunta di questa autorizzazione consente agli utenti di creare qualsiasi ruolo collegato al servizio. Accetta questo rischio per i suoi power user.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccessToListedServices",
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "kms:*",
                "cloudtrail:*",
                "logs:*",
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RequiredIamAndOrgsActions",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "organizations:DescribeOrganization"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

Per ridurre ulteriormente le autorizzazioni, Paulo può visualizzare gli eventi dell'account in Event history (Cronologia eventi) in AWS CloudTrail . Qui può visualizzare informazioni dettagliate sugli eventi, che può utilizzare per ridurre le autorizzazioni della policy in modo da includere solo le operazioni e le risorse di cui hanno bisogno gli sviluppatori. Per ulteriori informazioni, vedere Visualizzazione CloudTrail degli eventi nella CloudTrail console nella Guida per l'AWS CloudTrail utente.

Utilizzo delle informazioni per ridurre le autorizzazioni per un utente IAM

Puoi utilizzare le informazioni sull'ultimo accesso per ridurre le autorizzazioni per un singolo utente IAM.

Ad esempio, Martha Rivera è un'amministratore IT responsabile di garantire che le persone della sua azienda non dispongano di autorizzazioni in eccesso AWS . Come parte di un controllo periodico di sicurezza, l'amministratore esamina le autorizzazioni di tutti gli utenti IAM. Uno di questi utenti, Nikhil Jayashankar, è uno sviluppatore di applicazioni, che in precedenza ha ricoperto il ruolo di tecnico della sicurezza. A causa del cambiamento dei requisiti della mansione, Nikhil è membro sia del gruppo `app-dev` sia del gruppo `security-team`. Il gruppo `app-dev` per la sua nuova mansione concede autorizzazioni per più servizi, tra cui Amazon EC2, Amazon EBS, Auto Scaling, Amazon S3, Route 53 e Elastic Transcoder. Il `security-team` gruppo per il suo vecchio lavoro concede le autorizzazioni a IAM e. CloudTrail

L'amministratore Martha accede alla console IAM e seleziona Utenti, quindi sceglie il nome `nikhilj` e seleziona la scheda Ultimo accesso.

Martha esamina la colonna Ultimo accesso e nota che Nikhil non ha recentemente effettuato l'accesso a IAM, Route 53 CloudTrail, Amazon Elastic Transcoder e a una serie di altri servizi. AWS Nikhil ha effettuato l'accesso ad Amazon S3. Martha sceglie S3 dall'elenco dei servizi e apprende che Nikhil ha eseguito alcune azioni `List` di Amazon S3 nelle ultime due settimane. All'interno della sua azienda, Martha conferma che Nikhil non ha alcuna necessità aziendale di accedere a IAM e che non è CloudTrail più un membro del team di sicurezza interno.

Martha è ora pronta ad agire in base al servizio e all'azione dell'ultimo accesso alle informazioni. Tuttavia, diversamente da quanto avviene con il gruppo dell'esempio precedente, un utente IAM

come nikhilj potrebbe essere soggetto a più policy ed essere membro di più gruppi. Martha deve procedere con cautela per evitare di interrompere inavvertitamente l'accesso per nikhilj o per altri membri del gruppo. Oltre a conoscere il tipo di accesso di cui Nikhil deve disporre, deve determinare il modo in cui Nikhil riceve le autorizzazioni.

Martha sceglie la scheda Permissions (Autorizzazioni), dove visualizza le policy collegate direttamente a nikhilj e quelle collegate da un gruppo. Espande ogni policy e visualizza il riepilogo per scoprire quale policy consente l'accesso ai servizi che Nikhil non sta utilizzando:

- IAM — La policy IAMFullAccess AWS gestita è collegata direttamente nikhilj e collegata al gruppo. security-team
- CloudTrail — La policy AWS CloudTrailReadOnlyAccess AWS gestita è allegata al security-team gruppo.
- Route 53: la policy gestita dal cliente App-Dev-Route53 è collegata al gruppo app-dev.
- Elastic Transcoder: la policy gestita dal cliente App-Dev-ElasticTranscoder è collegata al gruppo app-dev.

Martha decide di rimuovere la politica IAMFullAccess AWS gestita a cui è collegata nikhilj direttamente. Rimuove inoltre l'appartenenza di Nikhil al gruppo security-team. Queste due azioni rimuovono l'accesso non necessario a IAM e CloudTrail.

Le autorizzazioni di Nikhil per accedere a Route 53 ed Elastic Transcoder sono concesse dal gruppo app-dev. Anche se Nikhil non li utilizza, questi servizi potrebbero essere utili ad altri membri del gruppo. Martha esamina le informazioni sull'ultimo accesso per il gruppo app-dev e scopre che diversi membri hanno recentemente effettuato l'accesso a Route 53 e Amazon S3. Ma nessun membro del gruppo ha avuto accesso a Elastic Transcoder nell'ultimo anno. Rimuove quindi dal gruppo la policy gestita dal cliente App-Dev-ElasticTranscoder.

Martha esamina poi i dati sull'ultimo accesso al servizio per la policy gestita dal cliente App-Dev-ElasticTranscoder. Scopre che la policy non è collegata a nessun'altra identità IAM. Indaga all'interno della sua azienda per accertarsi che la policy non sarà necessaria in futuro e quindi la elimina.

Utilizzo delle informazioni prima dell'eliminazione delle risorse IAM

Puoi utilizzare le informazioni sull'ultimo accesso al servizio prima di eliminare una risorsa IAM per assicurarti che sia trascorso un determinato intervallo di tempo dall'ultimo utilizzo di tale risorsa.

Questo si applica a utenti, gruppi, ruoli e policy. Per ulteriori informazioni su queste operazioni, consulta i seguenti argomenti:

- Utenti IAM: rimuovere o disattivare un utente IAM
- Gruppi: eliminare un gruppo IAM
- Ruoli: eliminare ruoli o profili dell'istanza
- Policy: eliminazione delle policy IAM (comporta inoltre lo scollegamento della policy dalle identità)

Utilizzo delle informazioni prima della modifica delle policy IAM

Puoi esaminare le informazioni sull'ultimo accesso per un'identità IAM (utente, gruppo o ruolo) o per una policy IAM prima di modificare una policy che influisce sulla risorsa. È un'opzione importante per non rimuovere l'accesso per qualcuno che la utilizza.

Ad esempio, Arnav Desai è sviluppatore e AWS amministratore di Example Corp. Quando il suo team ha iniziato a utilizzare AWS, ha concesso a tutti gli sviluppatori l'accesso tramite power user che ha consentito loro l'accesso completo a tutti i servizi tranne IAM e. AWS Organizations Come primo passo verso la concessione di privilegi minimi, Arnav desidera utilizzare l' AWS CLI per esaminare le policy gestite nel suo account.

A tale scopo, Arnav elenca innanzitutto le policy di autorizzazione gestite dal cliente nel suo account che sono collegate a un'identità, utilizzando il seguente comando:

```
aws iam list-policies --scope Local --only-attached --policy-usage-filter
 PermissionsPolicy
```

Dalla risposta, acquisisce l'ARN per ogni policy. Arnav genera quindi un report relativo ai dati sull'ultimo accesso al servizio per ogni policy, utilizzando il comando seguente.

```
aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:policy/
ExamplePolicy1
```

Da questa risposta, acquisisce l'ID del report generato dal campo `JobId`. Arnav testa il comando seguente finché il campo `JobStatus` restituisce un valore `COMPLETED` o `FAILED`. Se il processo ha esito negativo, acquisisce l'errore.

```
aws iam get-service-last-accessed-details --job-id 98a765b4-3cde-2101-2345-example678f9
```

Quando il job ha lo stato di`COMPLETED`, Arnav analizza il contenuto dell'array. JSON-formatted `ServicesLastAccessed`

```
"ServicesLastAccessed": [
       {
           "TotalAuthenticatedEntities": 1,
           "LastAuthenticated": 2018-11-01T21:24:33.222Z,
           "ServiceNamespace": "dynamodb",
           "LastAuthenticatedEntity": "arn:aws:iam::123456789012:user/IAMExampleUser",
           "ServiceName": "Amazon DynamoDB"
       },

       {
           "TotalAuthenticatedEntities": 0,
           "ServiceNamespace": "ec2",
           "ServiceName": "Amazon EC2"
       },

       {
           "TotalAuthenticatedEntities": 3,
           "LastAuthenticated": 2018-08-25T15:29:51.156Z,
           "ServiceNamespace": "s3",
           "LastAuthenticatedEntity": "arn:aws:iam::123456789012:role/IAMExampleRole",
           "ServiceName": "Amazon S3"
       }
   ]
```

Da queste informazioni, Arnav apprende che la policy `ExamplePolicy1` consente l'accesso a tre servizi: Amazon DynamoDB, Amazon S3 e Amazon EC2. L'utente IAM denominato `IAMExampleUser` ha provato accedere l'ultima volta a DynamoDB il 1° novembre e qualcuno ha utilizzato il ruolo `IAMExampleRole` per provare ad accedere ad Amazon S3 il 25 agosto. Ci sono anche altre due entità che hanno provato ad accedere ad Amazon S3 nell'ultimo anno. Tuttavia, nessuno ha provato ad accedere ad Amazon EC2 nell'ultimo anno.

Questo significa che Arnav può rimuovere in modo sicuro le operazioni Amazon EC2 dalla policy. Arnav vuole esaminare l'attuale documento JSON per la policy. In primo luogo, deve determinare il numero di versione della policy utilizzando il comando seguente.

```
aws iam list-policy-versions --policy-arn arn:aws:iam::123456789012:policy/
ExamplePolicy1
```

Dalla risposta, Arnav raccoglie l'attuale numero di versione predefinita dall'array `Versions`. Utilizza quindi quel numero di versione (`v2`) per richiedere il documento JSON della policy utilizzando il comando seguente.

```
aws iam get-policy-version --policy-arn arn:aws:iam::123456789012:policy/ExamplePolicy1
 --version-id v2
```

Arnav memorizza il documento JSON della policy restituito nel campo `Document` dell'array `PolicyVersion`. Nel documento della policy, Arnav ricerca le operazioni nel namespace `ec2`. Se non ci sono operazioni dagli altri spazi dei nomi rimanenti nella policy, scollega la policy dalle identità interessate (utenti, gruppi e ruoli) e la elimina. In questo caso, la policy include i servizi Amazon DynamoDB e Amazon S3. Pertanto, Arnav rimuove le operazioni Amazon EC2 dal documento e salva le modifiche. Utilizza quindi il seguente comando per aggiornare la policy con la nuova versione del documento e impostare tale versione come versione predefinita della policy.

```
aws iam create-policy-version --policy-arn arn:aws:iam::123456789012:policy/
ExamplePolicy1 --policy-document file://UpdatedPolicy.json --set-as-default
```

La policy `ExamplePolicy1` è ora aggiornata per rimuovere l'accesso al servizio Amazon EC2 non necessario.

Altri scenari IAM

Le informazioni sull'ultimo tentativo di accesso di una risorsa IAM (utente, gruppo, ruolo o policy) a un servizio possono essere utili per completare una delle seguenti attività:

- Policy: Modifica di una policy esistente in linea o gestita dal cliente per rimuovere le autorizzazioni
- Policy: Conversione di una policy in linea in una policy gestita e successiva eliminazione
- Policy: Aggiunta di un rifiuto esplicito a una policy esistente
- Policy: Scollegamento di una policy gestita da un'identità (utente, gruppo o ruolo)
- Entità: Impostazione di un limite di autorizzazioni per controllare il numero massimo di autorizzazioni che un'entità (utente o ruolo) può avere
- Gruppi: Rimozione di utenti da un gruppo

Utilizzo dei dati per perfezionare le autorizzazioni di un'unità organizzativa

Puoi utilizzare i dati sull'ultimo accesso al servizio per perfezionare le autorizzazioni per un'unità organizzativa in AWS Organizations.

Ad esempio, John Stiles è un amministratore. AWS Organizations È responsabile di garantire che le persone in azienda Account AWS non dispongano di autorizzazioni eccessive. Come parte di un audit di sicurezza periodico, esamina le autorizzazioni dell'organizzazione. La sua unità organizzativa `Development` contiene gli account che vengono spesso utilizzati per testare nuovi servizi AWS . John decide di controllare periodicamente il report per servizi a cui non è stato effettuato alcun accesso da più di 180 giorni. Quindi, rimuove le autorizzazioni concesse ai membri dell'unità organizzativa per accedere a questi servizi.

John esegue l'accesso alla console IAM utilizzando le sue credenziali dell'account di gestione. Nella console IAM, individua i AWS Organizations dati per l'`Development`unità organizzativa. Esamina la tabella dei report sugli accessi ai servizi e vede due AWS servizi a cui non accede da più del periodo che preferisce, ossia 180 giorni. Ricorda di aver aggiunto le autorizzazioni per i team di sviluppo per accedere ad Amazon Lex e. AWS Database Migration Service John contatta i team di sviluppo e conferma che non hanno più l'esigenza aziendale di testare questi servizi.

John è pronto a usare le ultime informazioni di accesso. Sceglie Modifica in AWS Organizations e gli viene segnalato che la SCP è collegata a più entità. Sceglie Continue (Continua). Nel AWS Organizations, esamina gli obiettivi per scoprire a quali AWS Organizations entità è collegato l'SCP. Tutte le entità si trovano all'interno dell'unità organizzativa `Development`.

John decide di negare l'accesso ad Amazon Lex e AWS Database Migration Service le azioni nell'`NewServiceTest`SCP. Questa operazione rimuove l'accesso non necessario ai servizi.

Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM

La tabella seguente elenca i AWS servizi per i quali vengono visualizzate le informazioni sull'ultimo accesso all'azione IAM. Per un elenco delle azioni in ogni servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi nel Service Authorization Reference.

AWS fornisce le informazioni sull'ultimo accesso all'azione in formato JSON per semplificare l'automazione dei flussi di lavoro di gestione delle policy. Con le informazioni di riferimento del servizio, è possibile accedere alle informazioni relative all'ultima azione a cui si è avuto accesso direttamente da file leggibili Servizi AWS da computer. Per ulteriori informazioni, consulta Simplified Servizio AWS information for programmatic access nella documentazione di riferimento sull'autorizzazione dei servizi.

Servizio	Prefisso del servizio
AWS Identity and Access Management e Access Analyzer	access-analyzer
Gestione dell'account AWS	account
AWS Certificate Manager	acm
Flussi di lavoro gestiti da Amazon per Apache Airflow	airflow
AWS Amplify	amplify
AWS Amplify Generatore di interfacce utente	amplifyuibuilder
Amazon AppIntegrations	app-integrations
AWS AppConfig	appconfig
Amazon AppFlow	appflow
AWS Application Cost Profiler	application-cost-profiler
Informazioni approfondite sulle CloudWatch applicazioni Amazon	applicationinsights
AWS App Mesh	appmesh
WorkSpaces Applicazioni Amazon	appstream
AWS AppSync	appsync
Amazon Managed Service per Prometheus	aps
Amazon Athena	atena
AWS Audit Manager	auditmanager
AWS Auto Scaling	autoscaling
Marketplace AWS	aws-marketplace

Servizio	Prefisso del servizio
AWS Backup	backup
AWS Batch	batch
Amazon Braket	braket
Budget AWS	budgets
AWS Cloud9	cloud9
AWS CloudFormation	cloudformation
Amazon CloudFront	cloudfront
AWS CloudHSM	cloudhsm
Amazon CloudSearch	cloudsearch
AWS CloudTrail	cloudtrail
Amazon CloudWatch	cloudwatch
AWS CodeArtifact	codeartifact
AWS CodeDeploy	codedeploy
Amazon CodeGuru Profiler	codeguru-profiler
CodeGuru Revisore Amazon	codeguru-reviewer
AWS CodePipeline	codepipeline
AWS CodeStar	codestar
Notifiche AWS CodeStar	codestar-notifications
Amazon Cognito Identity	cognito-identity

Servizio	Prefisso del servizio
Pool di utenti Amazon Cognito	cognito-idp
Amazon Cognito Sync	cognito-sync
Amazon Comprehend Medical	comprehendmedical
AWS Compute Optimizer	compute-optimizer
AWS Config	config
Cliente Amazon Connect	connect
AWS Cost and Usage Report	cur
AWS Glue DataBrew	databrew
AWS Data Exchange	dataexchange
AWS Data Pipeline	datapipeline
DynamoDB Accelerator	dax
AWS Device Farm	devicefarm
Amazon DevOps Guru	devops-guru
AWS Direct Connect	directconnect
Amazon Data Lifecycle Manager	dlm
AWS Database Migration Service	dms
Cluster elastici Amazon DocumentDB	docdb-elastic
Amazon DynamoDB	dynamodb
Amazon Elastic Block Store	ebs

Servizio	Prefisso del servizio
Amazon Elastic Compute Cloud	ec2
Amazon Elastic Container Registry	ecr
Amazon Elastic Container Registry Public	ecr-public
Amazon Elastic Container Service	ecs
Amazon Elastic Kubernetes Service	eks
Amazon ElastiCache	elasticache
AWS Elastic Beanstalk	elasticbeanstalk
Amazon Elastic File System	elasticfilesystem
Elastic Load Balancing	elasticloadbalancing
Amazon EMR su EKS (Containers EMR)	emr-containers
Amazon EMR Serverless	emr-serverless
OpenSearch Servizio Amazon	es
Amazon EventBridge	eventi
Amazon FinSpace	finspace
Amazon Data Firehose	firehose
AWS Fault Injection Service	fis
AWS Firewall Manager	fms
Amazon Fraud Detector	frauddetector
Amazon FSx	fsx

Servizio	Prefisso del servizio
GameLift Server Amazon	gamelift
Servizio di posizione Amazon	geo
Amazon Glacier	glacier
Grafana gestito da Amazon	grafana
AWS IoT Greengrass	greengrass
AWS Ground Station	groundstation
Amazon GuardDuty	guardduty
AWS HealthLake	healthlake
Amazon Honeycode	honeycode
AWS Identity and Access Management	iam
AWS Archivio di identità	identitystore
EC2 Image Builder	imagebuilder
Amazon Inspector Classic	inspector
Amazon Inspector	inspector2
AWS IoT	iot
AWS IoT Analytics	iotanalytics
AWS IoT Core Device Advisor	iotdeviceadvisor
AWS IoT Events	iotevents
AWS IoT Fleet Hub	iotfleethub
AWS IoT SiteWise	iotsitewise

Servizio	Prefisso del servizio
AWS IoT TwinMaker	iottwinmaker
Wireless AWS IoT	iotwireless
Amazon Interactive Video Service	ivs
Amazon Interactive Video Service Chat	ivschat
Amazon Managed Streaming per Apache Kafka	kafka
Amazon Managed Streaming per Kafka Connect	kafkaconnect
Amazon Kendra	kendra
Amazon Kinesis	kinesis
Amazon Kinesis Analytics V2	kinesisanalytics
AWS Key Management Service	kms
AWS Lambda	lambda
Amazon Lex V2	lex
AWS License Manager Gestore di abbonamenti Linux	license-manager-linux-subscriptions
Amazon Lightsail	lightsail
CloudWatch Registri Amazon	log
Amazon Lookout per le apparecchiature	lookoutequipment
Amazon Lookout per le metriche	lookoutmetrics
Modernizzazione del mainframe AWS	m2
Blockchain gestita da Amazon	managedblockchain

Servizio	Prefisso del servizio
AWS Elemental MediaConnect	mediaconnect
AWS Elemental MediaConvert	mediaconvert
AWS Elemental MediaLive	medialive
AWS Elemental MediaTailor	mediatailor
Amazon MemoryDB	memorydb
AWS Transform MGN	mgn
AWS Migration Hub	mgh
AWS Suggerimenti di strategia dell'Hub di migrazione	migration hub-strategy
Amazon Pinpoint	mobiletargeting
Amazon MQ	mq
AWS Network Manager	networkmanager
Amazon Nimble Studio	nimble
AWS HealthOmics	omics
AWS OpsWorks	opsworks
AWS OpsWorks CM	opsworks-cm
AWS Outposts	outposts
AWS Organizations	organizations
AWS Performance Insights	pi
EventBridgeTubi Amazon	pipes

Servizio	Prefisso del servizio
Amazon Polly	polly
Profili dei clienti Amazon Connect	profile
Amazon QLDB	qldb
AWS Resource Access Manager	ram
AWS Cestino di riciclaggio	rbin
Amazon Relational Database Service	rds
Amazon Redshift	redshift
API dati di Amazon Redshift	redshift-data
AWS Migration Hub Refactor Spaces	refactor-spaces
Amazon Rekognition	rekognition
AWS Resilience Hub	resiliencehub
Esploratore di risorse AWS	resource-explorer-2
AWS Resource Groups	resource-groups
AWS RoboMaker	robomaker
AWS Identity and Access Management Ruoli ovunque	rolesanywhere
Amazon Route 53	route53
Controlli di ripristino Amazon Route 53	route53-recovery-control-config
Preparazione al ripristino di Amazon Route 53	route53-recovery-readiness

Servizio	Prefisso del servizio
Amazon Route 53 Resolver	route53resolver
AWS CloudWatchRUM	rum
Amazon Simple Storage Service	s3
Amazon S3 su Outposts	s3-outposts
Funzionalità geospaziali di Amazon SageMaker AI	sagemaker-geospatial
Savings Plans	savingsplans
EventBridgeSchemi Amazon	schemas
Amazon SimpleDB	sdb
AWS Secrets Manager	secretsmanager
AWS Security Hub CSPM	securityhub
Amazon Security Lake	securitylake
AWS Serverless Application Repository	serverlessrepo
AWS Service Catalog	servicecatalog
AWS Cloud Map	servicediscovery
Service Quotas	servicequotas
Amazon Simple Email Service	ses
AWS Shield	shield
AWS Signer	signer
AWS SimSpace Weaver	simspaceweaver

Servizio	Prefisso del servizio
AWS Server Migration Service	sms
Servizio di SMS e messaggi vocali Amazon Pinpoint	sms-voice
AWS Snowball Edge	snowball
Amazon Simple Queue Service	sqs
AWS Systems Manager	ssm
Strumento di gestione degli incidenti AWS Systems Manager	ssm-incidents
AWS Systems Manager per SAP	ssm-sap
AWS Step Functions	states
Servizio di token di sicurezza AWS	sts
Amazon Simple Workflow Service	swf
Amazon CloudWatch Synthetics	synthetics
AWS Resource Groups Tagging API	tag
Amazon Textract	textract
Amazon Timestream	timestream
AWS Costruttore di reti di telecomunicazioni	tnb
Amazon Transcribe	transcribe
AWS Transfer Family	transfer
Amazon Translate	translate
Connect Customer Voice ID	voiceid
Amazon VPC Lattice	vpc-lattice

Servizio	Prefisso del servizio
AWS WAFV2	wafv2
AWS Well-Architected Tool	wellarchitected
La saggezza dei clienti di Amazon Connect	wisdom
Amazon WorkLink	worklink
Amazon WorkSpaces	workspace
AWS X-Ray	xray

Operazioni per le quali sono disponibili le informazioni relative all'ultimo accesso

La tabella seguente elenca le operazioni per le quali sono disponibili le informazioni relative all'ultimo accesso all'operazione stessa.

> **Important**
>
> L'operazione `iam:UpdateAccountName` sarà dichiarata obsoleta il 22 aprile 2026. Dopo il 22 aprile 2026, l'accesso all'aggiornamento del nome dell'account sarà controllato solo dall'autorizzazione `account:PutAccountName`. Ti consigliamo vivamente di aggiornare tutte le policy di controllo dei servizi che controllano gli aggiornamenti dei nomi degli account di modo che utilizzino l'autorizzazione `account:PutAccountName`.

Prefisso del servizio	Azioni
access-analyzer	analizzatore di accesso: ApplyArchiveRule
	analizzatore di accesso: CancelPolicyGeneration
	analizzatore di accesso: CheckAccessNotGranted
	analizzatore di accesso: CheckNoNewAccess

Prefisso del servizio	Azioni
	analizzatore di accesso: CheckNoPublicAccess
	analizzatore di accesso: CreateAccessPreview
	analizzatore di accesso: CreateAnalyzer
	analizzatore di accesso: CreateArchiveRule
	analizzatore di accesso: DeleteAnalyzer
	analizzatore di accesso: DeleteArchiveRule
	analizzatore di accesso: GenerateFindingRecommendation
	analizzatore di accesso: GetAccessPreview
	analizzatore di accesso: GetAnalyzedResource
	analizzatore di accesso: GetAnalyzer
	analizzatore di accesso: GetArchiveRule
	analizzatore di accesso: GetFinding
	analizzatore di accesso: GetFindingRecommendation
	analizzatore di accesso: GetFindingsStatistics
	analizzatore di accesso: GetGeneratedPolicy
	analizzatore di accesso: ListAccessPreviewFindings
	analizzatore di accesso: ListAccessPreviews
	analizzatore di accesso: ListAnalyzedResources
	analizzatore di accesso: ListAnalyzers
	analizzatore di accesso: ListArchiveRules
	analizzatore di accesso: ListFindings

Prefisso del servizio	Azioni
	analizzatore di accesso: ListPolicyGenerations
	analizzatore di accesso: StartPolicyGeneration
	analizzatore di accesso: StartResourceScan
	analizzatore di accesso: UpdateAnalyzer
	analizzatore di accesso: UpdateArchiveRule
	analizzatore di accesso: UpdateFindings
	analizzatore di accesso: ValidatePolicy

Prefisso del servizio	Azioni
account	conto: AcceptPrimaryEmailUpdate conto: DeleteAlternateContact conto: DisableRegion conto: EnableRegion conto: GetAccountInformation conto: GetAlternateContact conto: GetContactInformation conto: GetGovCloudAccountInformation conto: GetPrimaryEmail conto: GetRegionOptStatus conto: ListRegions conto: PutAccountName conto: PutAlternateContact conto: PutContactInformation conto: StartPrimaryEmailUpdate

Prefisso del servizio	Azioni
acm	macchina fotografica: CreateAcmeEndpoint acm: DeleteCertificate acm: DescribeCertificate acm: ExportCertificate acm: GetAccountConfiguration acm: GetCertificate acm: ImportCertificate acm: ListAcmeEndpoints acm: ListCertificates acm: PutAccountConfiguration acm: RenewCertificate acm: RequestCertificate acm: ResendValidationEmail acm: SearchCertificates acm: UpdateCertificateOptions

Prefisso del servizio	Azioni
airflow	flusso d'aria: CreateCliToken flusso d'aria: CreateEnvironment flusso d'aria: CreateWebLoginToken flusso d'aria: DeleteEnvironment flusso d'aria: GetEnvironment flusso d'aria: ListEnvironments flusso d'aria: PublishMetrics flusso d'aria: UpdateEnvironment

Prefisso del servizio	Azioni
amplify	amplificare: CreateApp
	amplificare: CreateBackendEnvironment
	amplificare: CreateBranch
	amplificare: CreateDeployment
	amplificare: CreateDomainAssociation
	amplificare: CreateWebHook
	amplificare: DeleteApp
	amplificare: DeleteBackendEnvironment
	amplificare: DeleteBranch
	amplificare: DeleteDomainAssociation
	amplificare: DeleteJob
	amplificare: DeleteWebHook
	amplificare: GenerateAccessLogs
	amplificare: GetApp
	amplificare: GetArtifactUrl
	amplificare: GetBackendEnvironment
	amplificare: GetBranch
	amplificare: GetDomainAssociation
	amplificare: GetJob
	amplificare: GetWebHook
	amplificare: ListApps

Prefisso del servizio	Azioni
	amplificare: ListArtifacts
	amplificare: ListBackendEnvironments
	amplificare: ListBranches
	amplificare: ListDomainAssociations
	amplificare: ListJobs
	amplificare: ListWebHooks
	amplificare: StartDeployment
	amplificare: StartJob
	amplificare: StopJob
	amplificare: UpdateApp
	amplificare: UpdateBranch
	amplificare: UpdateDomainAssociation
	amplificare: UpdateWebHook

Prefisso del servizio	Azioni
amplifyuibuilder	amplifyuibuilder: CreateComponent amplificyuibuilder: CreateForm amplificyuibuilder: CreateTheme amplificyuibuilder: DeleteComponent amplificyuibuilder: DeleteForm amplificyuibuilder: DeleteTheme amplificyuibuilder: ExportComponents amplificyuibuilder: ExportThemes amplificyuibuilder: GetCodegenJob amplificyuibuilder: ListCodegenJobs amplificyuibuilder: ListComponents amplificyuibuilder: ListForms amplificyuibuilder: ListThemes amplificyuibuilder: ResetMetadataFlag amplificyuibuilder: StartCodegenJob amplificyuibuilder: UpdateComponent amplificyuibuilder: UpdateForm amplificyuibuilder: UpdateTheme

Prefisso del servizio	Azioni
app-integrations	integrazioni con app: CreateApplication
	integrazioni con app: CreateDataIntegration
	integrazioni con app: CreateDataIntegrationAssociation
	integrazioni con app: CreateEventIntegration
	integrazioni con app: DeleteApplication
	integrazioni con app: DeleteDataIntegration
	integrazioni con app: DeleteEventIntegration
	integrazioni con app: GetApplication
	integrazioni con app: GetDataIntegration
	integrazioni con app: GetEventIntegration
	integrazioni con app: ListApplicationAssociations
	integrazioni con app: ListApplications
	integrazioni con app: ListDataIntegrationAssociations
	integrazioni con app: ListDataIntegrations
	integrazioni con app: ListEventIntegrationAssociations
	integrazioni con app: ListEventIntegrations
	integrazioni con app: UpdateApplication
	integrazioni con app: UpdateDataIntegration
	integrazioni con app: UpdateDataIntegrationAssociation
	integrazioni con app: UpdateEventIntegration

Prefisso del servizio	Azioni
appconfig	appconfig: CreateApplication app config: CreateConfigurationProfile app config: CreateDeploymentStrategy app config: CreateEnvironment app config: CreateExtension app config: CreateExtensionAssociation app config: CreateHostedConfigurationVersion app config: DeleteApplication app config: DeleteConfigurationProfile app config: DeleteDeploymentStrategy app config: DeleteEnvironment app config: DeleteExtension app config: DeleteExtensionAssociation app config: DeleteHostedConfigurationVersion app config: GetAccountSettings app config: GetApplication app config: GetConfiguration app config: GetConfigurationProfile app config: GetDeployment app config: GetDeploymentStrategy app config: GetEnvironment

Prefisso del servizio	Azioni
	app config: GetExtension app config: GetExtensionAssociation app config: GetHostedConfigurationVersion app config: ListApplications app config: ListConfigurationProfiles app config: ListDeploymentStrategies app config: ListDeployments app config: ListEnvironments app config: ListExtensionAssociations app config: ListExtensions app config: ListHostedConfigurationVersions app config: StartDeployment app config: StopDeployment app config: UpdateAccountSettings app config: UpdateApplication app config: UpdateConfigurationProfile app config: UpdateDeploymentStrategy app config: UpdateEnvironment app config: UpdateExtension app config: UpdateExtensionAssociation app config: ValidateConfiguration

Prefisso del servizio	Azioni
appflow	flusso di app: CancelFlowExecutions
	flusso di app: CreateConnectorProfile
	flusso di app: CreateFlow
	flusso di app: DeleteConnectorProfile
	flusso di app: DeleteFlow
	flusso di app: DescribeConnector
	flusso di app: DescribeConnectorEntity
	flusso di app: DescribeConnectorProfiles
	flusso di app: DescribeConnectors
	flusso di app: DescribeFlow
	flusso di app: DescribeFlowExecutionRecords
	flusso di app: ListConnectorEntities
	flusso di app: ListConnectors
	flusso di app: ListFlows
	flusso di app: RegisterConnector
	flusso di app: ResetConnectorMetadataCache
	flusso di app: StartFlow
	flusso di app: StopFlow
	flusso di app: UnRegisterConnector
	flusso di app: UpdateConnectorProfile
	flusso di app: UpdateConnectorRegistration

Prefisso del servizio	Azioni
	flusso di app: UpdateFlow

Prefisso del servizio	Azioni
applicationinsights	approfondimenti sulle applicazioni: AddWorkload approfondimenti sulle applicazioni: CreateApplication approfondimenti sulle applicazioni: CreateComponent approfondimenti sulle applicazioni: CreateLogPattern approfondimenti sulle applicazioni: DeleteApplication approfondimenti sulle applicazioni: DeleteComponent approfondimenti sulle applicazioni: DeleteLogPattern approfondimenti sulle applicazioni: DescribeApplication approfondimenti sulle applicazioni: DescribeComponent approfondimenti sulle applicazioni: DescribeComponentConfiguration approfondimenti sulle applicazioni: DescribeComponentConfigurationRecommendation approfondimenti sulle applicazioni: DescribeLogPattern approfondimenti sulle applicazioni: DescribeObservation approfondimenti sulle applicazioni: DescribeProblem approfondimenti sulle applicazioni: DescribeProblemObservations approfondimenti sulle applicazioni: DescribeWorkload approfondimenti sulle applicazioni: ListApplications approfondimenti sulle applicazioni: ListComponents approfondimenti sulle applicazioni: ListConfigurationHistory approfondimenti sulle applicazioni: ListLogPatternSets

Prefisso del servizio	Azioni
	approfondimenti sulle applicazioni: ListLogPatterns approfondimenti sulle applicazioni: ListProblems approfondimenti sulle applicazioni: ListWorkloads approfondimenti sulle applicazioni: RemoveWorkload approfondimenti sulle applicazioni: UpdateApplication approfondimenti sulle applicazioni: UpdateComponent approfondimenti sulle applicazioni: UpdateComponentConfiguration approfondimenti sulle applicazioni: UpdateLogPattern approfondimenti sulle applicazioni: UpdateWorkload

Prefisso del servizio	Azioni
appmesh	app mesh: CreateGatewayRoute
	app mesh: CreateMesh
	app mesh: CreateRoute
	app mesh: CreateVirtualGateway
	app mesh: CreateVirtualNode
	app mesh: CreateVirtualRouter
	app mesh: CreateVirtualService
	app mesh: DeleteGatewayRoute
	app mesh: DeleteMesh
	app mesh: DeleteRoute
	app mesh: DeleteVirtualGateway
	app mesh: DeleteVirtualNode
	app mesh: DeleteVirtualRouter
	app mesh: DeleteVirtualService
	app mesh: DescribeGatewayRoute
	app mesh: DescribeMesh
	app mesh: DescribeRoute
	app mesh: DescribeVirtualGateway
	app mesh: DescribeVirtualNode
	app mesh: DescribeVirtualRouter
	app mesh: DescribeVirtualService

Prefisso del servizio	Azioni
	app mesh: ListGatewayRoutes
	app mesh: ListMeshes
	app mesh: ListRoutes
	app mesh: ListVirtualGateways
	app mesh: ListVirtualNodes
	app mesh: ListVirtualRouters
	app mesh: ListVirtualServices
	app mesh: StreamAggregatedResources
	app mesh: UpdateGatewayRoute
	app mesh: UpdateMesh
	app mesh: UpdateRoute
	app mesh: UpdateVirtualGateway
	app mesh: UpdateVirtualNode
	app mesh: UpdateVirtualRouter
	app mesh: UpdateVirtualService

Prefisso del servizio	Azioni
appstream	appstream: AssociateAppBlockBuilderAppBlock appstream: AssociateApplicationFleet appstream: AssociateApplicationToEntitlement appstream: AssociateFleet appstream: AssociateSoftwareToImageBuilder appstream: BatchAssociateUserStack appstream: BatchDisassociateUserStack appstream: CopyImage appstream: CreateAppBlock appstream: CreateAppBlockBuilder appstream: CreateAppBlockBuilderStreamingURL appstream: CreateApplication appstream: CreateDirectoryConfig appstream: CreateEntitlement appstream: CreateFleet appstream: CreateImageBuilder appstream: CreateImageBuilderStreamingURL appstream: CreateStack appstream: URL CreateStreaming appstream: CreateThemeForStack appstream: CreateUpdatedImage

Prefisso del servizio	Azioni
	appstream: CreateUsageReportSubscription appstream: CreateUser appstream: DeleteAppBlock appstream: DeleteAppBlockBuilder appstream: DeleteApplication appstream: DeleteDirectoryConfig appstream: DeleteEntitlement appstream: DeleteFleet appstream: DeleteImage appstream: DeleteImageBuilder appstream: DeleteImagePermissions appstream: DeleteStack appstream: DeleteThemeForStack appstream: DeleteUsageReportSubscription appstream: DeleteUser appstream: DescribeAppBlockBuilderAppBlockAssociations appstream: DescribeAppBlockBuilders appstream: DescribeAppBlocks appstream: DescribeAppLicenseUsage appstream: DescribeApplicationFleetAssociations appstream: DescribeApplications

Prefisso del servizio	Azioni
	appstream: DescribeDirectoryConfigs appstream: DescribeEntitlements appstream: DescribeFleets appstream: DescribeImageBuilders appstream: DescribeImagePermissions appstream: DescribeImages appstream: DescribeSessions appstream: DescribeStacks appstream: DescribeThemeForStack appstream: DescribeUsageReportSubscriptions appstream: DescribeUserStackAssociations appstream: DescribeUsers appstream: DisableUser appstream: DisassociateAppBlockBuilderAppBlock appstream: DisassociateApplicationFleet appstream: DisassociateApplicationFromEntitlement appstream: DisassociateFleet appstream: DisassociateSoftwareFromImageBuilder appstream: EnableUser appstream: ExpireSession appstream: GetExportImageTask

Prefisso del servizio	Azioni
	appstream: ListAssociatedFleets appstream: ListAssociatedStacks appstream: ListEntitledApplications appstream: ListExportImageTasks appstream: StartAppBlockBuilder appstream: StartFleet appstream: StartImageBuilder appstream: StartSoftwareDeploymentToImageBuilder appstream: StopAppBlockBuilder appstream: StopFleet appstream: StopImageBuilder appstream: UpdateAppBlockBuilder appstream: UpdateApplication appstream: UpdateDirectoryConfig appstream: UpdateEntitlement appstream: UpdateFleet appstream: UpdateImagePermissions appstream: UpdateStack appstream: UpdateThemeForStack

Prefisso del servizio	Azioni
appsync	sincronizzazione delle app: AssociateApi
	sincronizzazione delle app: AssociateMergedGraphqlApi
	sincronizzazione delle app: AssociateSourceGraphqlApi
	sincronizzazione delle app: CreateApi
	sincronizzazione delle app: CreateApiCache
	sincronizzazione delle app: CreateApiKey
	sincronizzazione delle app: CreateChannelNamespace
	sincronizzazione delle app: CreateDataSource
	sincronizzazione delle app: CreateDomainName
	sincronizzazione delle app: CreateFunction
	sincronizzazione delle app: CreateGraphqlApi
	sincronizzazione delle app: CreateResolver
	sincronizzazione delle app: CreateType
	sincronizzazione delle app: DeleteApi
	sincronizzazione delle app: DeleteApiCache
	sincronizzazione delle app: DeleteApiKey
	sincronizzazione delle app: DeleteChannelNamespace
	sincronizzazione delle app: DeleteDataSource
	sincronizzazione delle app: DeleteDomainName
	sincronizzazione delle app: DeleteFunction
	sincronizzazione delle app: DeleteGraphqlApi

Prefisso del servizio	Azioni
	sincronizzazione delle app: DeleteResolver
	sincronizzazione delle app: DeleteType
	sincronizzazione delle app: DisassociateApi
	sincronizzazione delle app: DisassociateMergedGraphqlApi
	sincronizzazione delle app: DisassociateSourceGraphqlApi
	sincronizzazione delle app: EvaluateCode
	sincronizzazione delle app: EvaluateMappingTemplate
	sincronizzazione delle app: FlushApiCache
	sincronizzazione delle app: GetApi
	sincronizzazione delle app: GetApiAssociation
	sincronizzazione delle app: GetApiCache
	sincronizzazione delle app: GetChannelNamespace
	sincronizzazione delle app: GetDataSource
	sincronizzazione delle app: GetDataSourceIntrospection
	sincronizzazione delle app: GetDomainName
	sincronizzazione delle app: GetFunction
	sincronizzazione delle app: GetGraphqlApi
	sincronizzazione delle app: GetGraphqlApiEnvironmentVariables
	sincronizzazione delle app: GetIntrospectionSchema
	sincronizzazione delle app: GetResolver
	sincronizzazione delle app: GetSchemaCreationStatus

Prefisso del servizio	Azioni
	sincronizzazione delle app: GetSourceApiAssociation
	sincronizzazione delle app: GetType
	sincronizzazione delle app: ListApiKeys
	sincronizzazione delle app: ListApis
	sincronizzazione delle app: ListChannelNamespaces
	sincronizzazione delle app: ListDataSources
	sincronizzazione delle app: ListDomainNames
	sincronizzazione delle app: ListFunctions
	sincronizzazione delle app: ListGraphqlApis
	sincronizzazione delle app: ListResolvers
	sincronizzazione delle app: ListResolversByFunction
	sincronizzazione app: ListSourceApiAssociations
	sincronizzazione app: ListTypes
	sincronizzazione app: ListTypesByAssociation
	sincronizzazione app: PutGraphqlApiEnvironmentVariables
	sincronizzazione app: StartDataSourceIntrospection
	sincronizzazione app: StartSchemaCreation
	sincronizzazione app: StartSchemaMerge
	sincronizzazione app: UpdateApi
	sincronizzazione app: UpdateApiCache
	sincronizzazione app: UpdateApiKey

Prefisso del servizio	Azioni
	sincronizzazione app: UpdateChannelNamespace
	sincronizzazione app: UpdateDataSource
	sincronizzazione app: UpdateDomainName
	sincronizzazione app: UpdateFunction
	sincronizzazione app: UpdateGraphqlApi
	sincronizzazione app: UpdateResolver
	sincronizzazione app: UpdateSourceApiAssociation
	sincronizzazione app: UpdateType

Prefisso del servizio	Azioni
aps	app: CreateAlertManagerDefinition rubinetti: CreateAnomalyDetector rubinetti: CreateLoggingConfiguration rubinetti: CreateQueryLoggingConfiguration rubinetti: CreateRuleGroupsNamespace rubinetti: CreateWorkspace rubinetti: DeleteAlertManagerDefinition rubinetti: DeleteAnomalyDetector rubinetti: DeleteLoggingConfiguration rubinetti: DeleteQueryLoggingConfiguration rubinetti: DeleteResourcePolicy rubinetti: DeleteRuleGroupsNamespace rubinetti: DeleteScraper rubinetti: DeleteScraperLoggingConfiguration rubinetti: DeleteWorkspace rubinetti: DescribeAlertManagerDefinition rubinetti: DescribeAnomalyDetector rubinetti: DescribeLoggingConfiguration rubinetti: DescribeQueryLoggingConfiguration rubinetti: DescribeResourcePolicy rubinetti: DescribeRuleGroupsNamespace

Prefisso del servizio	Azioni
	rubinetti: DescribeScraper rubinetti: DescribeScraperLoggingConfiguration rubinetti: DescribeWorkspace rubinetti: DescribeWorkspaceConfiguration rubinetti: GetDefaultScraperConfiguration rubinetti: ListAnomalyDetectors rubinetti: ListRuleGroupsNamespaces rubinetti: ListScrapers rubinetti: ListWorkspaces rubinetti: PutAlertManagerDefinition rubinetti: PutAnomalyDetector rubinetti: PutResourcePolicy rubinetti: PutRuleGroupsNamespace rubinetti: UpdateLoggingConfiguration rubinetti: UpdateQueryLoggingConfiguration rubinetti: UpdateScraper rubinetti: UpdateScraperLoggingConfiguration rubinetti: UpdateWorkspaceAlias rubinetti: UpdateWorkspaceConfiguration

Prefisso del servizio	Azioni
atena	atena: BatchGetNamedQuery
	atena: BatchGetPreparedStatement
	atena: BatchGetQueryExecution
	atena: CancelCapacityReservation
	atena: CreateCapacityReservation
	atena: CreateDataCatalog
	atena: CreateNamedQuery
	atena: CreateNotebook
	atena: CreatePreparedStatement
	atena: CreatePresignedNotebookUrl
	atena: CreateWorkGroup
	atena: DeleteCapacityReservation
	atena: DeleteDataCatalog
	atena: DeleteNamedQuery
	atena: DeleteNotebook
	atena: DeletePreparedStatement
	atena: DeleteWorkGroup
	atena: ExportNotebook
	atena: GetCalculationExecution
	atena: GetCalculationExecutionCode
	atena: GetCalculationExecutionStatus

Prefisso del servizio	Azioni
	atena: GetCapacityAssignmentConfiguration
	atena: GetCapacityReservation
	atena: GetDataCatalog
	atena: GetDatabase
	atena: GetNamedQuery
	atena: GetNotebookMetadata
	atena: GetPreparedStatement
	atena: GetQueryExecution
	atena: GetQueryResults
	atena: GetQueryResultsStream
	atena: GetQueryRuntimeStatistics
	atena: GetResourceDashboard
	atena: GetSession
	atena: GetSessionEndpoint
	atena: GetSessionStatus
	atena: GetTableMetadata
	atena: GetWorkGroup
	atena: ImportNotebook
	atena: ListApplicationDPUSizes
	atena: ListCalculationExecutions
	atena: ListCapacityReservations

Prefisso del servizio	Azioni
	atena: ListDataCatalogs
	atena: ListDatabases
	atena: ListEngineVersions
	atena: ListExecutors
	atena: ListNamedQueries
	atena: ListNotebookMetadata
	atena: ListNotebookSessions
	atena: ListPreparedStatements
	atena: ListQueryExecutions
	atena: ListSessions
	atena: ListTableMetadata
	atena: ListWorkGroups
	atena: PutCapacityAssignmentConfiguration
	atena: StartCalculationExecution
	atena: StartQueryExecution
	atena: StartSession
	atena: StopCalculationExecution
	atena: StopQueryExecution
	atena: TerminateSession
	atena: UpdateCapacityReservation
	atena: UpdateDataCatalog

Prefisso del servizio	Azioni
	atena: UpdateNamedQuery atena: UpdateNotebook atena: UpdateNotebookMetadata atena: UpdatePreparedStatement atena: UpdateWorkGroup

Prefisso del servizio	Azioni
auditmanager	responsabile dell'audit: AssociateAssessmentReportEvidenceFolder responsabile dell'audit: BatchAssociateAssessmentReportEvidence responsabile dell'audit: BatchCreateDelegationByAssessment responsabile dell'audit: BatchDeleteDelegationByAssessment responsabile dell'audit: BatchDisassociateAssessmentReportEvidence responsabile dell'audit: BatchImportEvidenceToAssessmentControl responsabile dell'audit: CreateAssessment responsabile dell'audit: CreateAssessmentFramework responsabile dell'audit: CreateAssessmentReport responsabile dell'audit: CreateControl responsabile dell'audit: DeleteAssessment responsabile dell'audit: DeleteAssessmentFramework responsabile dell'audit: DeleteAssessmentFrameworkShare responsabile dell'audit: DeleteAssessmentReport responsabile dell'audit: DeleteControl responsabile dell'audit: DeregisterAccount responsabile dell'audit: DeregisterOrganizationAdminAccount responsabile dell'audit: DisassociateAssessmentReportEvidenceFolder responsabile dell'audit: GetAccountStatus responsabile dell'audit: GetAssessment

Prefisso del servizio	Azioni
	responsabile dell'audit: GetAssessmentFramework responsabile dell'audit: GetAssessmentReportUrl responsabile dell'audit: GetChangeLogs responsabile dell'audit: GetControl responsabile dell'audit: GetDelegations responsabile dell'audit: GetEvidence responsabile dell'audit: GetEvidenceByEvidenceFolder responsabile dell'audit: GetEvidenceFileUploadUrl responsabile dell'audit: GetEvidenceFolder responsabile dell'audit: GetEvidenceFoldersByAssessment responsabile dell'audit: GetEvidenceFoldersByAssessmentControl responsabile dell'audit: GetInsights responsabile dell'audit: GetInsightsByAssessment responsabile dell'audit: GetOrganizationAdminAccount responsabile dell'audit: GetServicesInScope responsabile dell'audit: GetSettings responsabile dell'audit: ListAssessmentControlInsightsByControlDomain responsabile dell'audit: ListAssessmentFrameworkShareRequests responsabile dell'audit: ListAssessmentFrameworks responsabile dell'audit: ListAssessmentReports

Prefisso del servizio	Azioni
	responsabile dell'audit: ListAssessments responsabile dell'audit: ListControlDomainInsights responsabile dell'audit: ListControlDomainInsightsByAssessment responsabile dell'audit: ListControlInsightsByControlDomain responsabile dell'audit: ListControls responsabile dell'audit: ListKeywordsForDataSource responsabile dell'audit: ListNotifications responsabile dell'audit: RegisterAccount responsabile dell'audit: RegisterOrganizationAdminAccount responsabile dell'audit: StartAssessmentFrameworkShare responsabile dell'audit: UpdateAssessment responsabile dell'audit: UpdateAssessmentControl responsabile dell'audit: UpdateAssessmentControlSetStatus responsabile dell'audit: UpdateAssessmentFramework responsabile dell'audit: UpdateAssessmentFrameworkShare responsabile dell'audit: UpdateAssessmentStatus responsabile dell'audit: UpdateControl responsabile dell'audit: UpdateSettings responsabile dell'audit: ValidateAssessmentReportIntegrity

Prefisso del servizio	Azioni
scalabilità automatica	scalabilità automatica: AttachInstances
	scalabilità automatica: AttachLoadBalancerTargetGroups
	scalabilità automatica: AttachLoadBalancers
	scalabilità automatica: AttachTrafficSources
	scalabilità automatica: BatchDeleteScheduledAction
	scalabilità automatica: BatchPutScheduledUpdateGroupAction
	scalabilità automatica: CancelInstanceRefresh
	scalabilità automatica: CompleteLifecycleAction
	scalabilità automatica: CreateAutoScalingGroup
	scalabilità automatica: CreateLaunchConfiguration
	scalabilità automatica: DeleteAutoScalingGroup
	scalabilità automatica: DeleteLaunchConfiguration
	scalabilità automatica: DeleteLifecycleHook
	scalabilità automatica: DeleteNotificationConfiguration
	scalabilità automatica: DeletePolicy
	scalabilità automatica: DeleteScheduledAction
	scalabilità automatica: DeleteWarmPool
	scalabilità automatica: DescribeAccountLimits
	scalabilità automatica: DescribeAdjustmentTypes
	scalabilità automatica: DescribeAutoScalingGroups
	scalabilità automatica: DescribeAutoScalingInstances

Prefisso del servizio	Azioni
	scalabilità automatica: DescribeAutoScalingNotificationTypes
	scalabilità automatica: DescribeInstanceRefreshes
	scalabilità automatica: DescribeLaunchConfigurations
	scalabilità automatica: DescribeLifecycleHookTypes
	scalabilità automatica: DescribeLifecycleHooks
	scalabilità automatica: DescribeLoadBalancerTargetGroups
	scalabilità automatica: DescribeLoadBalancers
	scalabilità automatica: DescribeMetricCollectionTypes
	scalabilità automatica: DescribeNotificationConfigurations
	scalabilità automatica: DescribePolicies
	scalabilità automatica: DescribeScalingActivities
	scalabilità automatica: DescribeScalingProcessTypes
	scalabilità automatica: DescribeScheduledActions
	scalabilità automatica: DescribeTerminationPolicyTypes
	scalabilità automatica: DescribeTrafficSources
	scalabilità automatica: DescribeWarmPool
	scalabilità automatica: DetachInstances
	scalabilità automatica: DetachLoadBalancerTargetGroups
	scalabilità automatica: DetachLoadBalancers
	scalabilità automatica: DetachTrafficSources
	scalabilità automatica: DisableMetricsCollection

Prefisso del servizio	Azioni
	scalabilità automatica: EnableMetricsCollection
	scalabilità automatica: EnterStandby
	scalabilità automatica: ExecutePolicy
	scalabilità automatica: ExitStandby
	scalabilità automatica: GetPredictiveScalingForecast
	scalabilità automatica: PutLifecycleHook
	scalabilità automatica: PutNotificationConfiguration
	scalabilità automatica: PutScalingPolicy
	scalabilità automatica: PutScheduledUpdateGroupAction
	scalabilità automatica: PutWarmPool
	scalabilità automatica: RecordLifecycleActionHeartbeat
	scalabilità automatica: ResumeProcesses
	scalabilità automatica: RollbackInstanceRefresh
	scalabilità automatica: SetDesiredCapacity
	scalabilità automatica: SetInstanceHealth
	scalabilità automatica: SetInstanceProtection
	scalabilità automatica: StartInstanceRefresh
	scalabilità automatica: SuspendProcesses
	scalabilità automatica: TerminateInstanceInAutoScalingGroup
	scalabilità automatica: UpdateAutoScalingGroup
aws-marketplace	aws-marketplace: GetEntitlements

Prefisso del servizio	Azioni
backup	backup: CancelLegalHold
	backup: CreateBackupPlan
	backup: CreateBackupSelection
	backup: CreateBackupVault
	backup: CreateFramework
	backup: CreateLegalHold
	backup: CreateLogicallyAirGappedBackupVault
	backup: CreateReportPlan
	backup: CreateRestoreAccessBackupVault
	backup: CreateRestoreTestingPlan
	backup: CreateRestoreTestingSelection
	backup: CreateTieringConfiguration
	backup: DeleteBackupPlan
	backup: DeleteBackupSelection
	backup: DeleteBackupVault
	backup: DeleteBackupVaultAccessPolicy
	backup: DeleteBackupVaultLockConfiguration
	backup: DeleteBackupVaultNotifications
	backup: DeleteFramework
	backup: DeleteRecoveryPoint
	backup: DeleteReportPlan

Prefisso del servizio	Azioni
	backup: DeleteRestoreTestingPlan
	backup: DeleteRestoreTestingSelection
	backup: DeleteTieringConfiguration
	backup: DescribeBackupJob
	backup: DescribeBackupVault
	backup: DescribeCopyJob
	backup: DescribeFramework
	backup: DescribeGlobalSettings
	backup: DescribeProtectedResource
	backup: DescribeRecoveryPoint
	backup: DescribeRegionSettings
	backup: DescribeReportJob
	backup: DescribeReportPlan
	backup: DescribeRestoreJob
	backup: DescribeScanJob
	backup: DisassociateRecoveryPoint
	backup: DisassociateRecoveryPointFromParent
	backup: ExportBackupPlanTemplate
	backup: GetBackupPlan
	backup: GetBackupPlanFromJSON
	backup: GetBackupPlanFromTemplate

Prefisso del servizio	Azioni
	backup: GetBackupSelection
	backup: GetBackupVaultAccessPolicy
	backup: GetBackupVaultNotifications
	backup: GetLegalHold
	backup: GetRecoveryPointRestoreMetadata
	backup: GetRestoreJobMetadata
	backup: GetRestoreTestingInferredMetadata
	backup: GetRestoreTestingPlan
	backup: GetRestoreTestingSelection
	backup: GetSupportedResourceTypes
	backup: GetTieringConfiguration
	backup: ListBackupJobSummaries
	backup: ListBackupJobs
	backup: ListBackupPlanTemplates
	backup: ListBackupPlanVersions
	backup: ListBackupPlans
	backup: ListBackupSelections
	backup: ListBackupVaults
	backup: ListCopyJobSummaries
	backup: ListCopyJobs
	backup: ListFrameworks

Prefisso del servizio	Azioni
	backup: ListIndexedRecoveryPoints
	backup: ListLegalHolds
	backup: ListProtectedResources
	backup: ListRecoveryPointsByBackupVault
	backup: ListRecoveryPointsByLegalHold
	backup: ListRecoveryPointsByResource
	backup: ListReportJobs
	backup: ListReportPlans
	backup: ListRestoreAccessBackupVaults
	backup: ListRestoreJobSummaries
	backup: ListRestoreJobs
	backup: ListRestoreJobsByProtectedResource
	backup: ListRestoreTestingPlans
	backup: ListRestoreTestingSelections
	backup: ListScanJobSummaries
	backup: ListScanJobs
	backup: ListTieringConfigurations
	backup: PutBackupVaultAccessPolicy
	backup: PutBackupVaultLockConfiguration
	backup: PutBackupVaultNotifications
	backup: PutRestoreValidationResult

Prefisso del servizio	Azioni
	backup: StartBackupJob
	backup: StartCopyJob
	backup: StartReportJob
	backup: StartRestoreJob
	backup: StopBackupJob
	backup: UpdateBackupPlan
	backup: UpdateFramework
	backup: UpdateGlobalSettings
	backup: UpdateRecoveryPointLifecycle
	backup: UpdateRegionSettings
	backup: UpdateReportPlan
	backup: UpdateRestoreTestingPlan
	backup: UpdateRestoreTestingSelection
	backup: UpdateTieringConfiguration

Prefisso del servizio	Azioni
batch	lotto: CancelJob lotto: CreateComputeEnvironment lotto: CreateConsumableResource lotto: CreateJobQueue lotto: CreateQuotaShare lotto: CreateSchedulingPolicy lotto: CreateServiceEnvironment lotto: DeleteComputeEnvironment lotto: DeleteConsumableResource lotto: DeleteJobQueue lotto: DeleteQuotaShare lotto: DeleteSchedulingPolicy lotto: DeleteServiceEnvironment lotto: DeregisterJobDefinition lotto: DescribeComputeEnvironments lotto: DescribeConsumableResource lotto: DescribeJobDefinitions lotto: DescribeJobQueues lotto: DescribeJobs lotto: DescribeQuotaShare lotto: DescribeSchedulingPolicies

Prefisso del servizio	Azioni
	lotto: DescribeServiceEnvironments lotto: DescribeServiceJob lotto: GetJobQueueSnapshot lotto: ListConsumableResources lotto: ListJobs lotto: ListJobsByConsumableResource lotto: ListQuotaShares lotto: ListSchedulingPolicies lotto: ListServiceJobs lotto: RegisterJobDefinition lotto: SubmitJob lotto: SubmitServiceJob lotto: TerminateJob lotto: TerminateServiceJob lotto: UpdateComputeEnvironment lotto: UpdateConsumableResource lotto: UpdateJobQueue lotto: UpdateQuotaShare lotto: UpdateSchedulingPolicy lotto: UpdateServiceEnvironment lotto: UpdateServiceJob

Prefisso del servizio	Azioni
braket	staffa: CancelJob staffa: CancelQuantumTask staffa: CreateJob staffa: CreateQuantumTask staffa: CreateSpendingLimit staffa: GetDevice staffa: GetJob staffa: GetQuantumTask staffa: SearchDevices staffa: SearchJobs staffa: SearchQuantumTasks staffa: SearchSpendingLimits

Prefisso del servizio	Azioni
budgets	budget: CreateBudgetAction bilanci: DeleteBudgetAction bilanci: DescribeBudgetAction bilanci: DescribeBudgetActionHistories bilanci: DescribeBudgetActionsForAccount bilanci: DescribeBudgetActionsForBudget bilanci: ExecuteBudgetAction bilanci: ModifyBudget bilanci: UpdateBudgetAction bilanci: ViewBudget
cloud9	cloud9: EC2 CreateEnvironment nuvola 9: CreateEnvironmentMembership nuvola 9: DeleteEnvironment nuvola 9: DeleteEnvironmentMembership nuvola 9: DescribeEnvironmentMemberships nuvola 9: DescribeEnvironmentStatus nuvola 9: DescribeEnvironments nuvola 9: ListEnvironments nuvola 9: UpdateEnvironment nuvola 9: UpdateEnvironmentMembership

Prefisso del servizio	Azioni
cloudformation	formazione di nuvole: BatchDescribeTypeConfigurations formazione di nuvole: CancelUpdateStack formazione di nuvole: ContinueUpdateRollback formazione di nuvole: CreateChangeSet formazione di nuvole: CreateGeneratedTemplate formazione di nuvole: CreateStack formazione di nuvole: CreateStackInstances formazione di nuvole: CreateStackSet formazione di nuvole: DeactivateType formazione di nuvole: DeleteChangeSet formazione di nuvole: DeleteGeneratedTemplate formazione di nuvole: DeleteStack formazione di nuvole: DeleteStackInstances formazione di nuvole: DeleteStackSet formazione di nuvole: DeregisterType formazione di nuvole: DescribeAccountLimits formazione di nuvole: DescribeChangeSet formazione di nuvole: DescribeChangeSetHooks formazione di nuvole: DescribeEvents formazione di nuvole: DescribeGeneratedTemplate formazione di nuvole: DescribeOrganizationsAccess

Prefisso del servizio	Azioni
	formazione di nuvole: DescribePublisher formazione di nuvole: DescribeResourceScan formazione di nuvole: DescribeStackDriftDetectionStatus formazione di nuvole: DescribeStackEvents formazione di nuvole: DescribeStackInstance formazione di nuvole: DescribeStackResource formazione di nuvole: DescribeStackResourceDrifts formazione di nuvole: DescribeStackResources formazione di nuvole: DescribeStackSet formazione di nuvole: DescribeStackSetOperation formazione di nuvole: DescribeStacks formazione di nuvole: DescribeType formazione di nuvole: DescribeTypeRegistration formazione di nuvole: DetectStackDrift formazione di nuvole: DetectStackResourceDrift formazione di nuvole: DetectStackSetDrift formazione di nuvole: EstimateTemplateCost formazione di nuvole: ExecuteChangeSet formazione di nuvole: GetGeneratedTemplate formazione di nuvole: GetHookResult formazione di nuvole: GetStackPolicy

Prefisso del servizio	Azioni
	formazione di nuvole: GetTemplate formazione di nuvole: GetTemplateSummary formazione di nuvole: ImportStacksToStackSet formazione di nuvole: ListChangeSets formazione di nuvole: ListExports formazione di nuvole: ListGeneratedTemplates formazione di nuvole: ListHookResults formazione di nuvole: ListImports formazione di nuvole: ListResourceScanRelatedResources formazione di nuvole: ListResourceScanResources formazione di nuvole: ListResourceScans formazione di nuvole: ListStackInstanceResourceDrifts formazione di nuvole: ListStackInstances formazione di nuvole: ListStackRefactors formazione di nuvole: ListStackResources formazione di nuvole: ListStackSetAutoDeploymentTargets formazione di nuvole: ListStackSetOperationResults formazione di nuvole: ListStackSetOperations formazione di nuvole: ListStackSets formazione di nuvole: ListTypeRegistrations formazione di nuvole: ListTypeVersions

Prefisso del servizio	Azioni
	formazione di nuvole: ListTypes formazione di nuvole: PublishType formazione di nuvole: RecordHandlerProgress formazione di nuvole: RegisterPublisher formazione di nuvole: RegisterType formazione di nuvole: RollbackStack formazione di nuvole: SetStackPolicy formazione di nuvole: SetTypeConfiguration formazione di nuvole: SetTypeDefaultVersion formazione di nuvole: SignalResource formazione di nuvole: StartResourceScan formazione di nuvole: StopStackSetOperation formazione di nuvole: TestType formazione di nuvole: UpdateGeneratedTemplate formazione di nuvole: UpdateStack formazione di nuvole: UpdateStackInstances formazione di nuvole: UpdateStackSet formazione di nuvole: UpdateTerminationProtection formazione di nuvole: ValidateTemplate

Prefisso del servizio	Azioni
cloudfront	fronte cloud: AssociateAlias
	fronte cloud: AssociateDistributionTenantWebACL
	fronte cloud: AssociateDistributionWebACL
	fronte cloud: CreateCachePolicy
	fronte cloud: CreateCloudFrontOriginAccessIdentity
	fronte cloud: CreateConnectionFunction
	fronte cloud: CreateContinuousDeploymentPolicy
	fronte cloud: CreateDistributionTenant
	fronte cloud: CreateFieldLevelEncryptionConfig
	fronte cloud: CreateFieldLevelEncryptionProfile
	fronte cloud: CreateFunction
	fronte cloud: CreateInvalidation
	fronte cloud: CreateKeyGroup
	fronte cloud: CreateKeyValueStore
	fronte cloud: CreateMonitoringSubscription
	fronte cloud: CreateOriginAccessControl
	fronte cloud: CreateOriginRequestPolicy
	fronte cloud: CreatePublicKey
	fronte cloud: CreateRealtimeLogConfig
	fronte cloud: CreateResponseHeadersPolicy
	fronte cloud: CreateTrustStore

Prefisso del servizio	Azioni
	fronte cloud: DeleteAnycastIpList
	fronte cloud: DeleteCachePolicy
	fronte cloud: DeleteCloudFrontOriginAccessIdentity
	fronte cloud: DeleteConnectionFunction
	fronte cloud: DeleteConnectionGroup
	fronte cloud: DeleteContinuousDeploymentPolicy
	fronte cloud: DeleteDistribution
	fronte cloud: DeleteDistributionTenant
	fronte cloud: DeleteFieldLevelEncryptionConfig
	fronte cloud: DeleteFieldLevelEncryptionProfile
	fronte cloud: DeleteFunction
	fronte cloud: DeleteKeyGroup
	fronte cloud: DeleteKeyValueStore
	fronte cloud: DeleteMonitoringSubscription
	fronte cloud: DeleteOriginAccessControl
	fronte cloud: DeleteOriginRequestPolicy
	fronte cloud: DeletePublicKey
	fronte cloud: DeleteRealtimeLogConfig
	fronte cloud: DeleteResponseHeadersPolicy
	fronte cloud: DeleteStreamingDistribution
	fronte cloud: DeleteTrustStore

Prefisso del servizio	Azioni
	fronte cloud: DeleteVpcOrigin
	fronte cloud: DescribeFunction
	fronte cloud: DescribeKeyValueStore
	fronte cloud: DisassociateDistributionTenantWebACL
	fronte cloud: DisassociateDistributionWebACL
	fronte cloud: GetAnycastIpList
	fronte cloud: GetCachePolicy
	fronte cloud: GetCachePolicyConfig
	fronte cloud: GetCloudFrontOriginAccessIdentity
	fronte cloud: GetCloudFrontOriginAccessIdentityConfig
	fronte cloud: GetContinuousDeploymentPolicy
	fronte cloud: GetContinuousDeploymentPolicyConfig
	fronte cloud: GetDistributionConfig
	fronte cloud: GetFieldLevelEncryption
	fronte cloud: GetFieldLevelEncryptionConfig
	fronte cloud: GetFieldLevelEncryptionProfile
	fronte cloud: GetFieldLevelEncryptionProfileConfig
	fronte cloud: GetFunction
	fronte cloud: GetInvalidation
	fronte cloud: GetInvalidationForDistributionTenant
	fronte cloud: GetKeyGroup

Prefisso del servizio	Azioni
	fronte cloud: GetKeyGroupConfig
	fronte cloud: GetMonitoringSubscription
	fronte cloud: GetOriginAccessControl
	fronte cloud: GetOriginAccessControlConfig
	fronte cloud: GetOriginRequestPolicy
	fronte cloud: GetOriginRequestPolicyConfig
	fronte cloud: GetPublicKey
	fronte cloud: GetPublicKeyConfig
	fronte cloud: GetRealtimeLogConfig
	fronte cloud: GetResponseHeadersPolicy
	fronte cloud: GetResponseHeadersPolicyConfig
	fronte cloud: GetStreamingDistribution
	fronte cloud: GetStreamingDistributionConfig
	fronte cloud: GetVpcOrigin
	fronte cloud: ListAnycastIpLists
	fronte cloud: ListCachePolicies
	fronte cloud: ListCloudFrontOriginAccessIdentities
	fronte cloud: ListConflictingAliases
	fronte cloud: ListConnectionFunctions
	fronte cloud: ListConnectionGroups
	fronte cloud: ListContinuousDeploymentPolicies

Prefisso del servizio	Azioni
	fronte cloud: ListDistributionTenants
	fronte cloud: ListDistributionTenantsByCustomization
	fronte cloud: ListDistributions
	fronte cloud: ListDistributionsByAnycastIpListId
	fronte cloud: ListDistributionsByCachePolicyId
	fronte cloud: ListDistributionsByConnectionMode
	fronte cloud: ListDistributionsByKeyGroup
	fronte cloud: ListDistributionsByOriginRequestPolicyId
	fronte cloud: ListDistributionsByRealtimeLogConfig
	fronte cloud: ListDistributionsByResponseHeadersPolicyId
	fronte cloud: ListDistributionsByVpcOriginId
	fronte cloud: ListDistributionsByWebACLId
	fronte cloud: ListFieldLevelEncryptionConfigs
	fronte cloud: ListFieldLevelEncryptionProfiles
	fronte cloud: ListFunctions
	fronte cloud: ListInvalidations
	fronte cloud: ListInvalidationsForDistributionTenant
	fronte cloud: ListKeyGroups
	fronte cloud: ListKeyValueStores
	fronte cloud: ListOriginAccessControls
	fronte cloud: ListOriginRequestPolicies

Prefisso del servizio	Azioni
	fronte cloud: ListPublicKeys
	fronte cloud: ListRealtimeLogConfigs
	fronte cloud: ListResponseHeadersPolicies
	fronte cloud: ListStreamingDistributions
	fronte cloud: ListTrustStores
	fronte cloud: PublishConnectionFunction
	fronte cloud: PublishFunction
	fronte cloud: TestConnectionFunction
	fronte cloud: TestFunction
	fronte cloud: UpdateAnycastIpList
	fronte cloud: UpdateCachePolicy
	fronte cloud: UpdateCloudFrontOriginAccessIdentity
	fronte cloud: UpdateConnectionFunction
	fronte cloud: UpdateConnectionGroup
	fronte cloud: UpdateContinuousDeploymentPolicy
	fronte cloud: UpdateDistribution
	fronte cloud: UpdateDistributionTenant
	fronte cloud: UpdateFieldLevelEncryptionConfig
	fronte cloud: UpdateFieldLevelEncryptionProfile
	fronte cloud: UpdateFunction
	fronte cloud: UpdateKeyGroup

Prefisso del servizio	Azioni
	fronte cloud: UpdateKeyValueStore fronte cloud: UpdateOriginAccessControl fronte cloud: UpdateOriginRequestPolicy fronte cloud: UpdatePublicKey fronte cloud: UpdateRealtimeLogConfig fronte cloud: UpdateResponseHeadersPolicy fronte cloud: UpdateTrustStore
cloudhsm	cloudhsm: CreateHsm cloudhsm: DeleteBackup cloudhsm: DeleteHsm cloudhsm: DeleteResourcePolicy cloudhsm: DescribeBackups cloudhsm: DescribeClusters cloudhsm: GetResourcePolicy cloudhsm: InitializeCluster cloudhsm: ModifyBackupAttributes cloudhsm: ModifyCluster cloudhsm: PutResourcePolicy cloudhsm: RestoreBackup

Prefisso del servizio	Azioni
cloudsearch	ricerca nel cloud: BuildSuggesters
	ricerca nel cloud: CreateDomain
	ricerca nel cloud: DefineAnalysisScheme
	ricerca nel cloud: DefineExpression
	ricerca nel cloud: DefineIndexField
	ricerca nel cloud: DefineSuggester
	ricerca nel cloud: DeleteAnalysisScheme
	ricerca nel cloud: DeleteDomain
	ricerca nel cloud: DeleteExpression
	ricerca nel cloud: DeleteIndexField
	ricerca nel cloud: DeleteSuggester
	ricerca nel cloud: DescribeAnalysisSchemes
	ricerca nel cloud: DescribeAvailabilityOptions
	ricerca nel cloud: DescribeDomainEndpointOptions
	ricerca nel cloud: DescribeDomains
	ricerca nel cloud: DescribeExpressions
	ricerca nel cloud: DescribeIndexFields
	ricerca nel cloud: DescribeScalingParameters
	ricerca nel cloud: DescribeServiceAccessPolicies
	ricerca nel cloud: DescribeSuggesters
	ricerca nel cloud: IndexDocuments

Prefisso del servizio	Azioni
	ricerca nel cloud: ListDomainNames
	ricerca nel cloud: UpdateAvailabilityOptions
	ricerca nel cloud: UpdateDomainEndpointOptions
	ricerca nel cloud: UpdateScalingParameters
	ricerca nel cloud: UpdateServiceAccessPolicies

Prefisso del servizio	Azioni
cloudtrail	pista nuvolosa: CancelQuery pista nuvolosa: CreateChannel pista nuvolosa: CreateDashboard pista nuvolosa: CreateEventDataStore pista nuvolosa: CreateTrail pista nuvolosa: DeleteChannel pista nuvolosa: DeleteDashboard pista nuvolosa: DeleteEventDataStore pista nuvolosa: DeleteResourcePolicy pista nuvolosa: DeleteTrail pista nuvolosa: DeregisterOrganizationDelegatedAdmin pista nuvolosa: DescribeQuery pista nuvolosa: DescribeTrails pista nuvolosa: DisableFederation pista nuvolosa: GenerateQuery pista nuvolosa: GetChannel pista nuvolosa: GetDashboard pista nuvolosa: GetEventConfiguration pista nuvolosa: GetEventDataStore pista nuvolosa: GetEventDataStoreData pista nuvolosa: GetEventSelectors

Prefisso del servizio	Azioni
	pista nuvolosa: GetImport pista nuvolosa: GetInsightSelectors pista nuvolosa: GetResourcePolicy pista nuvolosa: GetTrail pista nuvolosa: GetTrailStatus pista nuvolosa: ListChannels pista nuvolosa: ListDashboards pista nuvolosa: ListEventDataStores pista nuvolosa: ListImportFailures pista nuvolosa: ListImports pista nuvolosa: ListInsightsData pista nuvolosa: ListPublicKeys pista nuvolosa: ListQueries pista nuvolosa: ListTrails pista nuvolosa: LookupEvents pista nuvolosa: PutEventConfiguration pista nuvolosa: PutEventSelectors pista nuvolosa: PutInsightSelectors pista nuvolosa: PutResourcePolicy pista nuvolosa: RegisterOrganizationDelegatedAdmin pista nuvolosa: RestoreEventDataStore

Prefisso del servizio	Azioni
	pista nuvolosa: SearchSampleQueries pista nuvolosa: StartEventDataStoreIngestion pista nuvolosa: StartImport pista nuvolosa: StartLogging pista nuvolosa: StartQuery pista nuvolosa: StopEventDataStoreIngestion pista nuvolosa: StopImport pista nuvolosa: StopLogging pista nuvolosa: UpdateChannel pista nuvolosa: UpdateDashboard pista nuvolosa: UpdateEventDataStore pista nuvolosa: UpdateTrail

Prefisso del servizio	Azioni
cloudwatch	orologio cloud: DeleteAlarmMuteRule
	orologio nuvoloso: DeleteAlarms
	orologio nuvoloso: DeleteAnomalyDetector
	orologio nuvoloso: DeleteDashboards
	orologio nuvoloso: DeleteInsightRules
	orologio nuvoloso: DeleteMetricStream
	orologio nuvoloso: DescribeAlarmHistory
	orologio nuvoloso: DescribeAlarms
	orologio nuvoloso: DescribeAlarmsForMetric
	orologio nuvoloso: DescribeAnomalyDetectors
	orologio nuvoloso: DescribeInsightRules
	orologio nuvoloso: DisableAlarmActions
	orologio nuvoloso: DisableInsightRules
	orologio nuvoloso: EnableAlarmActions
	orologio nuvoloso: EnableInsightRules
	orologio nuvoloso: GetAlarmMuteRule
	orologio nuvoloso: GetDashboard
	orologio nuvoloso: GetDataset
	orologio nuvoloso: GetInsightRuleReport
	orologio nuvoloso: GetMetricStatistics
	orologio nuvoloso: GetMetricStream

Prefisso del servizio	Azioni
	orologio nuvoloso: ListAlarmMuteRules
	orologio nuvoloso: ListDashboards
	orologio nuvoloso: ListManagedInsightRules
	orologio nuvoloso: ListMetricStreams
	orologio nuvoloso: PutAlarmMuteRule
	orologio nuvoloso: PutAnomalyDetector
	orologio nuvoloso: PutCompositeAlarm
	orologio nuvoloso: PutDashboard
	orologio nuvoloso: PutInsightRule
	orologio nuvoloso: PutLogAlarm
	orologio nuvoloso: PutManagedInsightRules
	orologio nuvoloso: PutMetricAlarm
	orologio nuvoloso: PutMetricStream
	orologio nuvoloso: SetAlarmState
	orologio nuvoloso: StartMetricStreams
	orologio nuvoloso: StopMetricStreams

Prefisso del servizio	Azioni
codeartifact	artefatto del codice: AssociateExternalConnection artefatto del codice: CopyPackageVersions artefatto del codice: CreateDomain artefatto del codice: CreateRepository artefatto del codice: DeleteDomain artefatto del codice: DeleteDomainPermissionsPolicy artefatto del codice: DeletePackage artefatto del codice: DeletePackageVersions artefatto del codice: DeleteRepository artefatto del codice: DeleteRepositoryPermissionsPolicy artefatto del codice: DescribeDomain artefatto del codice: DescribePackage artefatto del codice: DescribePackageVersion artefatto del codice: DescribeRepository artefatto del codice: DisassociateExternalConnection artefatto del codice: DisposePackageVersions artefatto del codice: GetAssociatedPackageGroup artefatto del codice: GetAuthorizationToken artefatto del codice: GetDomainPermissionsPolicy artefatto del codice: GetPackageVersionAsset artefatto del codice: GetPackageVersionReadme

Prefisso del servizio	Azioni
	artefatto del codice: GetRepositoryEndpoint artefatto del codice: GetRepositoryPermissionsPolicy artefatto del codice: ListDomains artefatto del codice: ListPackageGroups artefatto del codice: ListPackageVersionAssets artefatto del codice: ListPackageVersionDependencies artefatto del codice: ListPackageVersions artefatto del codice: ListPackages artefatto del codice: ListRepositories artefatto del codice: ListRepositoriesInDomain artefatto del codice: PublishPackageVersion artefatto del codice: PutDomainPermissionsPolicy artefatto del codice: PutPackageMetadata artefatto del codice: PutPackageOriginConfiguration artefatto del codice: PutRepositoryPermissionsPolicy artefatto del codice: ReadFromRepository artefatto del codice: UpdatePackageVersionsStatus artefatto del codice: UpdateRepository

Prefisso del servizio	Azioni
codedeploy	distribuzione del codice: BatchGetApplicationRevisions
	codedeploy: BatchGetApplications
	codedeploy: BatchGetDeploymentGroups
	codedeploy: BatchGetDeploymentInstances
	codedeploy: BatchGetDeploymentTargets
	codedeploy: BatchGetDeployments
	codedeploy: BatchGetOnPremisesInstances
	codedeploy: ContinueDeployment
	codedeploy: CreateApplication
	codedeploy: CreateDeployment
	codedeploy: CreateDeploymentConfig
	codedeploy: CreateDeploymentGroup
	codedeploy: DeleteApplication
	codedeploy: DeleteDeploymentConfig
	codedeploy: DeleteDeploymentGroup
	codedeploy: DeleteGitHubAccountToken
	codedeploy: DeleteResourcesByExternalId
	codedeploy: DeregisterOnPremisesInstance
	codedeploy: GetApplication
	codedeploy: GetApplicationRevision
	codedeploy: GetDeployment

Prefisso del servizio	Azioni
	codedeploy: GetDeploymentConfig
	codedeploy: GetDeploymentGroup
	codedeploy: GetDeploymentInstance
	codedeploy: GetDeploymentTarget
	codedeploy: GetOnPremisesInstance
	codedeploy: ListApplicationRevisions
	codedeploy: ListApplications
	codedeploy: ListDeploymentConfigs
	codedeploy: ListDeploymentGroups
	codedeploy: ListDeploymentInstances
	codedeploy: ListDeploymentTargets
	codedeploy: ListDeployments
	codedeploy: ListGitHubAccountTokenNames
	codedeploy: ListOnPremisesInstances
	codedeploy: PutLifecycleEventHookExecutionStatus
	codedeploy: RegisterApplicationRevision
	codedeploy: RegisterOnPremisesInstance
	codedeploy: SkipWaitTimeForInstanceTermination
	codedeploy: StopDeployment
	codedeploy: UpdateApplication
	codedeploy: UpdateDeploymentGroup

Prefisso del servizio	Azioni
codeguru-profiler	profilo codeguru: AddNotificationChannels codeguru profiler: BatchGetFrameMetricData codeguru profiler: CreateProfilingGroup codeguru profiler: DeleteProfilingGroup codeguru profiler: DescribeProfilingGroup codeguru profiler: GetFindingsReportAccountSummary codeguru profiler: GetNotificationConfiguration codeguru profiler: GetPolicy codeguru profiler: GetProfile codeguru profiler: GetRecommendations codeguru profiler: ListFindingsReports codeguru profiler: ListProfileTimes codeguru profiler: ListProfilingGroups codeguru profiler: PutPermission codeguru profiler: RemoveNotificationChannel codeguru profiler: RemovePermission codeguru profiler: SubmitFeedback codeguru profiler: UpdateProfilingGroup

Prefisso del servizio	Azioni
codeguru-reviewer	revisore di codeguru: AssociateRepository
	revisore di codeguru: CreateCodeReview
	revisore di codeguru: DescribeCodeReview
	revisore di codeguru: DescribeRecommendationFeedback
	revisore di codeguru: DescribeRepositoryAssociation
	revisore di codeguru: DisassociateRepository
	revisore di codeguru: ListCodeReviews
	revisore di codeguru: ListRecommendationFeedback
	revisore di codeguru: ListRecommendations
	revisore di codeguru: ListRepositoryAssociations
	revisore di codeguru: PutRecommendationFeedback

Prefisso del servizio	Azioni
codepipeline	pipeline di codici: AcknowledgeJob pipeline di codice: AcknowledgeThirdPartyJob pipeline di codice: CreateCustomActionType pipeline di codice: CreatePipeline pipeline di codice: DeleteCustomActionType pipeline di codice: DeletePipeline pipeline di codice: DeleteWebhook pipeline di codice: DeregisterWebhookWithThirdParty pipeline di codice: GetActionType pipeline di codice: GetJobDetails pipeline di codice: GetPipeline pipeline di codice: GetPipelineExecution pipeline di codice: GetPipelineState pipeline di codice: GetThirdPartyJobDetails pipeline di codice: ListActionExecutions pipeline di codice: ListActionTypes pipeline di codice: ListPipelineExecutions pipeline di codice: ListPipelines pipeline di codice: ListRuleExecutions pipeline di codice: ListRuleTypes pipeline di codice: ListWebhooks

Prefisso del servizio	Azioni
	pipeline di codice: OverrideStageCondition pipeline di codice: PollForJobs pipeline di codice: PollForThirdPartyJobs pipeline di codice: PutActionRevision pipeline di codice: PutApprovalResult pipeline di codice: PutJobFailureResult pipeline di codice: PutJobSuccessResult pipeline di codice: PutThirdPartyJobFailureResult pipeline di codice: PutThirdPartyJobSuccessResult pipeline di codice: PutWebhook pipeline di codice: RegisterWebhookWithThirdParty pipeline di codice: RollbackStage pipeline di codice: StartPipelineExecution pipeline di codice: StopPipelineExecution pipeline di codice: UpdateActionType pipeline di codice: UpdatePipeline

Prefisso del servizio	Azioni
codestar	codestar: AssociateTeamMember codestar: CreateProject codestar: CreateUserProfile codestar: DeleteProject codestar: DeleteUserProfile codestar: DescribeProject codestar: DescribeUserProfile codestar: DisassociateTeamMember codestar: ListProjects codestar: ListResources codestar: ListTeamMembers codestar: ListUserProfiles codestar: UpdateProject codestar: UpdateTeamMember codestar: UpdateUserProfile

Prefisso del servizio	Azioni
codestar-notifications	notifiche codestar: CreateNotificationRule notifiche codestar: DeleteNotificationRule notifiche codestar: DeleteTarget notifiche codestar: DescribeNotificationRule notifiche codestar: ListEventTypes notifiche codestar: ListNotificationRules notifiche codestar: ListTargets codestar-notifications:Subscribe codestar-notifications:Unsubscribe notifiche codestar: UpdateNotificationRule

Prefisso del servizio	Azioni
cognito-identity	identità cognitiva: CreateIdentityPool
	identità cognitiva: DeleteIdentities
	identità cognitiva: DeleteIdentityPool
	identità cognitiva: DescribeIdentity
	identità cognitiva: DescribeIdentityPool
	identità cognitiva: GetIdentityPoolRoles
	identità cognitiva: ListIdentities
	identità cognitiva: ListIdentityPools
	identità cognitiva: LookupDeveloperIdentity
	identità cognitiva: MergeDeveloperIdentities
	identità cognitiva: SetIdentityPoolRoles
	identità cognitiva: UnlinkDeveloperIdentity
	identità cognitiva: UpdateIdentityPool

Prefisso del servizio	Azioni
cognito-idp	cognito-idp: AddCustomAttributes cognito-idp: AddUserPoolClientSecret cognito-idp: AdminAddUserToGroup cognito-idp: AdminConfirmSignUp cognito-idp: AdminCreateUser cognito-idp: AdminDeleteUser cognito-idp: AdminDeleteUserAttributes cognito-idp: AdminDisableProviderForUser cognito-idp: AdminDisableUser cognito-idp: AdminEnableUser cognito-idp: AdminForgetDevice cognito-idp: AdminGetDevice cognito-idp: AdminGetUser cognito-idp: AdminGetUserAuthFactors cognito-idp: AdminInitiateAuth cognito-idp: AdminLinkProviderForUser cognito-idp: AdminListDevices cognito-idp: AdminListGroupsForUser cognito-idp: AdminListUserAuthEvents cognito-idp: AdminRemoveUserFromGroup cognito-idp: AdminResetUserPassword

Prefisso del servizio	Azioni
	cognito-idp: AdminRespondToAuthChallenge cognito-idp: AdminSetUserMFAPreference cognito-idp: AdminSetUserPassword cognito-idp: AdminSetUserSettings cognito-idp: AdminUpdateAuthEventFeedback cognito-idp: AdminUpdateDeviceStatus cognito-idp: AdminUpdateUserAttributes cognito-idp: AdminUserGlobalSignOut cognito-idp: AssociateSoftwareToken cognito-idp: ChangePassword cognito-idp: ConfirmDevice cognito-idp: ConfirmForgotPassword cognito-idp: ConfirmSignUp cognito-idp: CreateGroup cognito-idp: CreateIdentityProvider cognito-idp: CreateManagedLoginBranding cognito-idp: CreateResourceServer cognito-idp: CreateTerms cognito-idp: CreateUserImportJob cognito-idp: CreateUserPool cognito-idp: CreateUserPoolClient

Prefisso del servizio	Azioni
	cognito-idp: CreateUserPoolDomain cognito-idp: CreateUserPoolReplica cognito-idp: DeleteGroup cognito-idp: DeleteIdentityProvider cognito-idp: DeleteManagedLoginBranding cognito-idp: DeleteResourceServer cognito-idp: DeleteTerms cognito-idp: DeleteUser cognito-idp: DeleteUserAttributes cognito-idp: DeleteUserPool cognito-idp: DeleteUserPoolClient cognito-idp: DeleteUserPoolClientSecret cognito-idp: DeleteUserPoolDomain cognito-idp: DeleteUserPoolReplica cognito-idp: DescribeIdentityProvider cognito-idp: DescribeManagedLoginBranding cognito-idp: DescribeManagedLoginBrandingByClient cognito-idp: DescribeResourceServer cognito-idp: DescribeRiskConfiguration cognito-idp: DescribeTerms cognito-idp: DescribeUserImportJob

Prefisso del servizio	Azioni
	cognito-idp: DescribeUserPool cognito-idp: DescribeUserPoolClient cognito-idp: DescribeUserPoolDomain cognito-idp: ForgetDevice cognito-idp: ForgotPassword cognito-idp:GetCSVHeader cognito-idp: GetDevice cognito-idp: GetGroup cognito-idp: GetIdentityProviderByIdentifier cognito-idp: GetLogDeliveryConfiguration cognito-idp: GetProvisionedLimit cognito-idp: GetSigningCertificate cognito-idp:GetUICustomization cognito-idp: GetUser cognito-idp: GetUserAttributeVerificationCode cognito-idp: GetUserPoolMfaConfig cognito-idp: GlobalSignOut cognito-idp: InitiateAuth cognito-idp: ListDevices cognito-idp: ListGroups cognito-idp: ListIdentityProviders

Prefisso del servizio	Azioni
	cognito-idp: ListResourceServers cognito-idp: ListTerms cognito-idp: ListUserImportJobs cognito-idp: ListUserPoolClientSecrets cognito-idp: ListUserPoolClients cognito-idp: ListUserPoolReplicas cognito-idp: ListUserPools cognito-idp: ListUsers cognito-idp: ListUsersInGroup cognito-idp: ResendConfirmationCode cognito-idp: RespondToAuthChallenge cognito-idp: RevokeToken cognito-idp: SetLogDeliveryConfiguration cognito-idp: SetRiskConfiguration cognito-idp:SetUICustomization cognito-idp: SetUserMFAPreference cognito-idp: SetUserPoolMfaConfig cognito-idp: SetUserSettings cognito-idp: SignUp cognito-idp: StartUserImportJob cognito-idp: StopUserImportJob

Prefisso del servizio	Azioni
	cognito-idp: UpdateAuthEventFeedback cognito-idp: UpdateDeviceStatus cognito-idp: UpdateGroup cognito-idp: UpdateIdentityProvider cognito-idp: UpdateProvisionedLimit cognito-idp: UpdateResourceServer cognito-idp: UpdateTerms cognito-idp: UpdateUserAttributes cognito-idp: UpdateUserPool cognito-idp: UpdateUserPoolClient cognito-idp: UpdateUserPoolDomain cognito-idp: UpdateUserPoolReplica cognito-idp: VerifySoftwareToken cognito-idp: VerifyUserAttribute

Prefisso del servizio	Azioni
cognito-sync	sincronizzazione cognitiva: BulkPublish
	sincronizzazione cognitiva: DeleteDataset
	sincronizzazione cognitiva: DescribeDataset
	sincronizzazione cognitiva: DescribeIdentityPoolUsage
	sincronizzazione cognitiva: DescribeIdentityUsage
	sincronizzazione cognitiva: GetBulkPublishDetails
	sincronizzazione cognitiva: GetCognitoEvents
	sincronizzazione cognitiva: GetIdentityPoolConfiguration
	sincronizzazione cognitiva: ListDatasets
	sincronizzazione cognitiva: ListIdentityPoolUsage
	sincronizzazione cognitiva: ListRecords
	sincronizzazione cognitiva: RegisterDevice
	sincronizzazione cognitiva: SetCognitoEvents
	sincronizzazione cognitiva: SetIdentityPoolConfiguration
	sincronizzazione cognitiva: SubscribeToDataset
	sincronizzazione cognitiva: UnsubscribeFromDataset
	sincronizzazione cognitiva: UpdateRecords

Prefisso del servizio	Azioni
comprehendmedical	comprende aspetti medici: DescribeEntitiesDetectionV2Job comprendere la medicina: DescribeICD10CMInferenceJob comprendere la medicina: DescribePHIDetectionJob comprendere la medicina: DescribeRxNormInferenceJob comprendere la medicina: DescribeSNOMEDCTInferenceJob comprendere la medicina: V2 DetectEntities comprehendmedical:DetectPHI comprehendmedical:InferICD10CM comprendere la medicina: InferRxNorm comprehendmedical:InferSNOMEDCT comprendere la medicina: ListEntitiesDetectionV2Jobs comprendere la medicina: ListICD10CMInferenceJobs comprendere la medicina: ListPHIDetectionJobs comprendere la medicina: ListRxNormInferenceJobs comprendere la medicina: ListSNOMEDCTInferenceJobs comprendere la medicina: StartEntitiesDetectionV2Job comprendere la medicina: StartICD10CMInferenceJob comprendere la medicina: StartPHIDetectionJob comprendere la medicina: StartRxNormInferenceJob comprendere la medicina: StartSNOMEDCTInferenceJob comprendere la medicina: StopEntitiesDetectionV2Job

Prefisso del servizio	Azioni
	comprendere la medicina: StopICD10CMInferenceJob comprendere la medicina: StopPHIDetectionJob comprendere la medicina: StopRxNormInferenceJob comprendere la medicina: StopSNOMEDCTInferenceJob

Prefisso del servizio	Azioni
compute-optimizer	ottimizzatore di calcolo: DeleteRecommendationPreferences
	ottimizzatore di calcolo: DescribeRecommendationExportJobs
	ottimizzatore di calcolo: ExportAutoScalingGroupRecommendations
	ottimizzatore di calcolo: ExportEBSVolumeRecommendations
	ottimizzatore di calcolo: ExportEC2InstanceRecommendations
	ottimizzatore di calcolo: ExportECSServiceRecommendations
	ottimizzatore di calcolo: ExportIdleRecommendations
	ottimizzatore di calcolo: ExportLambdaFunctionRecommendations
	ottimizzatore di calcolo: ExportLicenseRecommendations
	ottimizzatore di calcolo: ExportRDSDatabaseRecommendations
	ottimizzatore di calcolo: GetEC2RecommendationProjectedMetrics
	ottimizzatore di calcolo: GetECSServiceRecommendationProjectedMetrics
	ottimizzatore di calcolo: GetEffectiveRecommendationPreferences
	ottimizzatore di calcolo: GetEnrollmentStatus
	ottimizzatore di calcolo: GetEnrollmentStatusesForOrganization
	ottimizzatore di calcolo: GetRDSDatabaseRecommendationProjectedMetrics
	ottimizzatore di calcolo: GetRecommendationPreferences
	ottimizzatore di calcolo: GetRecommendationSummaries
	ottimizzatore di calcolo: PutRecommendationPreferences

Prefisso del servizio	Azioni
	ottimizzatore di calcolo: UpdateEnrollmentStatus

Prefisso del servizio	Azioni
config	configurazione: BatchGetResourceConfig configurazione: DeleteAggregationAuthorization configurazione: DeleteConfigRule configurazione: DeleteConfigurationAggregator configurazione: DeleteConfigurationRecorder configurazione: DeleteConformancePack configurazione: DeleteDeliveryChannel configurazione: DeleteEvaluationResults configurazione: DeleteOrganizationConfigRule configurazione: DeleteOrganizationConformancePack configurazione: DeletePendingAggregationRequest configurazione: DeleteRemediationConfiguration configurazione: DeleteRemediationExceptions configurazione: DeleteResourceConfig configurazione: DeleteRetentionConfiguration configurazione: DeleteStoredQuery configurazione: DeliverConfigSnapshot configurazione: DescribeAggregateComplianceByConfigRules configurazione: DescribeAggregateComplianceByConformancePacks configurazione: DescribeAggregationAuthorizations

Prefisso del servizio	Azioni
	configurazione: DescribeComplianceByConfigRule configurazione: DescribeComplianceByResource configurazione: DescribeConfigRuleEvaluationStatus configurazione: DescribeConfigRules configurazione: DescribeConfigurationAggregatorSourcesStatus configurazione: DescribeConfigurationAggregators configurazione: DescribeConfigurationRecorderStatus configurazione: DescribeConfigurationRecorders configurazione: DescribeConformancePackCompliance configurazione: DescribeConformancePackStatus configurazione: DescribeConformancePacks configurazione: DescribeDeliveryChannelStatus configurazione: DescribeDeliveryChannels configurazione: DescribeOrganizationConfigRuleStatuses configurazione: DescribeOrganizationConfigRules configurazione: DescribeOrganizationConformancePackStatuses configurazione: DescribeOrganizationConformancePacks configurazione: DescribePendingAggregationRequests configurazione: DescribeRemediationConfigurations configurazione: DescribeRemediationExceptions configurazione: DescribeRemediationExecutionStatus

Prefisso del servizio	Azioni
	configurazione: DescribeRetentionConfigurations configurazione: GetComplianceDetailsByConfigRule configurazione: GetComplianceDetailsByResource configurazione: GetComplianceSummaryByConfigRule configurazione: GetComplianceSummaryByResourceType configurazione: GetConformancePackComplianceDetails configurazione: GetConformancePackComplianceSummary configurazione: GetCustomRulePolicy configurazione: GetDiscoveredResourceCounts configurazione: GetOrganizationConfigRuleDetailedStatus configurazione: GetOrganizationConformancePackDetailedStatus configurazione: GetOrganizationCustomRulePolicy configurazione: GetResourceConfigHistory configurazione: GetResourceEvaluationSummary configurazione: GetStoredQuery configurazione: ListConfigurationRecorders configurazione: ListConformancePackComplianceScores configurazione: ListConnectors configurazione: ListDiscoveredResources configurazione: ListResourceEvaluations configurazione: ListStoredQueries

Prefisso del servizio	Azioni
	configurazione: PutConfigRule configurazione: PutConfigurationAggregator configurazione: PutConfigurationRecorder configurazione: PutConformancePack configurazione: PutConnector configurazione: PutDeliveryChannel configurazione: PutEvaluations configurazione: PutExternalEvaluation configurazione: PutOrganizationConfigRule configurazione: PutOrganizationConformancePack configurazione: PutRemediationConfigurations configurazione: PutRemediationExceptions configurazione: PutResourceConfig configurazione: PutRetentionConfiguration configurazione: PutStoredQuery configurazione: SelectResourceConfig configurazione: StartConfigRulesEvaluation configurazione: StartConfigurationRecorder configurazione: StartRemediationExecution configurazione: StartResourceEvaluation configurazione: StopConfigurationRecorder

Prefisso del servizio	Azioni
connect	connettere: ActivateEvaluationForm
	connettere: AssociateAnalyticsDataSet
	connettere: AssociateApprovedOrigin
	connettere: AssociateBot
	connettere: AssociateContactWithUser
	connettere: AssociateDefaultVocabulary
	connettere: AssociateEmailAddressAlias
	connettere: AssociateFlow
	connettere: AssociateInstanceStorageConfig
	connettere: AssociateLambdaFunction
	connettere: AssociateLexBot
	connettere: AssociatePhoneNumberContactFlow
	connettere: AssociateQueueEmailAddresses
	connettere: AssociateQueueQuickConnects
	connettere: AssociateRoutingProfileQueues
	connettere: AssociateSecurityKey
	connettere: AssociateUserProficiencies
	connettere: BatchAssociateAnalyticsDataSet
	connettere: BatchCreateDataTableValue
	connettere: BatchDeleteDataTableValue
	connettere: BatchDescribeDataTableValue

Prefisso del servizio	Azioni
	connettere: BatchDisassociateAnalyticsDataSet
	connettere: BatchGetFlowAssociation
	connettere: BatchPutContact
	connettere: BatchUpdateDataTableValue
	connettere: ClaimPhoneNumber
	connettere: CreateAgentStatus
	connettere: CreateAttachedFile
	connettere: CreateContact
	connettere: CreateContactFlow
	connettere: CreateContactFlowModule
	connettere: CreateContactFlowModuleAlias
	connettere: CreateContactFlowModuleVersion
	connettere: CreateContactFlowVersion
	connettere: CreateDataTable
	connettere: CreateDataTableAttribute
	connettere: CreateEmailAddress
	connettere: CreateEvaluationForm
	connettere: CreateHoursOfOperation
	connettere: CreateInstance
	connettere: CreateIntegrationAssociation
	connettere: CreateParticipant

Prefisso del servizio	Azioni
	connettere: CreatePersistentContactAssociation
	connettere: CreatePredefinedAttribute
	connettere: CreatePrompt
	connettere: CreatePushNotificationRegistration
	connettere: CreateQueue
	connettere: CreateQuickConnect
	connettere: CreateRoutingProfile
	connettere: CreateRule
	connettere: CreateSecurityProfile
	connettere: CreateTaskTemplate
	connettere: CreateTrafficDistributionGroup
	connettere: CreateUseCase
	connettere: CreateUser
	connettere: CreateUserHierarchyGroup
	connettere: CreateView
	connettere: CreateViewVersion
	connettere: CreateVocabulary
	connettere: CreateWorkspace
	connettere: DeactivateEvaluationForm
	connettere: DeleteContactEvaluation
	connettere: DeleteContactFlow

Prefisso del servizio	Azioni
	connettere: DeleteContactFlowModule
	connettere: DeleteContactFlowModuleAlias
	connettere: DeleteContactFlowModuleVersion
	connettere: DeleteContactFlowVersion
	connettere: DeleteDataTable
	connettere: DeleteDataTableAttribute
	connettere: DeleteEmailAddress
	connettere: DeleteEvaluationForm
	connettere: DeleteHoursOfOperation
	connettere: DeleteHoursOfOperationOverride
	connettere: DeleteInstance
	connettere: DeleteIntegrationAssociation
	connettere: DeleteNotification
	connettere: DeletePredefinedAttribute
	connettere: DeletePrompt
	connettere: DeletePushNotificationRegistration
	connettere: DeleteQueue
	connettere: DeleteQuickConnect
	connettere: DeleteRoutingProfile
	connettere: DeleteRule
	connettere: DeleteSecurityProfile

Prefisso del servizio	Azioni
	connettere: DeleteTaskTemplate
	connettere: DeleteTrafficDistributionGroup
	connettere: DeleteUseCase
	connettere: DeleteUser
	connettere: DeleteUserHierarchyGroup
	connettere: DeleteView
	connettere: DeleteVocabulary
	connettere: DeleteWorkspace
	connettere: DeleteWorkspaceMedia
	connettere: DescribeAttachedFilesConfiguration
	connettere: DescribeAuthenticationProfile
	connettere: DescribeContactFlowModuleAlias
	connettere: DescribeDataTableAttribute
	connettere: DescribeHoursOfOperationOverride
	connettere: DescribeInstanceAttribute
	connettere: DescribeInstanceStorageConfig
	connettere: DescribeNotification
	connettere: DescribePhoneNumber
	connettere: DescribeRule
	connettere: DescribeTrafficDistributionGroup
	connettere: DescribeUserHierarchyStructure

Prefisso del servizio	Azioni
	connettere: DescribeVocabulary
	connettere: DisassociateAnalyticsDataSet
	connettere: DisassociateApprovedOrigin
	connettere: DisassociateBot
	connettere: DisassociateEmailAddressAlias
	connettere: DisassociateFlow
	connettere: DisassociateInstanceStorageConfig
	connettere: DisassociateLambdaFunction
	connettere: DisassociateLexBot
	connettere: DisassociatePhoneNumberContactFlow
	connettere: DisassociateQueueEmailAddresses
	connettere: DisassociateQueueQuickConnects
	connettere: DisassociateRoutingProfileQueues
	connettere: DisassociateSecurityKey
	connettere: DisassociateUserProficiencies
	connettere: DismissUserContact
	connettere: EvaluateDataTableValues
	connettere: GetContactAttributes
	connettere: GetContactMetrics
	connettere: GetCurrentMetricData
	connettere: GetCurrentUserData

Prefisso del servizio	Azioni
	connettere: GetEffectiveHoursOfOperations
	connettere: GetEvaluationFormValidation
	connettere: GetFederationToken
	connettere: GetFlowAssociation
	connettere: GetMetricData
	connettere: GetMetricDataV2
	connettere: GetPromptFile
	connettere: GetTaskTemplate
	connettere: GetTrafficDistribution
	connettere: ImportPhoneNumber
	connettere: ImportWorkspaceMedia
	connettere: ListAnalyticsDataAssociations
	connettere: ListAnalyticsDataLakeDataSets
	connettere: ListApprovedOrigins
	connettere: ListAssociatedContacts
	connettere: ListAttachedFilesConfigurations
	connettere: ListAuthenticationProfiles
	connettere: ListBots
	connettere: ListContactEvaluations
	connettere: ListContactFlowModuleAliases
	connettere: ListContactFlowModuleVersions

Prefisso del servizio	Azioni
	connettere: ListContactFlowModules
	connettere: ListContactFlowVersions
	connettere: ListContactFlows
	connettere: ListContactReferences
	connettere: ListDataTableAttributes
	connettere: ListDataTablePrimaryValues
	connettere: ListDataTableValues
	connettere: ListDataTables
	connettere: ListDefaultVocabularies
	connettere: ListEvaluationFormVersions
	connettere: ListEvaluationForms
	connettere: ListFlowAssociations
	connettere: ListHoursOfOperations
	connettere: ListInstanceAttributes
	connettere: ListInstanceStorageConfigs
	connettere: ListIntegrationAssociations
	connettere: ListLambdaFunctions
	connettere: ListLexBots
	connettere: ListNotifications
	connettere: ListPhoneNumbers
	connettere: ListPhoneNumbersV2

Prefisso del servizio	Azioni
	connettere: ListPredefinedAttributes
	connettere: ListPrompts
	connettere: ListQueueEmailAddresses
	connettere: ListQueueQuickConnects
	connettere: ListQueues
	connettere: ListQuickConnects
	connettere: ListRealtimeContactAnalysisSegmentsV2
	connettere: ListRoutingProfileManualAssignmentQueues
	connettere: ListRoutingProfileQueues
	connettere: ListRoutingProfiles
	connettere: ListRules
	connettere: ListSecurityKeys
	connettere: ListSecurityProfileApplications
	connettere: ListSecurityProfileFlowModules
	connettere: ListSecurityProfilePermissions
	connettere: ListSecurityProfiles
	connettere: ListTaskTemplates
	connettere: ListTrafficDistributionGroups
	connettere: ListUseCases
	connettere: ListUserHierarchyGroups
	connettere: ListUserNotifications

Prefisso del servizio	Azioni
	connettere: ListUsers
	connettere: ListViewVersions
	connettere: ListViews
	connettere: ListWorkspaceMedia
	connettere: ListWorkspacePages
	connettere: ListWorkspaces
	connettere: MonitorContact
	connettere: PauseContact
	connettere: PutUserStatus
	connettere: ReleasePhoneNumber
	connettere: ReplicateInstance
	connettere: ResumeContact
	connettere: ResumeContactRecording
	connettere: SearchAgentStatuses
	connettere: SearchAvailablePhoneNumbers
	connettere: SearchContactEvaluations
	connettere: SearchContactFlowModules
	connettere: SearchContactFlows
	connettere: SearchContacts
	connettere: SearchDataTables
	connettere: SearchEmailAddresses

Prefisso del servizio	Azioni
	connettere: SearchEvaluationForms
	connettere: SearchHoursOfOperations
	connettere: SearchPredefinedAttributes
	connettere: SearchPrompts
	connettere: SearchQueues
	connettere: SearchQuickConnects
	connettere: SearchRoutingProfiles
	connettere: SearchRules
	connettere: SearchSecurityProfiles
	connettere: SearchUserHierarchyGroups
	connettere: SearchViews
	connettere: SearchVocabularies
	connettere: SearchWorkspaceAssociations
	connettere: SearchWorkspaces
	connettere: SendChatIntegrationEvent
	connettere: SendOutboundEmail
	connettere: StartChatContact
	connettere: StartContactConversationalAnalyticsJob
	connettere: StartContactEvaluation
	connettere: StartContactMediaProcessing
	connettere: StartContactRecording

Prefisso del servizio	Azioni
	connettere: StartContactStreaming
	connettere: StartEmailContact
	connettere: StartOutboundChatContact
	connettere: StartOutboundEmailContact
	connettere: StartOutboundVoiceContact
	connettere: StartScreenSharing
	connettere: StartTaskContact
	connettere: StartWebRTCContact
	connettere: StopContact
	connettere: StopContactMediaProcessing
	connettere: StopContactRecording
	connettere: StopContactStreaming
	connettere: SubmitContactEvaluation
	connettere: SuspendContactRecording
	connettere: TransferContact
	connettere: UpdateAgentStatus
	connettere: UpdateAttachedFilesConfiguration
	connettere: UpdateAuthenticationProfile
	connettere: UpdateContact
	connettere: UpdateContactAttributes
	connettere: UpdateContactEvaluation

Prefisso del servizio	Azioni
	connettere: UpdateContactFlowContent
	connettere: UpdateContactFlowMetadata
	connettere: UpdateContactFlowModuleAlias
	connettere: UpdateContactFlowModuleContent
	connettere: UpdateContactFlowModuleMetadata
	connettere: UpdateContactFlowName
	connettere: UpdateContactRoutingData
	connettere: UpdateContactSchedule
	connettere: UpdateDataTableAttribute
	connettere: UpdateDataTableMetadata
	connettere: UpdateDataTablePrimaryValues
	connettere: UpdateEmailAddressMetadata
	connettere: UpdateEvaluationForm
	connettere: UpdateHoursOfOperation
	connettere: UpdateHoursOfOperationOverride
	connettere: UpdateInstanceAttribute
	connettere: UpdateInstanceStorageConfig
	connettere: UpdateNotificationContent
	connettere: UpdateParticipantAuthentication
	connettere: UpdateParticipantRoleConfig
	connettere: UpdatePhoneNumber

Prefisso del servizio	Azioni
	connettere: UpdatePhoneNumberMetadata
	connettere: UpdatePredefinedAttribute
	connettere: UpdatePrompt
	connettere: UpdateQueueHoursOfOperation
	connettere: UpdateQueueMaxContacts
	connettere: UpdateQueueName
	connettere: UpdateQueueOutboundCallerConfig
	connettere: UpdateQueueOutboundEmailConfig
	connettere: UpdateQueueStatus
	connettere: UpdateQuickConnectConfig
	connettere: UpdateQuickConnectName
	connettere: UpdateRoutingProfileAgentAvailabilityTimer
	connettere: UpdateRoutingProfileConcurrency
	connettere: UpdateRoutingProfileDefaultOutboundQueue
	connettere: UpdateRoutingProfileName
	connettere: UpdateRoutingProfileQueues
	connettere: UpdateRule
	connettere: UpdateSecurityProfile
	connettere: UpdateTaskTemplate
	connettere: UpdateTrafficDistribution
	connettere: UpdateUserHierarchy

Prefisso del servizio	Azioni
	connettere: UpdateUserHierarchyGroupName connettere: UpdateUserHierarchyStructure connettere: UpdateUserIdentityInfo connettere: UpdateUserNotificationStatus connettere: UpdateUserPhoneConfig connettere: UpdateUserProficiencies connettere: UpdateUserRoutingProfile connettere: UpdateUserSecurityProfiles connettere: UpdateViewContent connettere: UpdateViewMetadata connettere: UpdateWorkspaceMetadata connettere: UpdateWorkspaceTheme connettere: UpdateWorkspaceVisibility
cur	cura: DeleteReportDefinition cura: DescribeReportDefinitions cura: ModifyReportDefinition cura: PutReportDefinition

Prefisso del servizio	Azioni
databrew	data brew: BatchDeleteRecipeVersion
	data brew: CreateDataset
	data brew: CreateProfileJob
	data brew: CreateProject
	data brew: CreateRecipe
	data brew: CreateRecipeJob
	data brew: CreateRuleset
	data brew: CreateSchedule
	data brew: DeleteDataset
	data brew: DeleteJob
	data brew: DeleteProject
	data brew: DeleteRecipeVersion
	data brew: DeleteRuleset
	data brew: DeleteSchedule
	data brew: DescribeDataset
	data brew: DescribeJob
	data brew: DescribeJobRun
	data brew: DescribeProject
	data brew: DescribeRecipe
	data brew: DescribeRuleset
	data brew: DescribeSchedule

Prefisso del servizio	Azioni
	data brew: ListDatasets
	data brew: ListJobRuns
	data brew: ListJobs
	data brew: ListProjects
	data brew: ListRecipeVersions
	data brew: ListRecipes
	data brew: ListRulesets
	data brew: ListSchedules
	data brew: PublishRecipe
	data brew: SendProjectSessionAction
	data brew: StartJobRun
	data brew: StartProjectSession
	data brew: StopJobRun
	data brew: UpdateDataset
	data brew: UpdateProfileJob
	data brew: UpdateProject
	data brew: UpdateRecipe
	data brew: UpdateRecipeJob
	data brew: UpdateRuleset
	data brew: UpdateSchedule

Prefisso del servizio	Azioni
dataexchange	scambio di dati: AcceptDataGrant scambio di dati: CancelJob scambio di dati: CreateDataGrant scambio di dati: CreateDataSet scambio di dati: CreateEventAction scambio di dati: CreateJob scambio di dati: CreateRevision scambio di dati: DeleteAsset scambio di dati: DeleteDataGrant scambio di dati: DeleteEventAction scambio di dati: DeleteRevision scambio di dati: GetDataGrant scambio di dati: GetEventAction scambio di dati: GetJob scambio di dati: GetReceivedDataGrant scambio di dati: ListDataGrants scambio di dati: ListDataSetRevisions scambio di dati: ListDataSets scambio di dati: ListEventActions scambio di dati: ListJobs scambio di dati: ListReceivedDataGrants

Prefisso del servizio	Azioni
	scambio di dati: ListRevisionAssets scambio di dati: RevokeRevision scambio di dati: SendDataSetNotification scambio di dati: StartJob scambio di dati: UpdateAsset scambio di dati: UpdateDataSet scambio di dati: UpdateEventAction scambio di dati: UpdateRevision

Prefisso del servizio	Azioni
datapipeline	pipeline dati: ActivatePipeline
	tubazione dati: CreatePipeline
	tubazione dati: DeactivatePipeline
	tubazione dati: DeletePipeline
	tubazione dati: DescribeObjects
	tubazione dati: DescribePipelines
	tubazione dati: EvaluateExpression
	tubazione dati: GetPipelineDefinition
	tubazione dati: ListPipelines
	tubazione dati: PollForTask
	tubazione dati: PutPipelineDefinition
	tubazione dati: QueryObjects
	tubazione dati: ReportTaskProgress
	tubazione dati: ReportTaskRunnerHeartbeat
	tubazione dati: SetStatus
	tubazione dati: SetTaskStatus
	tubazione dati: ValidatePipelineDefinition

Prefisso del servizio	Azioni
dax	fax: CreateCluster fax: DecreaseReplicationFactor fax: DeleteCluster fax: DeleteParameterGroup fax: DeleteSubnetGroup fax: DescribeClusters fax: DescribeDefaultParameters fax: DescribeEvents fax: DescribeParameterGroups fax: DescribeParameters fax: DescribeSubnetGroups fax: IncreaseReplicationFactor fax: RebootNode fax: UpdateCluster fax: UpdateParameterGroup fax: UpdateSubnetGroup

Prefisso del servizio	Azioni
devicefarm	fabbrica di dispositivi: CreateDevicePool
	fabbrica di dispositivi: CreateInstanceProfile
	fabbrica di dispositivi: CreateNetworkProfile
	fabbrica di dispositivi: CreateProject
	fabbrica di dispositivi: CreateRemoteAccessSession
	fabbrica di dispositivi: CreateTestGridProject
	fabbrica di dispositivi: CreateTestGridUrl
	fabbrica di dispositivi: CreateUpload
	devicefarm:CreateVPCEConfiguration
	fabbrica di dispositivi: DeleteDevicePool
	fabbrica di dispositivi: DeleteInstanceProfile
	fabbrica di dispositivi: DeleteNetworkProfile
	fabbrica di dispositivi: DeleteProject
	fabbrica di dispositivi: DeleteRemoteAccessSession
	fabbrica di dispositivi: DeleteRun
	fabbrica di dispositivi: DeleteTestGridProject
	fattoria di dispositivi: DeleteUpload
	devicefarm:DeleteVPCEConfiguration
	fattoria di dispositivi: GetAccountSettings
	fattoria di dispositivi: GetDevice
	fattoria di dispositivi: GetDeviceInstance

Prefisso del servizio	Azioni
	fattoria di dispositivi: GetDevicePool
	fattoria di dispositivi: GetDevicePoolCompatibility
	fattoria di dispositivi: GetInstanceProfile
	fattoria di dispositivi: GetJob
	fattoria di dispositivi: GetNetworkProfile
	fattoria di dispositivi: GetOfferingStatus
	fattoria di dispositivi: GetProject
	fattoria di dispositivi: GetRemoteAccessSession
	fattoria di dispositivi: GetRun
	fattoria di dispositivi: GetSuite
	fattoria di dispositivi: GetTest
	fattoria di dispositivi: GetTestGridProject
	fattoria di dispositivi: GetTestGridSession
	fattoria di dispositivi: GetUpload
	devicefarm:GetVPCEConfiguration
	fattoria di dispositivi: ListArtifacts
	fattoria di dispositivi: ListDeviceInstances
	fattoria di dispositivi: ListDevicePools
	fattoria di dispositivi: ListDevices
	fattoria di dispositivi: ListInstanceProfiles
	fattoria di dispositivi: ListJobs

Prefisso del servizio	Azioni
	fattoria di dispositivi: ListNetworkProfiles
	fattoria di dispositivi: ListOfferingPromotions
	fattoria di dispositivi: ListOfferingTransactions
	fattoria di dispositivi: ListOfferings
	fattoria di dispositivi: ListProjects
	fattoria di dispositivi: ListRemoteAccessSessions
	fattoria di dispositivi: ListRuns
	fattoria di dispositivi: ListSamples
	fattoria di dispositivi: ListSuites
	fattoria di dispositivi: ListTestGridProjects
	fattoria di dispositivi: ListTestGridSessionActions
	fattoria di dispositivi: ListTestGridSessionArtifacts
	fattoria di dispositivi: ListTestGridSessions
	fattoria di dispositivi: ListTests
	fattoria di dispositivi: ListUniqueProblems
	fattoria di dispositivi: ListUploads
	devicefarm:ListVPCEConfigurations
	fattoria di dispositivi: PurchaseOffering
	fattoria di dispositivi: RenewOffering
	fattoria di dispositivi: ScheduleRun
	fattoria di dispositivi: StopJob

Prefisso del servizio	Azioni
	fattoria di dispositivi: StopRemoteAccessSession
	fattoria di dispositivi: StopRun
	fattoria di dispositivi: UpdateDeviceInstance
	fattoria di dispositivi: UpdateDevicePool
	fattoria di dispositivi: UpdateInstanceProfile
	fattoria di dispositivi: UpdateNetworkProfile
	fattoria di dispositivi: UpdateProject
	fattoria di dispositivi: UpdateTestGridProject
	fattoria di dispositivi: UpdateUpload
	devicefarm:UpdateVPCEConfiguration

Prefisso del servizio	Azioni
devops-guru	devops-guru: AddNotificationChannel devops-guru: DeleteInsight devops-guru: DescribeAccountHealth devops-guru: DescribeAccountOverview devops-guru: DescribeAnomaly devops-guru: DescribeEventSourcesConfig devops-guru: DescribeFeedback devops-guru: DescribeInsight devops-guru: DescribeOrganizationHealth devops-guru: DescribeOrganizationOverview devops-guru: DescribeOrganizationResourceCollectionHealth devops-guru: DescribeResourceCollectionHealth devops-guru: DescribeServiceIntegration devops-guru: GetCostEstimation devops-guru: GetResourceCollection devops-guru: ListAnomaliesForInsight devops-guru: ListAnomalousLogGroups devops-guru: ListEvents devops-guru: ListInsights devops-guru: ListMonitoredResources devops-guru: ListNotificationChannels

Prefisso del servizio	Azioni
	devops-guru: ListOrganizationInsights devops-guru: ListRecommendations devops-guru: PutFeedback devops-guru: RemoveNotificationChannel devops-guru: SearchInsights devops-guru: SearchOrganizationInsights devops-guru: StartCostEstimation devops-guru: UpdateEventSourcesConfig devops-guru: UpdateResourceCollection devops-guru: UpdateServiceIntegration

Prefisso del servizio	Azioni
directconnect	connessione diretta: AcceptDirectConnectGatewayAssociationProposal
	connessione diretta: AllocateConnectionOnInterconnect
	connessione diretta: AllocateHostedConnection
	connessione diretta: AllocatePrivateVirtualInterface
	connessione diretta: AllocatePublicVirtualInterface
	connessione diretta: AllocateTransitVirtualInterface
	connessione diretta: AssociateConnectionWithLag
	connessione diretta: AssociateHostedConnection
	connessione diretta: AssociateMacSecKey
	connessione diretta: AssociateVirtualInterface
	connessione diretta: ConfirmConnection
	connessione diretta: ConfirmCustomerAgreement
	connessione diretta: ConfirmPrivateVirtualInterface
	connessione diretta: ConfirmPublicVirtualInterface
	connessione diretta: ConfirmTransitVirtualInterface
	directconnect:CreateBGPPeer
	connessione diretta: CreateConnection
	connessione diretta: CreateDirectConnectGateway
	connessione diretta: CreateDirectConnectGatewayAssociation
	connessione diretta: CreateDirectConnectGatewayAssociationProposal

Prefisso del servizio	Azioni
	connessione diretta: CreateInterconnect
	connessione diretta: CreateLag
	connessione diretta: CreatePrivateVirtualInterface
	connessione diretta: CreatePublicVirtualInterface
	connessione diretta: CreateTransitVirtualInterface
	directconnect:DeleteBGPPeer
	connessione diretta: DeleteConnection
	connessione diretta: DeleteDirectConnectGateway
	connessione diretta: DeleteDirectConnectGatewayAssociation
	connessione diretta: DeleteDirectConnectGatewayAssociationProposal
	connessione diretta: DeleteInterconnect
	connessione diretta: DeleteLag
	connessione diretta: DeleteVirtualInterface
	connessione diretta: DescribeConnectionLoa
	connessione diretta: DescribeConnections
	connessione diretta: DescribeConnectionsOnInterconnect
	connessione diretta: DescribeCustomerMetadata
	connessione diretta: DescribeDirectConnectGatewayAssociationProposals
	connessione diretta: DescribeDirectConnectGatewayAssociations
	connessione diretta: DescribeDirectConnectGatewayAttachments

Prefisso del servizio	Azioni
	connessione diretta: DescribeDirectConnectGateways
	connessione diretta: DescribeHostedConnections
	connessione diretta: DescribeInterconnectLoa
	connessione diretta: DescribeInterconnects
	connessione diretta: DescribeLags
	connessione diretta: DescribeLoa
	connessione diretta: DescribeLocations
	connessione diretta: DescribeRouterConfiguration
	connessione diretta: DescribeVirtualGateways
	connessione diretta: DescribeVirtualInterfaces
	connessione diretta: DisassociateConnectionFromLag
	connessione diretta: DisassociateMacSecKey
	connessione diretta: ListVirtualInterfaceTestHistory
	connessione diretta: StartBgpFailoverTest
	connessione diretta: StopBgpFailoverTest
	connessione diretta: UpdateConnection
	connessione diretta: UpdateDirectConnectGateway
	connessione diretta: UpdateDirectConnectGatewayAssociation
	connessione diretta: UpdateLag
	connessione diretta: UpdateVirtualInterfaceAttributes

Prefisso del servizio	Azioni
dlm	dm: CreateLifecyclePolicy dpm: DeleteLifecyclePolicy dpm: GetLifecyclePolicies dpm: GetLifecyclePolicy dpm: UpdateLifecyclePolicy

Prefisso del servizio	Azioni
dms	dighe: ApplyPendingMaintenanceAction
	dms: AssociateExtensionPack
	dms: BatchStartRecommendations
	dms: CancelMetadataModelCreation
	dms: CancelReplicationTaskAssessmentRun
	dms: CreateDataProvider
	dms: CreateEndpoint
	dms: CreateEventSubscription
	dms: CreateInstanceProfile
	dms: CreateMigrationProject
	dms: CreateReplicationConfig
	dms: CreateReplicationInstance
	dms: CreateReplicationSubnetGroup
	dms: CreateReplicationTask
	dms: DeleteCertificate
	dms: DeleteConnection
	dms: DeleteDataMigration
	dms: DeleteDataProvider
	dms: DeleteEndpoint
	dms: DeleteEventSubscription
	dms: DeleteFleetAdvisorCollector

Prefisso del servizio	Azioni
	dms: DeleteFleetAdvisorDatabases
	dms: DeleteInstanceProfile
	dms: DeleteMigrationProject
	dms: DeleteReplicationConfig
	dms: DeleteReplicationInstance
	dms: DeleteReplicationSubnetGroup
	dms: DeleteReplicationTask
	dms: DeleteReplicationTaskAssessmentRun
	dms: DescribeAccountAttributes
	dms: DescribeApplicableIndividualAssessments
	dms: DescribeCertificates
	dms: DescribeConnections
	dms: DescribeDataMigrations
	dms: DescribeEndpointSettings
	dms: DescribeEndpointTypes
	dms: DescribeEndpoints
	dms: DescribeEngineVersions
	dms: DescribeEventCategories
	dms: DescribeEventSubscriptions
	dms: DescribeEvents
	dms: DescribeFleetAdvisorCollectors

Prefisso del servizio	Azioni
	dms: DescribeFleetAdvisorDatabases
	dms: DescribeFleetAdvisorLsaAnalysis
	dms: DescribeFleetAdvisorSchemaObjectSummary
	dms: DescribeFleetAdvisorSchemas
	dms: DescribeMetadataModel
	dms: DescribeMetadataModelChildren
	dms: DescribeMetadataModelCreations
	dms: DescribeMetadataModelImports
	dms: DescribeOrderableReplicationInstances
	dms: DescribePendingMaintenanceActions
	dms: DescribeRecommendationLimitations
	dms: DescribeRecommendations
	dms: DescribeRefreshSchemasStatus
	dms: DescribeReplicationConfigs
	dms: DescribeReplicationInstanceTaskLogs
	dms: DescribeReplicationInstances
	dms: DescribeReplicationSubnetGroups
	dms: DescribeReplicationTableStatistics
	dms: DescribeReplicationTaskAssessmentResults
	dms: DescribeReplicationTaskAssessmentRuns
	dms: DescribeReplicationTaskIndividualAssessments

Prefisso del servizio	Azioni
	dms: DescribeReplicationTasks
	dms: DescribeReplications
	dms: DescribeSchemas
	dms: DescribeTableStatistics
	dms: ExportMetadataModelAssessment
	dms: ImportCertificate
	dms: ListDataProviders
	dms: ListExtensionPacks
	dms: ListInstanceProfiles
	dms: ListMetadataModelAssessments
	dms: ListMetadataModelConversions
	dms: ListMetadataModelExports
	dms: ListMigrationProjects
	dms: ModifyDataMigration
	dms: ModifyEndpoint
	dms: ModifyEventSubscription
	dms: ModifyReplicationConfig
	dms: ModifyReplicationInstance
	dms: ModifyReplicationSubnetGroup
	dms: ModifyReplicationTask
	dms: MoveReplicationTask

Prefisso del servizio	Azioni
	dms: RebootReplicationInstance
	dms: RefreshSchemas
	dms: ReloadReplicationTables
	dms: ReloadTables
	dms: RunFleetAdvisorLsaAnalysis
	dms: StartMetadataModelAssessment
	dms: StartMetadataModelConversion
	dms: StartMetadataModelCreation
	dms: StartMetadataModelExportAsScripts
	dms: StartMetadataModelExportToTarget
	dms: StartRecommendations
	dms: StartReplication
	dms: StartReplicationTask
	dms: StartReplicationTaskAssessment
	dms: StopDataMigration
	dms: StopReplicationTask
	dms: TestConnection
	dms: UpdateConversionConfiguration
	dms: UpdateDataProvider
	dms: UpdateInstanceProfile
	dms: UpdateMigrationProject

Prefisso del servizio	Azioni
	dms: UpdateSubscriptionsToEventBridge
docdb-elastic	docdb elastico: ApplyPendingMaintenanceAction docdb elastico: CopyClusterSnapshot docdb elastico: DeleteCluster docdb elastico: DeleteClusterSnapshot docdb elastico: GetCluster docdb elastico: GetClusterSnapshot docdb elastico: GetPendingMaintenanceAction docdb elastico: ListClusterSnapshots docdb elastico: ListClusters docdb elastico: ListPendingMaintenanceActions docdb elastico: RestoreClusterFromSnapshot docdb elastico: StartCluster docdb elastico: StopCluster docdb elastico: UpdateCluster

Prefisso del servizio	Azioni
dynamodb	dinamodb: CreateBackup
	dinamodb: CreateGlobalTable
	dinamodb: CreateTable
	dinamodb: DeleteBackup
	dinamodb: DeleteTable
	dinamodb: DescribeBackup
	dinamodb: DescribeContinuousBackups
	dinamodb: DescribeContributorInsights
	dinamodb: DescribeEndpoints
	dinamodb: DescribeExport
	dinamodb: DescribeGlobalTable
	dinamodb: DescribeGlobalTableSettings
	dinamodb: DescribeImport
	dinamodb: DescribeKinesisStreamingDestination
	dinamodb: DescribeLimits
	dinamodb: DescribeStream
	dinamodb: DescribeTable
	dinamodb: DescribeTableReplicaAutoScaling
	dinamodb: DescribeTimeToLive
	dinamodb: DisableKinesisStreamingDestination
	dinamodb: EnableKinesisStreamingDestination

Prefisso del servizio	Azioni
	dinamodb: ExportTableToPointInTime
	dinamodb: GetResourcePolicy
	dinamodb: ImportTable
	dinamodb: ListBackups
	dinamodb: ListContributorInsights
	dinamodb: ListExports
	dinamodb: ListGlobalTables
	dinamodb: ListImports
	dinamodb: ListStreams
	dinamodb: ListTables
	dinamodb: RestoreTableFromBackup
	dinamodb: RestoreTableToPointInTime
	dinamodb: UpdateContinuousBackups
	dinamodb: UpdateContributorInsights
	dinamodb: UpdateGlobalTable
	dinamodb: UpdateGlobalTableSettings
	dinamodb: UpdateKinesisStreamingDestination
	dinamodb: UpdateTable
	dinamodb: UpdateTableReplicaAutoScaling
	dinamodb: UpdateTimeToLive

Prefisso del servizio	Azioni
ebs	Web: CompleteSnapshot ebs: StartSnapshot

Prefisso del servizio	Azioni
ec2	ec2: AcceptAddressTransfer
	ec2: AcceptCapacityReservationBillingOwnership
	ec2: AcceptReservedInstancesExchangeQuote
	ec2: AcceptTransitGatewayClientVpnAttachment
	ec2: AcceptTransitGatewayMulticastDomainAssociations
	ec2: AcceptTransitGatewayPeeringAttachment
	ec2: AcceptTransitGatewayVpcAttachment
	ec2: AcceptVpcEndpointConnections
	ec2: AcceptVpcPeeringConnection
	ec2: AdvertiseByoipCidr
	ec2: AllocateAddress
	ec2: AllocateHosts
	ec2: AllocateIpamPoolCidr
	ec2: ApplySecurityGroupsToClientVpnTargetNetwork
	ec2: AssignIpv6Addresses
	ec2: AssignPrivateIpAddresses
	ec2: AssignPrivateNatGatewayAddress
	ec2: AssociateAddress
	ec2: AssociateCapacityReservationBillingOwner
	ec2: AssociateClientVpnTargetNetwork
	ec2: AssociateDhcpOptions

Prefisso del servizio	Azioni
	ec2: AssociateEnclaveCertificateIamRole
	ec2: AssociateIamInstanceProfile
	ec2: AssociateInstanceEventWindow
	ec2: AssociateIpamByoasn
	ec2: AssociateIpamResourceDiscovery
	ec2: AssociateNatGatewayAddress
	ec2: AssociateRouteServer
	ec2: AssociateRouteTable
	ec2: AssociateSecurityGroupVpc
	ec2: AssociateSubnetCidrBlock
	ec2: AssociateTransitGatewayMulticastDomain
	ec2: AssociateTransitGatewayPolicyTable
	ec2: AssociateTransitGatewayRouteTable
	ec2: AssociateTrunkInterface
	ec2: AssociateVpcCidrBlock
	ec2: AttachClassicLinkVpc
	ec2: AttachImageWatermark
	ec2: AttachInternetGateway
	ec2: AttachNetworkInterface
	ec2: AttachVerifiedAccessTrustProvider
	ec2: AttachVolume

Prefisso del servizio	Azioni
	ec2: AttachVpnGateway
	ec2: AuthorizeClientVpnIngress
	ec2: AuthorizeSecurityGroupEgress
	ec2: AuthorizeSecurityGroupIngress
	ec2: BundleInstance
	ec2: CancelBundleTask
	ec2: CancelCapacityReservation
	ec2: CancelCapacityReservationFleets
	ec2: CancelConversionTask
	ec2: CancelDeclarativePoliciesReport
	ec2: CancelExportTask
	ec2: CancelImageLaunchPermission
	ec2: CancelImportTask
	ec2: CancelReservedInstancesListing
	ec2: CancelSpotFleetRequests
	ec2: CancelSpotInstanceRequests
	ec2: ConfirmProductInstance
	ec2: CopyFpgaImage
	ec2: CopyImage
	ec2: CopySnapshot
	ec2: CopyVolumes

Prefisso del servizio	Azioni
	ec2: CreateCapacityManagerDataExport
	ec2: CreateCapacityReservation
	ec2: CreateCapacityReservationBySplitting
	ec2: CreateCapacityReservationCancellationQuote
	ec2: CreateCapacityReservationFleet
	ec2: CreateCarrierGateway
	ec2: CreateClientVpnEndpoint
	ec2: CreateClientVpnRoute
	ec2: CreateCoipCidr
	ec2: CreateCoipPool
	ec2: CreateCustomerGateway
	ec2: CreateDefaultSubnet
	ec2: CreateDefaultVpc
	ec2: CreateDelegateMacVolumeOwnershipTask
	ec2: CreateDhcpOptions
	ec2: CreateEgressOnlyInternetGateway
	ec2: CreateFleet
	ec2: CreateFlowLogs
	ec2: CreateFpgaImage
	ec2: CreateImage
	ec2: CreateImageUsageReport

Prefisso del servizio	Azioni
	ec2: CreateInstanceConnectEndpoint
	ec2: CreateInstanceEventWindow
	ec2: CreateInstanceExportTask
	ec2: CreateInternetGateway
	ec2: CreateInterruptibleCapacityReservationAllocation
	ec2: CreateIpam
	ec2: CreateIpamExternalResourceVerificationToken
	ec2: CreateIpamPolicy
	ec2: CreateIpamPool
	ec2: CreateIpamPrefixListResolver
	ec2: CreateIpamPrefixListResolverTarget
	ec2: CreateIpamResourceDiscovery
	ec2: CreateIpamScope
	ec2: CreateKeyPair
	ec2: CreateLaunchTemplateVersion
	ec2: CreateLocalGatewayRoute
	ec2: CreateLocalGatewayRouteTable
	ec2: CreateLocalGatewayRouteTableVirtualInterfaceGroupAssociation
	ec2: CreateLocalGatewayRouteTableVpcAssociation
	ec2: CreateLocalGatewayVirtualInterface

Prefisso del servizio	Azioni
	ec2: CreateLocalGatewayVirtualInterfaceGroup
	ec2: CreateMacSystemIntegrityProtectionModificationTask
	ec2: CreateManagedPrefixList
	ec2: CreateNatGateway
	ec2: CreateNetworkAcl
	ec2: CreateNetworkAclEntry
	ec2: CreateNetworkInsightsAccessScope
	ec2: CreateNetworkInsightsPath
	ec2: CreateNetworkInterface
	ec2: CreateNetworkInterfacePermission
	ec2: CreatePlacementGroup
	ec2: CreatePublicIpv4Pool
	ec2: CreateReplaceRootVolumeTask
	ec2: CreateReservedInstancesListing
	ec2: CreateRestoreImageTask
	ec2: CreateRoute
	ec2: CreateRouteServer
	ec2: CreateRouteServerEndpoint
	ec2: CreateRouteServerPeer
	ec2: CreateRouteTable
	ec2: CreateSecondaryNetwork

Prefisso del servizio	Azioni
	ec2: CreateSecondarySubnet
	ec2: CreateSecurityGroup
	ec2: CreateSnapshots
	ec2: CreateSpotDatafeedSubscription
	ec2: CreateStoreImageTask
	ec2: CreateSubnet
	ec2: CreateSubnetCidrReservation
	ec2: CreateTrafficMirrorFilter
	ec2: CreateTrafficMirrorFilterRule
	ec2: CreateTrafficMirrorSession
	ec2: CreateTrafficMirrorTarget
	ec2: CreateTransitGateway
	ec2: CreateTransitGatewayConnect
	ec2: CreateTransitGatewayConnectPeer
	ec2: CreateTransitGatewayMeteringPolicy
	ec2: CreateTransitGatewayMeteringPolicyEntry
	ec2: CreateTransitGatewayMulticastDomain
	ec2: CreateTransitGatewayPeeringAttachment
	ec2: CreateTransitGatewayPolicyTable
	ec2: CreateTransitGatewayPrefixListReference
	ec2: CreateTransitGatewayRoute

Prefisso del servizio	Azioni
	ec2: CreateTransitGatewayRouteTable
	ec2: CreateTransitGatewayRouteTableAnnouncement
	ec2: CreateTransitGatewayVpcAttachment
	ec2: CreateVerifiedAccessEndpoint
	ec2: CreateVerifiedAccessGroup
	ec2: CreateVerifiedAccessInstance
	ec2: CreateVerifiedAccessTrustProvider
	ec2: CreateVolume
	ec2: CreateVpc
	ec2: CreateVpcBlockPublicAccessExclusion
	ec2: CreateVpcEncryptionControl
	ec2: CreateVpcEndpoint
	ec2: CreateVpcEndpointConnectionNotification
	ec2: CreateVpcEndpointServiceConfiguration
	ec2: CreateVpcPeeringConnection
	ec2: CreateVpnConcentrator
	ec2: CreateVpnConnection
	ec2: CreateVpnConnectionRoute
	ec2: CreateVpnGateway
	ec2: DeleteCapacityManagerDataExport
	ec2: DeleteCarrierGateway

Prefisso del servizio	Azioni
	ec2: DeleteClientVpnEndpoint
	ec2: DeleteClientVpnRoute
	ec2: DeleteCoipCidr
	ec2: DeleteCoipPool
	ec2: DeleteCustomerGateway
	ec2: DeleteDhcpOptions
	ec2: DeleteEgressOnlyInternetGateway
	ec2: DeleteFleets
	ec2: DeleteFlowLogs
	ec2: DeleteFpgaImage
	ec2: DeleteImageUsageReport
	ec2: DeleteInstanceConnectEndpoint
	ec2: DeleteInstanceEventWindow
	ec2: DeleteInternetGateway
	ec2: DeleteIpam
	ec2: DeleteIpamExternalResourceVerificationToken
	ec2: DeleteIpamPolicy
	ec2: DeleteIpamPool
	ec2: DeleteIpamPrefixListResolver
	ec2: DeleteIpamPrefixListResolverTarget
	ec2: DeleteIpamResourceDiscovery

Prefisso del servizio	Azioni
	ec2: DeleteIpamScope
	ec2: DeleteKeyPair
	ec2: DeleteLaunchTemplate
	ec2: DeleteLaunchTemplateVersions
	ec2: DeleteLocalGatewayRoute
	ec2: DeleteLocalGatewayRouteTable
	ec2: DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssoc iation
	ec2: DeleteLocalGatewayRouteTableVpcAssociation
	ec2: DeleteLocalGatewayVirtualInterface
	ec2: DeleteLocalGatewayVirtualInterfaceGroup
	ec2: DeleteManagedPrefixList
	ec2: DeleteNatGateway
	ec2: DeleteNetworkAcl
	ec2: DeleteNetworkAclEntry
	ec2: DeleteNetworkInsightsAccessScope
	ec2: DeleteNetworkInsightsAccessScopeAnalysis
	ec2: DeleteNetworkInsightsAnalysis
	ec2: DeleteNetworkInsightsPath
	ec2: DeleteNetworkInterface
	ec2: DeleteNetworkInterfacePermission

Prefisso del servizio	Azioni
	ec2: DeletePlacementGroup
	ec2: DeletePublicIpv4Pool
	ec2: DeleteQueuedReservedInstances
	ec2: DeleteRoute
	ec2: DeleteRouteServer
	ec2: DeleteRouteServerEndpoint
	ec2: DeleteRouteServerPeer
	ec2: DeleteRouteTable
	ec2: DeleteSecondaryNetwork
	ec2: DeleteSecondarySubnet
	ec2: DeleteSecurityGroup
	ec2: DeleteSpotDatafeedSubscription
	ec2: DeleteSubnet
	ec2: DeleteSubnetCidrReservation
	ec2: DeleteTrafficMirrorFilter
	ec2: DeleteTrafficMirrorFilterRule
	ec2: DeleteTrafficMirrorSession
	ec2: DeleteTrafficMirrorTarget
	ec2: DeleteTransitGateway
	ec2: DeleteTransitGatewayClientVpnAttachment
	ec2: DeleteTransitGatewayConnect

Prefisso del servizio	Azioni
	ec2: DeleteTransitGatewayConnectPeer
	ec2: DeleteTransitGatewayMeteringPolicy
	ec2: DeleteTransitGatewayMeteringPolicyEntry
	ec2: DeleteTransitGatewayMulticastDomain
	ec2: DeleteTransitGatewayPeeringAttachment
	ec2: DeleteTransitGatewayPolicyTable
	ec2: DeleteTransitGatewayPrefixListReference
	ec2: DeleteTransitGatewayRoute
	ec2: DeleteTransitGatewayRouteTable
	ec2: DeleteTransitGatewayRouteTableAnnouncement
	ec2: DeleteTransitGatewayVpcAttachment
	ec2: DeleteVerifiedAccessEndpoint
	ec2: DeleteVerifiedAccessGroup
	ec2: DeleteVerifiedAccessInstance
	ec2: DeleteVerifiedAccessTrustProvider
	ec2: DeleteVolume
	ec2: DeleteVpc
	ec2: DeleteVpcBlockPublicAccessExclusion
	ec2: DeleteVpcEncryptionControl
	ec2: DeleteVpcEndpointConnectionNotifications
	ec2: DeleteVpcEndpointServiceConfigurations

Prefisso del servizio	Azioni
	ec2: DeleteVpcEndpoints
	ec2: DeleteVpcPeeringConnection
	ec2: DeleteVpnConcentrator
	ec2: DeleteVpnConnection
	ec2: DeleteVpnConnectionRoute
	ec2: DeleteVpnGateway
	ec2: DeprovisionByoipCidr
	ec2: DeprovisionIpamByoasn
	ec2: DeprovisionIpamPoolCidr
	ec2: DeprovisionPublicIpv4PoolCidr
	ec2: DeregisterImage
	ec2: DeregisterInstanceEventNotificationAttributes
	ec2: DeregisterTransitGatewayMulticastGroupMembers
	ec2: DeregisterTransitGatewayMulticastGroupSources
	ec2: DescribeAccountAttributes
	ec2: DescribeAddressTransfers
	ec2: DescribeAddresses
	ec2: DescribeAddressesAttribute
	ec2: DescribeAggregateIdFormat
	ec2: DescribeAvailabilityZones
	ec2: DescribeAwsNetworkPerformanceMetricSubscriptions

Prefisso del servizio	Azioni
	ec2: DescribeBundleTasks
	ec2: DescribeByoipCidrs
	ec2: DescribeCapacityBlockExtensionHistory
	ec2: DescribeCapacityBlockExtensionOfferings
	ec2: DescribeCapacityBlockStatus
	ec2: DescribeCapacityBlocks
	ec2: DescribeCapacityManagerDataExports
	ec2: DescribeCapacityReservationBillingRequests
	ec2: DescribeCapacityReservationCancellationQuotes
	ec2: DescribeCapacityReservationFleets
	ec2: DescribeCapacityReservationTopology
	ec2: DescribeCapacityReservations
	ec2: DescribeCarrierGateways
	ec2: DescribeClassicLinkInstances
	ec2: DescribeClientVpnAuthorizationRules
	ec2: DescribeClientVpnConnections
	ec2: DescribeClientVpnEndpoints
	ec2: DescribeClientVpnRoutes
	ec2: DescribeClientVpnTargetNetworks
	ec2: DescribeCoipPools
	ec2: DescribeConversionTasks

Prefisso del servizio	Azioni
	ec2: DescribeCustomerGateways
	ec2: DescribeDeclarativePoliciesReports
	ec2: DescribeDhcpOptions
	ec2: DescribeEgressOnlyInternetGateways
	ec2: DescribeElasticGpus
	ec2: DescribeExportImageTasks
	ec2: DescribeExportTasks
	ec2: DescribeFastLaunchImages
	ec2: DescribeFastSnapshotRestores
	ec2: DescribeFleetHistory
	ec2: DescribeFleetInstances
	ec2: DescribeFleets
	ec2: DescribeFlowLogs
	ec2: DescribeFpgaImageAttribute
	ec2: DescribeFpgaImages
	ec2: DescribeHostReservationOfferings
	ec2: DescribeHostReservations
	ec2: DescribeHosts
	ec2: DescribeIamInstanceProfileAssociations
	ec2: DescribeIdFormat
	ec2: DescribeIdentityIdFormat

Prefisso del servizio	Azioni
	ec2: DescribeImageAttribute
	ec2: DescribeImageReferences
	ec2: DescribeImageUsageReportEntries
	ec2: DescribeImageUsageReports
	ec2: DescribeImportImageTasks
	ec2: DescribeImportSnapshotTasks
	ec2: DescribeInstanceConnectEndpoints
	ec2: DescribeInstanceCreditSpecifications
	ec2: DescribeInstanceEventNotificationAttributes
	ec2: DescribeInstanceEventWindows
	ec2: DescribeInstanceImageMetadata
	ec2: DescribeInstanceSqlHaHistoryStates
	ec2: DescribeInstanceSqlHaStates
	ec2: DescribeInstanceTopology
	ec2: DescribeInstanceTypes
	ec2: DescribeInternetGateways
	ec2: DescribeIpamByoasn
	ec2: DescribeIpamExternalResourceVerificationTokens
	ec2: DescribeIpamPolicies
	ec2: DescribeIpamPoolAllocations
	ec2: DescribeIpamPools

Prefisso del servizio	Azioni
	ec2: DescribeIpamPrefixListResolverTargets
	ec2: DescribeIpamPrefixListResolvers
	ec2: DescribeIpamResourceDiscoveries
	ec2: DescribeIpamResourceDiscoveryAssociations
	ec2: DescribeIpamScopes
	ec2: DescribeIpams
	ec2: DescribeIpv6Pools
	ec2: DescribeKeyPairs
	ec2: DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations
	ec2: DescribeLocalGatewayRouteTableVpcAssociations
	ec2: DescribeLocalGatewayRouteTables
	ec2: DescribeLocalGatewayVirtualInterfaceGroups
	ec2: DescribeLocalGatewayVirtualInterfaces
	ec2: DescribeLocalGateways
	ec2: DescribeLockedSnapshots
	ec2: DescribeMacHosts
	ec2: DescribeMacModificationTasks
	ec2: DescribeManagedPrefixLists
	ec2: DescribeMovingAddresses
	ec2: DescribeNatGateways

Prefisso del servizio	Azioni
	ec2: DescribeNetworkAcls
	ec2: DescribeNetworkInsightsAccessScopeAnalyses
	ec2: DescribeNetworkInsightsAccessScopes
	ec2: DescribeNetworkInsightsAnalyses
	ec2: DescribeNetworkInsightsPaths
	ec2: DescribeNetworkInterfaceAttribute
	ec2: DescribeNetworkInterfacePermissions
	ec2: DescribeNetworkInterfaces
	ec2: DescribeOutpostLags
	ec2: DescribePlacementGroups
	ec2: DescribePrefixLists
	ec2: DescribePrincipalIdFormat
	ec2: DescribePublicIpv4Pools
	ec2: DescribeRegions
	ec2: DescribeReplaceRootVolumeTasks
	ec2: DescribeReservedInstances
	ec2: DescribeReservedInstancesListings
	ec2: DescribeReservedInstancesModifications
	ec2: DescribeReservedInstancesOfferings
	ec2: DescribeRouteServerEndpoints
	ec2: DescribeRouteServerPeers

Prefisso del servizio	Azioni
	ec2: DescribeRouteServers
	ec2: DescribeRouteTables
	ec2: DescribeScheduledInstanceAvailability
	ec2: DescribeScheduledInstances
	ec2: DescribeSecondaryInterfaces
	ec2: DescribeSecondaryNetworks
	ec2: DescribeSecondarySubnets
	ec2: DescribeSecurityGroupReferences
	ec2: DescribeSecurityGroupRules
	ec2: DescribeSecurityGroupVpcAssociations
	ec2: DescribeSecurityGroups
	ec2: DescribeServiceLinkVirtualInterfaces
	ec2: DescribeSnapshotAttribute
	ec2: DescribeSnapshotTierStatus
	ec2: DescribeSpotDatafeedSubscription
	ec2: DescribeSpotFleetInstances
	ec2: DescribeSpotFleetRequestHistory
	ec2: DescribeSpotFleetRequests
	ec2: DescribeSpotInstanceRequests
	ec2: DescribeSpotPriceHistory
	ec2: DescribeStaleSecurityGroups

Prefisso del servizio	Azioni
	ec2: DescribeStoreImageTasks
	ec2: DescribeTrafficMirrorFilterRules
	ec2: DescribeTrafficMirrorFilters
	ec2: DescribeTrafficMirrorSessions
	ec2: DescribeTrafficMirrorTargets
	ec2: DescribeTransitGatewayAttachments
	ec2: DescribeTransitGatewayConnectPeers
	ec2: DescribeTransitGatewayConnects
	ec2: DescribeTransitGatewayMeteringPolicies
	ec2: DescribeTransitGatewayMulticastDomains
	ec2: DescribeTransitGatewayPeeringAttachments
	ec2: DescribeTransitGatewayPolicyTables
	ec2: DescribeTransitGatewayRouteTableAnnouncements
	ec2: DescribeTransitGatewayRouteTables
	ec2: DescribeTransitGatewayVpcAttachments
	ec2: DescribeTransitGateways
	ec2: DescribeTrunkInterfaceAssociations
	ec2: DescribeVerifiedAccessEndpoints
	ec2: DescribeVerifiedAccessGroups
	ec2: DescribeVerifiedAccessInstanceLoggingConfigurations
	ec2: DescribeVerifiedAccessInstances

Prefisso del servizio	Azioni
	ec2: DescribeVerifiedAccessTrustProviders
	ec2: DescribeVolumeAttribute
	ec2: DescribeVolumeStatus
	ec2: DescribeVolumes
	ec2: DescribeVolumesModifications
	ec2: DescribeVpcAttribute
	ec2: DescribeVpcBlockPublicAccessExclusions
	ec2: DescribeVpcBlockPublicAccessOptions
	ec2: DescribeVpcClassicLink
	ec2: DescribeVpcClassicLinkDnsSupport
	ec2: DescribeVpcEncryptionControls
	ec2: DescribeVpcEndpointAssociations
	ec2: DescribeVpcEndpointConnectionNotifications
	ec2: DescribeVpcEndpointConnections
	ec2: DescribeVpcEndpointServiceConfigurations
	ec2: DescribeVpcEndpointServicePermissions
	ec2: DescribeVpcEndpointServices
	ec2: DescribeVpcEndpoints
	ec2: DescribeVpcPeeringConnections
	ec2: DescribeVpcs
	ec2: DescribeVpnConcentrators

Prefisso del servizio	Azioni
	ec2: DescribeVpnConnections
	ec2: DescribeVpnGateways
	ec2: DetachClassicLinkVpc
	ec2: DetachImageWatermark
	ec2: DetachInternetGateway
	ec2: DetachNetworkInterface
	ec2: DetachVerifiedAccessTrustProvider
	ec2: DetachVolume
	ec2: DetachVpnGateway
	ec2: DisableAddressTransfer
	ec2: DisableAllowedImagesSettings
	ec2: DisableAwsNetworkPerformanceMetricSubscription
	ec2: DisableCapacityManager
	ec2: DisableEbsEncryptionByDefault
	ec2: DisableFastLaunch
	ec2: DisableFastSnapshotRestores
	ec2: DisableImage
	ec2: DisableImageBlockPublicAccess
	ec2: DisableImageDeprecation
	ec2: DisableImageDeregistrationProtection
	ec2: DisableInstanceSqlHaStandbyDetections

Prefisso del servizio	Azioni
	ec2: DisableIpamOrganizationAdminAccount
	ec2: DisableIpamPolicy
	ec2: DisableRouteServerPropagation
	ec2: DisableSerialConsoleAccess
	ec2: DisableSnapshotBlockPublicAccess
	ec2: DisableTransitGatewayRouteTablePropagation
	ec2: DisableVgwRoutePropagation
	ec2: DisableVpcClassicLink
	ec2: DisableVpcClassicLinkDnsSupport
	ec2: DisassociateAddress
	ec2: DisassociateCapacityReservationBillingOwner
	ec2: DisassociateClientVpnTargetNetwork
	ec2: DisassociateEnclaveCertificateIamRole
	ec2: DisassociateIamInstanceProfile
	ec2: DisassociateInstanceEventWindow
	ec2: DisassociateIpamByoasn
	ec2: DisassociateIpamResourceDiscovery
	ec2: DisassociateNatGatewayAddress
	ec2: DisassociateRouteServer
	ec2: DisassociateRouteTable
	ec2: DisassociateSecurityGroupVpc

Prefisso del servizio	Azioni
	ec2: DisassociateSubnetCidrBlock
	ec2: DisassociateTransitGatewayMulticastDomain
	ec2: DisassociateTransitGatewayPolicyTable
	ec2: DisassociateTransitGatewayRouteTable
	ec2: DisassociateTrunkInterface
	ec2: DisassociateVpcCidrBlock
	ec2: EnableAddressTransfer
	ec2: EnableAllowedImagesSettings
	ec2: EnableAwsNetworkPerformanceMetricSubscription
	ec2: EnableCapacityManager
	ec2: EnableEbsEncryptionByDefault
	ec2: EnableFastLaunch
	ec2: EnableFastSnapshotRestores
	ec2: EnableImage
	ec2: EnableImageBlockPublicAccess
	ec2: EnableImageDeprecation
	ec2: EnableImageDeregistrationProtection
	ec2: EnableInstanceSqlHaStandbyDetections
	ec2: EnableIpamOrganizationAdminAccount
	ec2: EnableIpamPolicy
	ec2: EnableReachabilityAnalyzerOrganizationSharing

Prefisso del servizio	Azioni
	ec2: EnableRouteServerPropagation
	ec2: EnableSerialConsoleAccess
	ec2: EnableSnapshotBlockPublicAccess
	ec2: EnableTransitGatewayRouteTablePropagation
	ec2: EnableVgwRoutePropagation
	ec2: IO EnableVolume
	ec2: EnableVpcClassicLink
	ec2: EnableVpcClassicLinkDnsSupport
	ec2: ExportClientVpnClientCertificateRevocationList
	ec2: ExportClientVpnClientConfiguration
	ec2: ExportImage
	ec2: ExportTransitGatewayRoutes
	ec2: ExportVerifiedAccessInstanceClientConfiguration
	ec2: GetActiveVpnTunnelStatus
	ec2: GetAllowedImagesSettings
	ec2: GetAssociatedEnclaveCertificateIamRoles
	ec2: GetAssociatedIpv6PoolCidrs
	ec2: GetAwsNetworkPerformanceData
	ec2: GetCapacityManagerAttributes
	ec2: GetCapacityManagerMetricData
	ec2: GetCapacityManagerMetricDimensions

Prefisso del servizio	Azioni
	ec2: GetCapacityReservationUsage
	ec2: GetCoipPoolUsage
	ec2: GetConsoleOutput
	ec2: GetConsoleScreenshot
	ec2: GetDeclarativePoliciesReportSummary
	ec2: GetDefaultCreditSpecification
	ec2: GetEbsDefaultKmsKeyId
	ec2: GetEbsEncryptionByDefault
	ec2: GetEnabledIpamPolicy
	ec2: GetFlowLogsIntegrationTemplate
	ec2: GetGroupsForCapacityReservation
	ec2: GetHostReservationPurchasePreview
	ec2: GetImageAncestry
	ec2: GetImageBlockPublicAccessState
	ec2: GetInstanceMetadataDefaults
	ec2: GetInstanceTpmEkPub
	ec2: GetInstanceTypesFromInstanceRequirements
	ec2: GetInstanceUefiData
	ec2: GetIpamAddressHistory
	ec2: GetIpamDiscoveredAccounts
	ec2: GetIpamDiscoveredPublicAddresses

Prefisso del servizio	Azioni
	ec2: GetIpamDiscoveredResourceCidrs
	ec2: GetIpamPolicyAllocationRules
	ec2: GetIpamPolicyOrganizationTargets
	ec2: GetIpamPoolAllocations
	ec2: GetIpamPoolCidrs
	ec2: GetIpamPrefixListResolverRules
	ec2: GetIpamPrefixListResolverVersionEntries
	ec2: GetIpamPrefixListResolverVersions
	ec2: GetIpamResourceCidrs
	ec2: GetLaunchTemplateData
	ec2: GetManagedPrefixListAssociations
	ec2: GetManagedPrefixListEntries
	ec2: GetManagedResourceVisibility
	ec2: GetNetworkInsightsAccessScopeAnalysisFindings
	ec2: GetNetworkInsightsAccessScopeContent
	ec2: GetPasswordData
	ec2: GetReservedInstancesExchangeQuote
	ec2: GetRouteServerAssociations
	ec2: GetRouteServerPropagations
	ec2: GetRouteServerRoutingDatabase
	ec2: GetSecurityGroupsForVpc

Prefisso del servizio	Azioni
	ec2: GetSerialConsoleAccessStatus
	ec2: GetSnapshotBlockPublicAccessState
	ec2: GetSpotPlacementScores
	ec2: GetSubnetCidrReservations
	ec2: GetTransitGatewayAttachmentPropagations
	ec2: GetTransitGatewayMeteringPolicyEntries
	ec2: GetTransitGatewayMulticastDomainAssociations
	ec2: GetTransitGatewayPolicyTableAssociations
	ec2: GetTransitGatewayPolicyTableEntries
	ec2: GetTransitGatewayPrefixListReferences
	ec2: GetTransitGatewayRouteTableAssociations
	ec2: GetTransitGatewayRouteTablePropagations
	ec2: GetVerifiedAccessEndpointPolicy
	ec2: GetVerifiedAccessEndpointTargets
	ec2: GetVerifiedAccessGroupPolicy
	ec2: GetVpcResourcesBlockingEncryptionEnforcement
	ec2: GetVpnConnectionDeviceSampleConfiguration
	ec2: GetVpnConnectionDeviceTypes
	ec2: GetVpnTunnelReplacementStatus
	ec2: ImportClientVpnClientCertificateRevocationList
	ec2: ImportImage

Prefisso del servizio	Azioni
	ec2: ImportInstance
	ec2: ImportKeyPair
	ec2: ImportSnapshot
	ec2: ImportVolume
	ec2: ListImagesInRecycleBin
	ec2: ListSnapshotsInRecycleBin
	ec2: ListVolumesInRecycleBin
	ec2: LockSnapshot
	ec2: ModifyAddressAttribute
	ec2: ModifyAvailabilityZoneGroup
	ec2: ModifyCapacityReservation
	ec2: ModifyCapacityReservationFleet
	ec2: ModifyClientVpnEndpoint
	ec2: ModifyDefaultCreditSpecification
	ec2: ModifyEbsDefaultKmsKeyId
	ec2: ModifyFleet
	ec2: ModifyFpgaImageAttribute
	ec2: ModifyHosts
	ec2: ModifyIdFormat
	ec2: ModifyIdentityIdFormat
	ec2: ModifyImageAttribute

Prefisso del servizio	Azioni
	ec2: ModifyInstanceAttribute
	ec2: ModifyInstanceCapacityReservationAttributes
	ec2: ModifyInstanceConnectEndpoint
	ec2: ModifyInstanceCpuOptions
	ec2: ModifyInstanceCreditSpecification
	ec2: ModifyInstanceEventStartTime
	ec2: ModifyInstanceEventWindow
	ec2: ModifyInstanceMaintenanceOptions
	ec2: ModifyInstanceMetadataDefaults
	ec2: ModifyInstanceMetadataOptions
	ec2: ModifyInstanceNetworkPerformanceOptions
	ec2: ModifyInstancePlacement
	ec2: ModifyIpam
	ec2: ModifyIpamPolicyAllocationRules
	ec2: ModifyIpamPool
	ec2: ModifyIpamPoolAllocation
	ec2: ModifyIpamPrefixListResolver
	ec2: ModifyIpamPrefixListResolverTarget
	ec2: ModifyIpamResourceCidr
	ec2: ModifyIpamResourceDiscovery
	ec2: ModifyIpamScope

Prefisso del servizio	Azioni
	ec2: ModifyLaunchTemplate
	ec2: ModifyLocalGatewayRoute
	ec2: ModifyManagedPrefixList
	ec2: ModifyManagedResourceVisibility
	ec2: ModifyNetworkInterfaceAttribute
	ec2: ModifyPrivateDnsNameOptions
	ec2: ModifyPublicIpDnsNameOptions
	ec2: ModifyReservedInstances
	ec2: ModifyRouteServer
	ec2: ModifySecurityGroupRules
	ec2: ModifySnapshotAttribute
	ec2: ModifySnapshotTier
	ec2: ModifySpotFleetRequest
	ec2: ModifySubnetAttribute
	ec2: ModifyTrafficMirrorFilterNetworkServices
	ec2: ModifyTrafficMirrorFilterRule
	ec2: ModifyTrafficMirrorSession
	ec2: ModifyTransitGateway
	ec2: ModifyTransitGatewayMeteringPolicy
	ec2: ModifyTransitGatewayPrefixListReference
	ec2: ModifyTransitGatewayVpcAttachment

Prefisso del servizio	Azioni
	ec2: ModifyVerifiedAccessEndpoint
	ec2: ModifyVerifiedAccessEndpointPolicy
	ec2: ModifyVerifiedAccessGroup
	ec2: ModifyVerifiedAccessGroupPolicy
	ec2: ModifyVerifiedAccessInstance
	ec2: ModifyVerifiedAccessInstanceLoggingConfiguration
	ec2: ModifyVerifiedAccessTrustProvider
	ec2: ModifyVolume
	ec2: ModifyVolumeAttribute
	ec2: ModifyVpcAttribute
	ec2: ModifyVpcBlockPublicAccessExclusion
	ec2: ModifyVpcBlockPublicAccessOptions
	ec2: ModifyVpcEncryptionControl
	ec2: ModifyVpcEndpoint
	ec2: ModifyVpcEndpointConnectionNotification
	ec2: ModifyVpcEndpointServiceConfiguration
	ec2: ModifyVpcEndpointServicePayerResponsibility
	ec2: ModifyVpcEndpointServicePermissions
	ec2: ModifyVpcPeeringConnectionOptions
	ec2: ModifyVpcTenancy
	ec2: ModifyVpnConnection

Prefisso del servizio	Azioni
	ec2: ModifyVpnConnectionOptions
	ec2: ModifyVpnTunnelCertificate
	ec2: ModifyVpnTunnelOptions
	ec2: MonitorInstances
	ec2: MoveAddressToVpc
	ec2: MoveByoipCidrToIpam
	ec2: MoveCapacityReservationInstances
	ec2: ProvisionByoipCidr
	ec2: ProvisionIpamByoasn
	ec2: ProvisionIpamPoolCidr
	ec2: ProvisionPublicIpv4PoolCidr
	ec2: PurchaseCapacityBlockExtension
	ec2: PurchaseHostReservation
	ec2: PurchaseReservedInstancesOffering
	ec2: PurchaseScheduledInstances
	ec2: RebootInstances
	ec2: RegisterImage
	ec2: RegisterInstanceEventNotificationAttributes
	ec2: RegisterTransitGatewayMulticastGroupMembers
	ec2: RegisterTransitGatewayMulticastGroupSources
	ec2: RejectCapacityReservationBillingOwnership

Prefisso del servizio	Azioni
	ec2: RejectTransitGatewayClientVpnAttachment
	ec2: RejectTransitGatewayMulticastDomainAssociations
	ec2: RejectTransitGatewayPeeringAttachment
	ec2: RejectTransitGatewayVpcAttachment
	ec2: RejectVpcEndpointConnections
	ec2: RejectVpcPeeringConnection
	ec2: ReleaseAddress
	ec2: ReleaseHosts
	ec2: ReleaseIpamPoolAllocation
	ec2: ReplaceIamInstanceProfileAssociation
	ec2: ReplaceImageCriteriaInAllowedImagesSettings
	ec2: ReplaceNetworkAclAssociation
	ec2: ReplaceNetworkAclEntry
	ec2: ReplaceRoute
	ec2: ReplaceRouteTableAssociation
	ec2: ReplaceTransitGatewayRoute
	ec2: ReplaceVpnTunnel
	ec2: ReportInstanceStatus
	ec2: RequestSpotFleet
	ec2: RequestSpotInstances
	ec2: ResetAddressAttribute

Prefisso del servizio	Azioni
	ec2: ResetEbsDefaultKmsKeyId
	ec2: ResetFpgaImageAttribute
	ec2: ResetImageAttribute
	ec2: ResetInstanceAttribute
	ec2: ResetNetworkInterfaceAttribute
	ec2: ResetSnapshotAttribute
	ec2: RestoreAddressToClassic
	ec2: RestoreImageFromRecycleBin
	ec2: RestoreManagedPrefixListVersion
	ec2: RestoreSnapshotFromRecycleBin
	ec2: RestoreSnapshotTier
	ec2: RestoreVolumeFromRecycleBin
	ec2: RevokeClientVpnIngress
	ec2: RevokeSecurityGroupEgress
	ec2: RevokeSecurityGroupIngress
	ec2: RunInstances
	ec2: RunScheduledInstances
	ec2: SearchLocalGatewayRoutes
	ec2: SearchTransitGatewayMulticastGroups
	ec2: SearchTransitGatewayRoutes
	ec2: SendDiagnosticInterrupt

Prefisso del servizio	Azioni
	ec2: StartDeclarativePoliciesReport
	ec2: StartInstances
	ec2: StartNetworkInsightsAccessScopeAnalysis
	ec2: StartNetworkInsightsAnalysis
	ec2: StartVpcEndpointServicePrivateDnsVerification
	ec2: TerminateClientVpnConnections
	ec2: UnassignIpv6Addresses
	ec2: UnassignPrivateIpAddresses
	ec2: UnassignPrivateNatGatewayAddress
	ec2: UnlockSnapshot
	ec2: UnmonitorInstances
	ec2: UpdateCapacityManagerOrganizationsAccess
	ec2: UpdateInterruptibleCapacityReservationAllocation
	ec2: UpdateSecurityGroupRuleDescriptionsEgress
	ec2: UpdateSecurityGroupRuleDescriptionsIngress
	ec2: WithdrawByoipCidr

Prefisso del servizio	Azioni
ecr	ecr: BatchCheckLayerAvailability ecr: BatchDeleteImage ecr: BatchGetImage ecr: BatchGetRepositoryScanningConfiguration ecr: CompleteLayerUpload ecr: CreatePullThroughCacheRule ecr: CreateRepositoryCreationTemplate ecr: DeleteLifecyclePolicy ecr: DeletePullThroughCacheRule ecr: DeleteRegistryPolicy ecr: DeleteRepository ecr: DeleteRepositoryCreationTemplate ecr: DeleteRepositoryPolicy ecr: DeleteSigningConfiguration ecr: DescribeImageReplicationStatus ecr: DescribeImageScanFindings ecr: DescribeImages ecr: DescribePullThroughCacheRules ecr: DescribeRegistry ecr: DescribeRepositories ecr: DescribeRepositoryCreationTemplates

Prefisso del servizio	Azioni
	ecr: GetAccountSetting ecr: GetAuthorizationToken ecr: GetDownloadUrlForLayer ecr: GetLifecyclePolicy ecr: GetLifecyclePolicyPreview ecr: GetRegistryPolicy ecr: GetRegistryScanningConfiguration ecr: GetRepositoryPolicy ecr: GetSigningConfiguration ecr: InitiateLayerUpload ecr: ListImages ecr: ListPullTimeUpdateExclusions ecr: PutAccountSetting ecr: PutImage ecr: PutImageScanningConfiguration ecr: PutRegistryPolicy ecr: PutRegistryScanningConfiguration ecr: PutReplicationConfiguration ecr: StartImageScan ecr: StartLifecyclePolicyPreview ecr: UpdatePullThroughCacheRule

Prefisso del servizio	Azioni
	ecr: UpdateRepositoryCreationTemplate ecr: UploadLayerPart ecr: ValidatePullThroughCacheRule

Prefisso del servizio	Azioni
ecr-public	ecr-pubblico: BatchCheckLayerAvailability
	ecr-pubblico: BatchDeleteImage
	ecr-pubblico: CompleteLayerUpload
	ecr-pubblico: CreateRepository
	ecr-pubblico: DeleteRepository
	ecr-pubblico: DeleteRepositoryPolicy
	ecr-pubblico: DescribeImages
	ecr-pubblico: DescribeRegistries
	ecr-pubblico: DescribeRepositories
	ecr-pubblico: GetAuthorizationToken
	ecr-pubblico: GetRegistryCatalogData
	ecr-pubblico: GetRepositoryCatalogData
	ecr-pubblico: GetRepositoryPolicy
	ecr-pubblico: InitiateLayerUpload
	ecr-pubblico: PutImage
	ecr-pubblico: PutRegistryCatalogData
	ecr-pubblico: PutRepositoryCatalogData
	ecr-pubblico: SetRepositoryPolicy
	ecr-pubblico: UploadLayerPart

Prefisso del servizio	Azioni
ecs	ecs: ContinueServiceDeployment ecs: CreateCapacityProvider ecs: CreateCluster ecs: CreateDaemon ecs: CreateService ecs: CreateTaskSet ecs: DeleteAccountSetting ecs: DeleteAttributes ecs: DeleteCapacityProvider ecs: DeleteCluster ecs: DeleteDaemon ecs: DeleteDaemonTaskDefinition ecs: DeleteExpressGatewayService ecs: DeleteService ecs: DeleteTaskDefinitions ecs: DeleteTaskSet ecs: DeregisterContainerInstance ecs: DeregisterTaskDefinition ecs: DescribeCapacityProviders ecs: DescribeClusters ecs: DescribeContainerInstances

Prefisso del servizio	Azioni
	ecs: DescribeDaemon ecs: DescribeDaemonDeployments ecs: DescribeDaemonRevisions ecs: DescribeDaemonTaskDefinition ecs: DescribeExpressGatewayService ecs: DescribeServiceDeployments ecs: DescribeServiceRevisions ecs: DescribeServices ecs: DescribeTaskDefinition ecs: DescribeTaskSets ecs: DescribeTasks ecs: DiscoverPollEndpoint ecs: ExecuteCommand ecs: GetTaskProtection ecs: ListAccountSettings ecs: ListAttributes ecs: ListClusters ecs: ListContainerInstances ecs: ListDaemonDeployments ecs: ListDaemonTaskDefinitions ecs: ListDaemons

Prefisso del servizio	Azioni
	ecs: ListServiceDeployments ecs: ListServices ecs: ListServicesByNamespace ecs: ListTaskDefinitionFamilies ecs: ListTaskDefinitions ecs: ListTasks ecs: PutAccountSetting ecs: PutAccountSettingDefault ecs: PutAttributes ecs: PutClusterCapacityProviders ecs: RegisterContainerInstance ecs: RegisterDaemonTaskDefinition ecs: RunTask ecs: StartTask ecs: StopServiceDeployment ecs: StopTask ecs: SubmitAttachmentStateChanges ecs: SubmitContainerStateChange ecs: SubmitTaskStateChange ecs: UpdateCapacityProvider ecs: UpdateCluster

Prefisso del servizio	Azioni
	ecs: UpdateClusterSettings ecs: UpdateContainerAgent ecs: UpdateContainerInstancesState ecs: UpdateDaemon ecs: UpdateExpressGatewayService ecs: UpdateService ecs: UpdateServicePrimaryTaskSet ecs: UpdateTaskProtection ecs: UpdateTaskSet

Prefisso del servizio	Azioni
eks	ex: AssociateAccessPolicy
	ex: AssociateEncryptionConfig
	ex: AssociateIdentityProviderConfig
	ex: CancelUpdate
	ex: CreateAccessEntry
	ex: CreateAddon
	ex: CreateCluster
	ex: CreateEksAnywhereSubscription
	ex: CreateFargateProfile
	ex: CreateNodegroup
	ex: DeleteAccessEntry
	ex: DeleteAddon
	ex: DeleteCapability
	ex: DeleteCluster
	ex: DeleteEksAnywhereSubscription
	ex: DeleteFargateProfile
	ex: DeleteNodegroup
	ex: DeletePodIdentityAssociation
	ex: DeregisterCluster
	ex: DescribeAccessEntry
	ex: DescribeAddon

Prefisso del servizio	Azioni
	ex: DescribeAddonConfiguration
	ex: DescribeAddonVersions
	ex: DescribeCapability
	ex: DescribeCluster
	ex: DescribeClusterVersions
	ex: DescribeEksAnywhereSubscription
	ex: DescribeFargateProfile
	ex: DescribeIdentityProviderConfig
	ex: DescribeInsight
	ex: DescribeInsightsRefresh
	ex: DescribeNodegroup
	ex: DescribePodIdentityAssociation
	ex: DescribeUpdate
	ex: DisassociateAccessPolicy
	ex: DisassociateIdentityProviderConfig
	ex: ListAccessEntries
	ex: ListAccessPolicies
	ex: ListAddons
	ex: ListAssociatedAccessPolicies
	ex: ListCapabilities
	ex: ListClusters

Prefisso del servizio	Azioni
	ex: ListEksAnywhereSubscriptions ex: ListFargateProfiles ex: ListIdentityProviderConfigs ex: ListInsights ex: ListNodegroups ex: ListPodIdentityAssociations ex: ListUpdates ex: RegisterCluster ex: StartInsightsRefresh ex: UpdateAccessEntry ex: UpdateAddon ex: UpdateCapability ex: UpdateClusterConfig ex: UpdateClusterVersion ex: UpdateEksAnywhereSubscription ex: UpdateNodegroupConfig ex: UpdateNodegroupVersion ex: UpdatePodIdentityAssociation

Prefisso del servizio	Azioni
elasticache	dolore elastico: AuthorizeCacheSecurityGroupIngress dolore elastico: BatchApplyUpdateAction dolore elastico: BatchStopUpdateAction dolore elastico: CompleteMigration dolore elastico: CopyServerlessCacheSnapshot dolore elastico: CopySnapshot dolore elastico: CreateCacheCluster dolore elastico: CreateCacheParameterGroup dolore elastico: CreateCacheSecurityGroup dolore elastico: CreateCacheSubnetGroup dolore elastico: CreateGlobalReplicationGroup dolore elastico: CreateReplicationGroup dolore elastico: CreateServerlessCache dolore elastico: CreateServerlessCacheSnapshot dolore elastico: CreateSnapshot dolore elastico: CreateUser dolore elastico: CreateUserGroup dolore elastico: DecreaseNodeGroupsInGlobalReplicationGroup dolore elastico: DecreaseReplicaCount dolore elastico: DeleteCacheCluster dolore elastico: DeleteCacheParameterGroup

Prefisso del servizio	Azioni
	dolore elastico: DeleteCacheSecurityGroup dolore elastico: DeleteCacheSubnetGroup dolore elastico: DeleteGlobalReplicationGroup dolore elastico: DeleteReplicationGroup dolore elastico: DeleteServerlessCache dolore elastico: DeleteServerlessCacheSnapshot dolore elastico: DeleteSnapshot dolore elastico: DeleteUser dolore elastico: DeleteUserGroup dolore elastico: DescribeCacheClusters dolore elastico: DescribeCacheEngineVersions dolore elastico: DescribeCacheParameterGroups dolore elastico: DescribeCacheParameters dolore elastico: DescribeCacheSecurityGroups dolore elastico: DescribeCacheSubnetGroups dolore elastico: DescribeEngineDefaultParameters dolore elastico: DescribeEvents dolore elastico: DescribeGlobalReplicationGroups dolore elastico: DescribeReplicationGroups dolore elastico: DescribeReservedCacheNodes dolore elastico: DescribeReservedCacheNodesOfferings

Prefisso del servizio	Azioni
	dolore elastico: DescribeServerlessCacheSnapshots dolore elastico: DescribeServerlessCaches dolore elastico: DescribeServiceUpdates dolore elastico: DescribeSnapshots dolore elastico: DescribeUpdateActions dolore elastico: DescribeUserGroups dolore elastico: DescribeUsers dolore elastico: DisassociateGlobalReplicationGroup dolore elastico: ExportServerlessCacheSnapshot dolore elastico: FailoverGlobalReplicationGroup dolore elastico: IncreaseNodeGroupsInGlobalReplicationGroup dolore elastico: IncreaseReplicaCount dolore elastico: ListAllowedNodeTypeModifications dolore elastico: ModifyCacheCluster dolore elastico: ModifyCacheParameterGroup dolore elastico: ModifyCacheSubnetGroup dolore elastico: ModifyGlobalReplicationGroup dolore elastico: ModifyReplicationGroup dolore elastico: ModifyReplicationGroupShardConfiguration dolore elastico: ModifyServerlessCache dolore elastico: ModifyUser

Prefisso del servizio	Azioni
	dolore elastico: ModifyUserGroup dolore elastico: PurchaseReservedCacheNodesOffering dolore elastico: RebalanceSlotsInGlobalReplicationGroup dolore elastico: RebootCacheCluster dolore elastico: ResetCacheParameterGroup dolore elastico: RevokeCacheSecurityGroupIngress dolore elastico: StartMigration dolore elastico: TestFailover dolore elastico: TestMigration

Prefisso del servizio	Azioni
elasticbeanstalk	gambo elastico di fagioli: AbortEnvironmentUpdate
	gambo elastico di fagioli: ApplyEnvironmentManagedAction
	gambo elastico di fagioli: AssociateEnvironmentOperationsRole
	elasticbeanstalk:CheckDNSAvailability
	gambo elastico di fagioli: ComposeEnvironments
	gambo elastico di fagioli: CreateApplication
	gambo elastico di fagioli: CreateApplicationVersion
	gambo elastico di fagioli: CreateConfigurationTemplate
	gambo elastico di fagioli: CreateEnvironment
	gambo elastico di fagioli: CreatePlatformVersion
	gambo elastico di fagioli: CreateStorageLocation
	gambo elastico di fagioli: DeleteApplication
	gambo elastico di fagioli: DeleteApplicationVersion
	gambo elastico di fagioli: DeleteConfigurationTemplate
	gambo elastico di fagioli: DeleteEnvironmentConfiguration
	gambo elastico di fagioli: DeletePlatformVersion
	gambo elastico di fagioli: DescribeAccountAttributes
	gambo elastico di fagioli: DescribeApplicationVersions
	gambo elastico di fagioli: DescribeApplications
	gambo elastico di fagioli: DescribeConfigurationOptions
	gambo elastico di fagioli: DescribeConfigurationSettings

Prefisso del servizio	Azioni
	gambo elastico di fagioli: DescribeEnvironmentHealth
	gambo elastico di fagioli: DescribeEnvironmentManagedActionHistory
	gambo elastico di fagioli: DescribeEnvironmentManagedActions
	gambo elastico di fagioli: DescribeEnvironmentResources
	gambo elastico di fagioli: DescribeEnvironments
	gambo elastico di fagioli: DescribeEvents
	gambo elastico di fagioli: DescribeInstancesHealth
	gambo elastico di fagioli: DescribePlatformVersion
	gambo elastico di fagioli: DisassociateEnvironmentOperationsRole
	gambo elastico di fagioli: ListAvailableSolutionStacks
	gambo elastico di fagioli: ListPlatformBranches
	gambo elastico di fagioli: ListPlatformVersions
	gambo elastico di fagioli: RebuildEnvironment
	gambo elastico di fagioli: RequestEnvironmentInfo
	gambo elastico di fagioli: RestartAppServer
	gambo elastico di fagioli: RetrieveEnvironmentInfo
	gambo elastico di fagioli: SwapEnvironmentCNAMEs
	gambo elastico di fagioli: TerminateEnvironment
	gambo elastico di fagioli: UpdateApplication
	gambo elastico di fagioli: UpdateApplicationResourceLifecycle

Prefisso del servizio	Azioni
	gambo elastico di fagioli: UpdateApplicationVersion
	gambo elastico di fagioli: UpdateConfigurationTemplate
	gambo elastico di fagioli: UpdateEnvironment
	gambo elastico di fagioli: ValidateConfigurationSettings

Prefisso del servizio	Azioni
elasticfilesystem	file system elastico: CreateAccessPoint filesystem elastico: CreateFileSystem filesystem elastico: CreateMountTarget filesystem elastico: CreateReplicationConfiguration filesystem elastico: DeleteAccessPoint filesystem elastico: DeleteFileSystem filesystem elastico: DeleteFileSystemPolicy filesystem elastico: DeleteMountTarget filesystem elastico: DeleteReplicationConfiguration filesystem elastico: DescribeAccessPoints filesystem elastico: DescribeAccountPreferences filesystem elastico: DescribeBackupPolicy filesystem elastico: DescribeFileSystemPolicy filesystem elastico: DescribeFileSystems filesystem elastico: DescribeLifecycleConfiguration filesystem elastico: DescribeMountTargetSecurityGroups filesystem elastico: DescribeMountTargets filesystem elastico: DescribeReplicationConfigurations filesystem elastico: ModifyMountTargetSecurityGroups filesystem elastico: PutAccountPreferences filesystem elastico: PutBackupPolicy

Prefisso del servizio	Azioni
	filesystem elastico: PutFileSystemPolicy filesystem elastico: PutLifecycleConfiguration filesystem elastico: UpdateFileSystem filesystem elastico: UpdateFileSystemProtection

Prefisso del servizio	Azioni
elasticloadbalancing	bilanciamento elastico del carico: AddListenerCertificates
	bilanciamento elastico del carico: AddTrustStoreRevocations
	bilanciamento elastico del carico: ApplySecurityGroupsToLoadBalancer
	bilanciamento elastico del carico: AttachLoadBalancerToSubnets
	bilanciamento elastico del carico: ConfigureHealthCheck
	bilanciamento elastico del carico: CreateAppCookieStickinessPolicy
	bilanciamento elastico del carico: CreateLBCookieStickinessPolicy
	bilanciamento elastico del carico: CreateListener
	bilanciamento elastico del carico: CreateLoadBalancer
	bilanciamento elastico del carico: CreateLoadBalancerListeners
	bilanciamento elastico del carico: CreateLoadBalancerPolicy
	bilanciamento elastico del carico: CreateRule
	bilanciamento elastico del carico: CreateTargetGroup
	bilanciamento elastico del carico: CreateTrustStore
	bilanciamento elastico del carico: DeleteListener
	bilanciamento elastico del carico: DeleteLoadBalancer
	bilanciamento elastico del carico: DeleteLoadBalancerListeners
	bilanciamento elastico del carico: DeleteLoadBalancerPolicy
	bilanciamento elastico del carico: DeleteRule
	bilanciamento elastico del carico: DeleteSharedTrustStoreAssociation

Prefisso del servizio	Azioni
	bilanciamento elastico del carico: DeleteTargetGroup
	bilanciamento elastico del carico: DeleteTrustStore
	bilanciamento elastico del carico: DeregisterInstancesFromLoad Balancer
	bilanciamento elastico del carico: DeregisterTargets
	bilanciamento elastico del carico: DescribeAccountLimits
	bilanciamento elastico del carico: DescribeCapacityReservation
	bilanciamento elastico del carico: DescribeInstanceHealth
	bilanciamento elastico del carico: DescribeListenerAttributes
	bilanciamento elastico del carico: DescribeListenerCertificates
	bilanciamento elastico del carico: DescribeListeners
	bilanciamento elastico del carico: DescribeLoadBalancerAttributes
	bilanciamento elastico del carico: DescribeLoadBalancerPolicies
	bilanciamento elastico del carico: DescribeLoadBalancerPolicyT ypes
	bilanciamento elastico del carico: DescribeLoadBalancers
	bilanciamento elastico del carico: DescribeRules
	elasticloadbalancing:DescribeSSLPolicies
	bilanciamento elastico del carico: DescribeTargetGroupAttributes
	bilanciamento elastico del carico: DescribeTargetGroups
	bilanciamento elastico del carico: DescribeTargetHealth
	bilanciamento elastico del carico: DescribeTrustStoreAssociations

Prefisso del servizio	Azioni
	bilanciamento elastico del carico: DescribeTrustStoreRevocations
	bilanciamento elastico del carico: DescribeTrustStores
	bilanciamento elastico del carico: DetachLoadBalancerFromSubnets
	bilanciamento elastico del carico: DisableAvailabilityZonesForLoadBalancer
	bilanciamento elastico del carico: EnableAvailabilityZonesForLoadBalancer
	bilanciamento elastico del carico: GetResourcePolicy
	bilanciamento elastico del carico: GetTrustStoreCaCertificatesBundle
	bilanciamento elastico del carico: GetTrustStoreRevocationContent
	bilanciamento elastico del carico: ModifyCapacityReservation
	bilanciamento elastico del carico: ModifyIpPools
	bilanciamento elastico del carico: ModifyListener
	bilanciamento elastico del carico: ModifyLoadBalancerAttributes
	bilanciamento elastico del carico: ModifyRule
	bilanciamento elastico del carico: ModifyTargetGroup
	bilanciamento elastico del carico: ModifyTargetGroupAttributes
	bilanciamento elastico del carico: ModifyTrustStore
	bilanciamento elastico del carico: RegisterInstancesWithLoadBalancer
	bilanciamento elastico del carico: RegisterTargets

Prefisso del servizio	Azioni
	bilanciamento elastico del carico: RemoveListenerCertificates
	bilanciamento elastico del carico: RemoveTrustStoreRevocations
	bilanciamento elastico del carico: SetIpAddressType
	bilanciamento elastico del carico: SetLoadBalancerListenerSSLCertificate
	bilanciamento elastico del carico: SetLoadBalancerPoliciesForBackendServer
	bilanciamento elastico del carico: SetLoadBalancerPoliciesOfListener
	bilanciamento elastico del carico: SetRulePriorities
	bilanciamento elastico del carico: SetSecurityGroups
	bilanciamento elastico del carico: SetSubnets

Prefisso del servizio	Azioni
emr-containers	contenitori emr: CancelJobRun contenitori emr: CreateJobTemplate contenitori emr: CreateManagedEndpoint contenitori emr: CreateSecurityConfiguration contenitori emr: CreateVirtualCluster contenitori emr: DeleteJobTemplate contenitori emr: DeleteManagedEndpoint contenitori emr: DeleteSecurityConfiguration contenitori emr: DeleteVirtualCluster contenitori emr: DescribeJobRun contenitori emr: DescribeJobTemplate contenitori emr: DescribeManagedEndpoint contenitori emr: DescribeSecurityConfiguration contenitori emr: DescribeVirtualCluster contenitori emr: GetManagedEndpointSessionCredentials contenitori emr: ListJobRuns contenitori emr: ListJobTemplates contenitori emr: ListManagedEndpoints contenitori emr: ListSecurityConfigurations contenitori emr: ListVirtualClusters contenitori emr: StartJobRun

Prefisso del servizio	Azioni
emr-serverless	emr-senza server: CancelJobRun
	emr-senza server: CreateApplication
	emr-senza server: DeleteApplication
	emr-senza server: GetApplication
	emr-senza server: GetDashboardForJobRun
	emr-senza server: GetJobRun
	emr-senza server: GetResourceDashboard
	emr-senza server: GetSession
	emr-senza server: GetSessionEndpoint
	emr-senza server: ListApplications
	emr-senza server: ListJobRunAttempts
	emr-senza server: ListJobRuns
	emr-senza server: ListSessions
	emr-senza server: StartApplication
	emr-senza server: StartJobRun
	emr-senza server: StopApplication
	emr-senza server: TerminateSession
	emr-senza server: UpdateApplication

Prefisso del servizio	Azioni
es	Sì: AcceptInboundConnection Sì: AcceptInboundCrossClusterSearchConnection Sì: AssociatePackage Sì: AuthorizeVpcEndpointAccess Sì: CancelElasticsearchServiceSoftwareUpdate Sì: CancelServiceSoftwareUpdate Sì: CreateDomain Sì: CreateElasticsearchDomain Sì: CreateIndex Sì: CreateOutboundConnection Sì: CreateOutboundCrossClusterSearchConnection Sì: CreatePackage Sì: CreateVpcEndpoint Sì: DeleteDomain Sì: DeleteElasticsearchDomain Sì: DeleteElasticsearchServiceRole Sì: DeleteInboundConnection Sì: DeleteInboundCrossClusterSearchConnection Sì: DeleteIndex Sì: DeleteOutboundConnection Sì: DeleteOutboundCrossClusterSearchConnection

Prefisso del servizio	Azioni
	Sì: DeletePackage Sì: DeleteVpcEndpoint Sì: DescribeDomain Sì: DescribeDomainAutoTunes Sì: DescribeDomainChangeProgress Sì: DescribeDomainConfig Sì: DescribeDomainHealth Sì: DescribeDomainNodes Sì: DescribeDomains Sì: DescribeDryRunProgress Sì: DescribeElasticsearchDomain Sì: DescribeElasticsearchDomainConfig Sì: DescribeElasticsearchDomains Sì: DescribeElasticsearchInstanceTypeLimits Sì: DescribeInboundConnections Sì: DescribeInboundCrossClusterSearchConnections Sì: DescribeInstanceTypeLimits Sì: DescribeOutboundConnections Sì: DescribeOutboundCrossClusterSearchConnections Sì: DescribePackages Sì: DescribeReservedElasticsearchInstanceOfferings

Prefisso del servizio	Azioni
	Sì: DescribeReservedElasticsearchInstances Sì: DescribeReservedInstanceOfferings Sì: DescribeReservedInstances Sì: DescribeVpcEndpoints Sì: DissociatePackage Sì: DissociatePackages Sì: GetCompatibleElasticsearchVersions Sì: GetCompatibleVersions Sì: GetDataSource Sì: GetDomainMaintenanceStatus Sì: GetPackageVersionHistory Sì: GetUpgradeHistory Sì: GetUpgradeStatus Sì: ListDataSources Sì: ListDomainNames Sì: ListDomainsForPackage Sì: ListElasticsearchInstanceTypes Sì: ListElasticsearchVersions Sì: ListInsights Sì: ListInstanceTypeDetails Sì: ListPackagesForDomain

Prefisso del servizio	Azioni
	Sì: ListScheduledActions Sì: ListVersions Sì: ListVpcEndpointAccess Sì: ListVpcEndpoints Sì: ListVpcEndpointsForDomain Sì: PurchaseReservedElasticsearchInstanceOffering Sì: PurchaseReservedInstanceOffering Sì: RejectInboundConnection Sì: RejectInboundCrossClusterSearchConnection Sì: RevokeVpcEndpointAccess Sì: RollbackServiceSoftwareUpdate Sì: StartDomainMaintenance Sì: StartElasticsearchServiceSoftwareUpdate Sì: StartServiceSoftwareUpdate Sì: UpdateDataSource Sì: UpdateDomainConfig Sì: UpdateElasticsearchDomainConfig Sì: UpdateIndex Sì: UpdatePackage Sì: UpdatePackageScope Sì: UpdateScheduledAction

Prefisso del servizio	Azioni
	Sì: UpdateVpcEndpoint Sì: UpgradeDomain Sì: UpgradeElasticsearchDomain

Prefisso del servizio	Azioni
eventi	eventi: ActivateEventSource
	eventi: CancelReplay
	eventi: CreateApiDestination
	eventi: CreateArchive
	eventi: CreateConnection
	eventi: CreateEndpoint
	eventi: CreateEventBus
	eventi: CreatePartnerEventSource
	eventi: DeactivateEventSource
	eventi: DeauthorizeConnection
	eventi: DeleteApiDestination
	eventi: DeleteArchive
	eventi: DeleteConnection
	eventi: DeleteEndpoint
	eventi: DeleteEventBus
	eventi: DeletePartnerEventSource
	eventi: DeleteRule
	eventi: DescribeApiDestination
	eventi: DescribeArchive
	eventi: DescribeConnection
	eventi: DescribeEndpoint

Prefisso del servizio	Azioni
	eventi: DescribeEventBus
	eventi: DescribeEventSource
	eventi: DescribePartnerEventSource
	eventi: DescribeReplay
	eventi: DescribeRule
	eventi: DisableRule
	eventi: EnableRule
	eventi: ListApiDestinations
	eventi: ListArchives
	eventi: ListConnections
	eventi: ListEndpoints
	eventi: ListEventBuses
	eventi: ListEventSources
	eventi: ListPartnerEventSourceAccounts
	eventi: ListPartnerEventSources
	eventi: ListReplays
	eventi: ListRuleNamesByTarget
	eventi: ListRules
	eventi: ListTargetsByRule
	eventi: PutPermission
	eventi: PutRule

Prefisso del servizio	Azioni
	eventi: PutTargets
	eventi: RemovePermission
	eventi: RemoveTargets
	eventi: StartReplay
	eventi: TestEventPattern
	eventi: UpdateApiDestination
	eventi: UpdateArchive
	eventi: UpdateConnection
	eventi: UpdateEndpoint
	eventi: UpdateEventBus

Prefisso del servizio	Azioni
finspace	finspace: CreateEnvironment spazio interno: CreateKxChangeset spazio interno: CreateKxCluster spazio interno: CreateKxDatabase spazio interno: CreateKxDataview spazio interno: CreateKxEnvironment spazio interno: CreateKxScalingGroup spazio interno: CreateKxUser spazio interno: CreateKxVolume spazio interno: CreateUser spazio interno: DeleteEnvironment spazio interno: DeleteKxCluster spazio interno: DeleteKxClusterNode spazio interno: DeleteKxDatabase spazio interno: DeleteKxDataview spazio interno: DeleteKxEnvironment spazio interno: DeleteKxScalingGroup spazio interno: DeleteKxUser spazio interno: DeleteKxVolume spazio interno: GetEnvironment spazio interno: GetKxChangeset

Prefisso del servizio	Azioni
	spazio interno: GetKxCluster spazio interno: GetKxConnectionString spazio interno: GetKxDatabase spazio interno: GetKxDataview spazio interno: GetKxEnvironment spazio interno: GetKxScalingGroup spazio interno: GetKxUser spazio interno: GetKxVolume spazio interno: GetLoadSampleDataSetGroupIntoEnvironmentStatus spazio interno: GetUser spazio interno: ListEnvironments spazio interno: ListKxChangesets spazio interno: ListKxClusterNodes spazio interno: ListKxClusters spazio interno: ListKxDatabases spazio interno: ListKxDataviews spazio interno: ListKxEnvironments spazio interno: ListKxScalingGroups spazio interno: ListKxUsers spazio interno: ListKxVolumes

Prefisso del servizio	Azioni
	spazio interno: ListUsers spazio interno: LoadSampleDataSetGroupIntoEnvironment spazio interno: ResetUserPassword spazio interno: UpdateEnvironment spazio interno: UpdateKxClusterCodeConfiguration spazio interno: UpdateKxClusterDatabases spazio interno: UpdateKxDatabase spazio interno: UpdateKxDataview spazio interno: UpdateKxEnvironment spazio interno: UpdateKxEnvironmentNetwork spazio interno: UpdateKxUser spazio interno: UpdateKxVolume spazio interno: UpdateUser
firehose	manichetta antincendio: CreateDeliveryStream manichetta antincendio: DeleteDeliveryStream manichetta antincendio: DescribeDeliveryStream manichetta antincendio: ListDeliveryStreams manichetta antincendio: StartDeliveryStreamEncryption manichetta antincendio: StopDeliveryStreamEncryption manichetta antincendio: UpdateDestination

Prefisso del servizio	Azioni
fis	pinza: CreateExperimentTemplate pesce: CreateTargetAccountConfiguration pesce: DeleteExperimentTemplate pesce: DeleteTargetAccountConfiguration pesce: GetAction pesce: GetExperiment pesce: GetExperimentTargetAccountConfiguration pesce: GetExperimentTemplate pesce: GetSafetyLever pesce: GetTargetAccountConfiguration pesce: GetTargetResourceType pesce: ListActions pesce: ListExperimentResolvedTargets pesce: ListExperimentTargetAccountConfigurations pesce: ListExperimentTemplates pesce: ListExperiments pesce: ListTargetAccountConfigurations pesce: ListTargetResourceTypes pesce: StartExperiment pesce: StopExperiment pesce: UpdateExperimentTemplate

Prefisso del servizio	Azioni
	pesce: UpdateSafetyLeverState pesce: UpdateTargetAccountConfiguration

Prefisso del servizio	Azioni
fms	fms: AssociateAdminAccount
	fms: AssociateThirdPartyFirewall
	fms: BatchAssociateResource
	fms: BatchDisassociateResource
	fms: DeleteAppsList
	fms: DeleteNotificationChannel
	fms: DeletePolicy
	fms: DeleteProtocolsList
	fms: DeleteResourceSet
	fms: DisassociateAdminAccount
	fms: DisassociateThirdPartyFirewall
	fms: GetAdminAccount
	fms: GetAdminScope
	fms: GetAppsList
	fms: GetComplianceDetail
	fms: GetNotificationChannel
	fms: GetPolicy
	fms: GetProtectionStatus
	fms: GetProtocolsList
	fms: GetResourceSet
	fms: GetThirdPartyFirewallAssociationStatus

Prefisso del servizio	Azioni
	fms: GetViolationDetails
	fms: ListAdminAccountsForOrganization
	fms: ListAdminsManagingAccount
	fms: ListAppsLists
	fms: ListComplianceStatus
	fms: ListDiscoveredResources
	fms: ListMemberAccounts
	fms: ListPolicies
	fms: ListProtocolsLists
	fms: ListResourceSetResources
	fms: ListResourceSets
	fms: ListThirdPartyFirewallFirewallPolicies
	fms: PutAdminAccount
	fms: PutAppsList
	fms: PutNotificationChannel
	fms: PutPolicy
	fms: PutProtocolsList
	fms: PutResourceSet

Prefisso del servizio	Azioni
frauddetector	rilevatore di frodi: BatchCreateVariable rilevatore di frodi: BatchGetVariable rilevatore di frodi: CancelBatchImportJob rilevatore di frodi: CancelBatchPredictionJob rilevatore di frodi: CreateBatchImportJob rilevatore di frodi: CreateBatchPredictionJob rilevatore di frodi: CreateDetectorVersion rilevatore di frodi: CreateList rilevatore di frodi: CreateModel rilevatore di frodi: CreateModelVersion rilevatore di frodi: CreateRule rilevatore di frodi: CreateVariable rilevatore di frodi: DeleteBatchImportJob rilevatore di frodi: DeleteBatchPredictionJob rilevatore di frodi: DeleteDetector rilevatore di frodi: DeleteDetectorVersion rilevatore di frodi: DeleteEntityType rilevatore di frodi: DeleteEvent rilevatore di frodi: DeleteEventType rilevatore di frodi: DeleteEventsByEventType rilevatore di frodi: DeleteExternalModel

Prefisso del servizio	Azioni
	rilevatore di frodi: DeleteLabel rilevatore di frodi: DeleteList rilevatore di frodi: DeleteModel rilevatore di frodi: DeleteModelVersion rilevatore di frodi: DeleteOutcome rilevatore di frodi: DeleteRule rilevatore di frodi: DeleteVariable rilevatore di frodi: DescribeDetector rilevatore di frodi: DescribeModelVersions rilevatore di frodi: GetBatchImportJobs rilevatore di frodi: GetBatchPredictionJobs rilevatore di frodi: GetDeleteEventsByEventTypeStatus rilevatore di frodi: GetDetectorVersion rilevatore di frodi: GetDetectors rilevatore di frodi: GetEntityTypes rilevatore di frodi: GetEvent rilevatore di frodi: GetEventPrediction rilevatore di frodi: GetEventPredictionMetadata rilevatore di frodi: GetEventTypes rilevatore di frodi: GetExternalModels rilevatore di frodi: GetKMSEncryptionKey

Prefisso del servizio	Azioni
	rilevatore di frodi: GetLabels rilevatore di frodi: GetListElements rilevatore di frodi: GetListsMetadata rilevatore di frodi: GetModelVersion rilevatore di frodi: GetModels rilevatore di frodi: GetOutcomes rilevatore di frodi: GetRules rilevatore di frodi: GetVariables rilevatore di frodi: ListEventPredictions rilevatore di frodi: PutDetector rilevatore di frodi: PutEntityType rilevatore di frodi: PutEventType rilevatore di frodi: PutExternalModel rilevatore di frodi: PutKMSEncryptionKey rilevatore di frodi: PutLabel rilevatore di frodi: PutOutcome rilevatore di frodi: SendEvent rilevatore di frodi: UpdateDetectorVersion rilevatore di frodi: UpdateDetectorVersionMetadata rilevatore di frodi: UpdateDetectorVersionStatus rilevatore di frodi: UpdateEventLabel

Prefisso del servizio	Azioni
	rilevatore di frodi: UpdateList rilevatore di frodi: UpdateModel rilevatore di frodi: UpdateModelVersion rilevatore di frodi: UpdateModelVersionStatus rilevatore di frodi: UpdateRuleMetadata rilevatore di frodi: UpdateRuleVersion rilevatore di frodi: UpdateVariable

Prefisso del servizio	Azioni
fsx	fax: AssociateFileSystemAliases
	fax: CancelDataRepositoryTask
	fax: CopyBackup
	fax: CreateDataRepositoryTask
	fax: CreateFileCache
	fax: CreateFileSystem
	fax: CreateFileSystemFromBackup
	fax: CreateSnapshot
	fax: CreateStorageVirtualMachine
	fax: CreateVolume
	fax: CreateVolumeFromBackup
	fax: DeleteBackup
	fax: DeleteFileCache
	fax: DeleteFileSystem
	fax: DeleteSnapshot
	fax: DeleteStorageVirtualMachine
	fax: DeleteVolume
	fax: DescribeBackups
	fax: DescribeDataRepositoryAssociations
	fax: DescribeDataRepositoryTasks
	fax: DescribeFileCaches

Prefisso del servizio	Azioni
	fax: DescribeFileSystemAliases
	fax: DescribeFileSystems
	fax: DescribeS3AccessPointAttachments
	fax: DescribeSharedVpcConfiguration
	fax: DescribeSnapshots
	fax: DescribeStorageVirtualMachines
	fax: DescribeVolumes
	fax: DetachAndDeleteS3AccessPoint
	fax: DisassociateFileSystemAliases
	fax: ReleaseFileSystemNfsV3Locks
	fax: RestoreVolumeFromSnapshot
	fax: StartMisconfiguredStateRecovery
	fax: UpdateDataRepositoryAssociation
	fax: UpdateFileCache
	fax: UpdateFileSystem
	fax: UpdateSharedVpcConfiguration
	fax: UpdateSnapshot
	fax: UpdateStorageVirtualMachine
	fax: UpdateVolume

Prefisso del servizio	Azioni
gamelift	rinnovamento del gioco: AcceptMatch rinnovamento del gioco: ClaimGameServer rinnovamento del gioco: CreateAlias rinnovamento del gioco: CreateBuild rinnovamento del gioco: CreateContainerGroupDefinition rinnovamento del gioco: CreateFleet rinnovamento del gioco: CreateFleetLocations rinnovamento del gioco: CreateGameServerGroup rinnovamento del gioco: CreateGameSession rinnovamento del gioco: CreateGameSessionQueue rinnovamento del gioco: CreateLocation rinnovamento del gioco: CreateMatchmakingConfiguration rinnovamento del gioco: CreateMatchmakingRuleSet rinnovamento del gioco: CreatePlayerSession rinnovamento del gioco: CreatePlayerSessions rinnovamento del gioco: CreateScript rinnovamento del gioco: CreateVpcPeeringAuthorization rinnovamento del gioco: CreateVpcPeeringConnection rinnovamento del gioco: DeleteAlias rinnovamento del gioco: DeleteBuild rinnovamento del gioco: DeleteContainerGroupDefinition

Prefisso del servizio	Azioni
	rinnovamento del gioco: DeleteFleet rinnovamento del gioco: DeleteFleetLocations rinnovamento del gioco: DeleteGameServerGroup rinnovamento del gioco: DeleteGameSessionQueue rinnovamento del gioco: DeleteLocation rinnovamento del gioco: DeleteMatchmakingConfiguration rinnovamento del gioco: DeleteMatchmakingRuleSet rinnovamento del gioco: DeleteScalingPolicy rinnovamento del gioco: DeleteScript rinnovamento del gioco: DeleteVpcPeeringAuthorization rinnovamento del gioco: DeleteVpcPeeringConnection rinnovamento del gioco: DeregisterCompute rinnovamento del gioco: DeregisterGameServer rinnovamento del gioco: DescribeAlias rinnovamento del gioco: DescribeBuild rinnovamento del gioco: DescribeCompute rinnovamento del gioco: DescribeContainerFleet rinnovamento del gioco: DescribeContainerGroupDefinition rinnovamento del gioco: DescribeContainerGroupPortMappings rinnovamento del gioco: DescribeEC2InstanceLimits rinnovamento del gioco: DescribeFleetAttributes

Prefisso del servizio	Azioni
	rinnovamento del gioco: DescribeFleetCapacity
	rinnovamento del gioco: DescribeFleetEvents
	rinnovamento del gioco: DescribeFleetLocationAttributes
	rinnovamento del gioco: DescribeFleetLocationCapacity
	rinnovamento del gioco: DescribeFleetLocationUtilization
	rinnovamento del gioco: DescribeFleetPortSettings
	rinnovamento del gioco: DescribeFleetUtilization
	rinnovamento del gioco: DescribeGameServer
	rinnovamento del gioco: DescribeGameServerGroup
	rinnovamento del gioco: DescribeGameServerInstances
	rinnovamento del gioco: DescribeGameSessionDetails
	rinnovamento del gioco: DescribeGameSessionPlacement
	rinnovamento del gioco: DescribeGameSessionQueues
	rinnovamento del gioco: DescribeGameSessions
	rinnovamento del gioco: DescribeInstances
	rinnovamento del gioco: DescribeMatchmaking
	rinnovamento del gioco: DescribeMatchmakingConfigurations
	rinnovamento del gioco: DescribeMatchmakingRuleSets
	rinnovamento del gioco: DescribePlayerSessions
	rinnovamento del gioco: DescribeRuntimeConfiguration
	rinnovamento del gioco: DescribeScalingPolicies

Prefisso del servizio	Azioni
	rinnovamento del gioco: DescribeScript rinnovamento del gioco: DescribeVpcPeeringAuthorizations rinnovamento del gioco: DescribeVpcPeeringConnections rinnovamento del gioco: GetComputeAccess rinnovamento del gioco: GetComputeAuthToken rinnovamento del gioco: GetGameSessionLogUrl rinnovamento del gioco: GetInstanceAccess rinnovamento del gioco: GetPlayerConnectionDetails rinnovamento del gioco: ListAliases rinnovamento del gioco: ListBuilds rinnovamento del gioco: ListCompute rinnovamento del gioco: ListContainerFleets rinnovamento del gioco: ListContainerGroupDefinitionVersions rinnovamento del gioco: ListContainerGroupDefinitions rinnovamento del gioco: ListFleetDeployments rinnovamento del gioco: ListFleets rinnovamento del gioco: ListGameServerGroups rinnovamento del gioco: ListGameServers rinnovamento del gioco: ListLocations rinnovamento del gioco: ListScripts rinnovamento del gioco: PutScalingPolicy

Prefisso del servizio	Azioni
	rinnovamento del gioco: RegisterCompute rinnovamento del gioco: RegisterGameServer rinnovamento del gioco: RequestUploadCredentials rinnovamento del gioco: ResolveAlias rinnovamento del gioco: ResumeGameServerGroup rinnovamento del gioco: SearchGameSessions rinnovamento del gioco: StartFleetActions rinnovamento del gioco: StartGameSessionPlacement rinnovamento del gioco: StartMatchBackfill rinnovamento del gioco: StartMatchmaking rinnovamento del gioco: StopFleetActions rinnovamento del gioco: StopGameSessionPlacement rinnovamento del gioco: StopMatchmaking rinnovamento del gioco: SuspendGameServerGroup rinnovamento del gioco: TerminateGameSession rinnovamento del gioco: UpdateAlias rinnovamento del gioco: UpdateBuild rinnovamento del gioco: UpdateContainerGroupDefinition rinnovamento del gioco: UpdateFleetAttributes rinnovamento del gioco: UpdateFleetCapacity rinnovamento del gioco: UpdateFleetPortSettings

Prefisso del servizio	Azioni
	rinnovamento del gioco: UpdateGameServer rinnovamento del gioco: UpdateGameServerGroup rinnovamento del gioco: UpdateGameSession rinnovamento del gioco: UpdateGameSessionQueue rinnovamento del gioco: UpdateMatchmakingConfiguration rinnovamento del gioco: UpdateRuntimeConfiguration rinnovamento del gioco: UpdateScript rinnovamento del gioco: ValidateMatchmakingRuleSet

Prefisso del servizio	Azioni
geo	geo: AssociateTrackerConsumer
	geo: BatchDeleteDevicePositionHistory
	geo: BatchDeleteGeofence
	geo: BatchEvaluateGeofences
	geo: BatchGetDevicePosition
	geo: BatchPutGeofence
	geo: BatchUpdateDevicePosition
	geo: CalculateRoute
	geo: CalculateRouteMatrix
	geo: CancelJob
	geo: CreateGeofenceCollection
	geo: CreateMap
	geo: CreatePlaceIndex
	geo: CreateRouteCalculator
	geo: CreateTracker
	geo: DeleteGeofenceCollection
	geo: DeleteKey
	geo: DeleteMap
	geo: DeletePlaceIndex
	geo: DeleteRouteCalculator
	geo: DeleteTracker

Prefisso del servizio	Azioni
	geo: DescribeGeofenceCollection
	geo: DescribeKey
	geo: DescribeMap
	geo: DescribePlaceIndex
	geo: DescribeRouteCalculator
	geo: DescribeTracker
	geo: DisassociateTrackerConsumer
	geo: ForecastGeofenceEvents
	geo: GetDevicePosition
	geo: GetDevicePositionHistory
	geo: GetGeofence
	geo: GetJob
	geo: GetMapGlyphs
	geo: GetMapSprites
	geo: GetMapStyleDescriptor
	geo: GetMapTile
	geo: GetPlace
	geo: ListDevicePositions
	geo: ListGeofenceCollections
	geo: ListGeofences
	geo: ListJobs

Prefisso del servizio	Azioni
	geo: ListKeys
	geo: ListMaps
	geo: ListPlaceIndexes
	geo: ListRouteCalculators
	geo: ListTrackerConsumers
	geo: ListTrackers
	geo: PutGeofence
	geo: SearchPlaceIndexForPosition
	geo: SearchPlaceIndexForSuggestions
	geo: SearchPlaceIndexForText
	geo: UpdateGeofenceCollection
	geo: UpdateKey
	geo: UpdateMap
	geo: UpdatePlaceIndex
	geo: UpdateRouteCalculator
	geo: UpdateTracker
	geo: VerifyDevicePosition

Prefisso del servizio	Azioni
glacier	ghiacciaio: AbortMultipartUpload ghiacciaio: AbortVaultLock ghiacciaio: CompleteMultipartUpload ghiacciaio: CompleteVaultLock ghiacciaio: CreateVault ghiacciaio: DeleteArchive ghiacciaio: DeleteVault ghiacciaio: DeleteVaultAccessPolicy ghiacciaio: DeleteVaultNotifications ghiacciaio: DescribeJob ghiacciaio: DescribeVault ghiacciaio: GetDataRetrievalPolicy ghiacciaio: GetJobOutput ghiacciaio: GetVaultAccessPolicy ghiacciaio: GetVaultLock ghiacciaio: GetVaultNotifications ghiacciaio: InitiateJob ghiacciaio: InitiateMultipartUpload ghiacciaio: InitiateVaultLock ghiacciaio: ListJobs ghiacciaio: ListMultipartUploads

Prefisso del servizio	Azioni
	ghiacciaio: ListParts ghiacciaio: ListProvisionedCapacity ghiacciaio: ListVaults ghiacciaio: PurchaseProvisionedCapacity ghiacciaio: SetDataRetrievalPolicy ghiacciaio: SetVaultAccessPolicy ghiacciaio: SetVaultNotifications ghiacciaio: UploadArchive ghiacciaio: UploadMultipartPart

Prefisso del servizio	Azioni
grafana	grafana: AssociateLicense
	grafana: CreateWorkspace
	grafana: CreateWorkspaceApiKey
	grafana: CreateWorkspaceServiceAccount
	grafana: CreateWorkspaceServiceAccountToken
	grafana: DeleteWorkspace
	grafana: DeleteWorkspaceApiKey
	grafana: DeleteWorkspaceServiceAccount
	grafana: DeleteWorkspaceServiceAccountToken
	grafana: DescribeWorkspace
	grafana: DescribeWorkspaceAuthentication
	grafana: DescribeWorkspaceConfiguration
	grafana: DisassociateLicense
	grafana: ListPermissions
	grafana: ListVersions
	grafana: ListWorkspaceServiceAccountTokens
	grafana: ListWorkspaceServiceAccounts
	grafana: ListWorkspaces
	grafana: UpdatePermissions
	grafana: UpdateWorkspace
	grafana: UpdateWorkspaceAuthentication

Prefisso del servizio	Azioni
	grafana: UpdateWorkspaceConfiguration

Prefisso del servizio	Azioni
greengrass	erba verde: AssociateRoleToGroup erba verde: AssociateServiceRoleToAccount erba verde: BatchAssociateClientDeviceWithCoreDevice erba verde: BatchDisassociateClientDeviceFromCoreDevice erba verde: CancelDeployment erba verde: CreateComponentVersion erba verde: CreateConnectorDefinition erba verde: CreateConnectorDefinitionVersion erba verde: CreateCoreDefinition erba verde: CreateCoreDefinitionVersion erba verde: CreateDeployment erba verde: CreateDeviceDefinition erba verde: CreateDeviceDefinitionVersion erba verde: CreateFunctionDefinition erba verde: CreateFunctionDefinitionVersion erba verde: CreateGroup erba verde: CreateGroupCertificateAuthority erba verde: CreateGroupVersion erba verde: CreateLoggerDefinition erba verde: CreateLoggerDefinitionVersion erba verde: CreateResourceDefinition

Prefisso del servizio	Azioni
	erba verde: CreateResourceDefinitionVersion erba verde: CreateSoftwareUpdateJob erba verde: CreateSubscriptionDefinition erba verde: CreateSubscriptionDefinitionVersion erba verde: DeleteComponent erba verde: DeleteConnectorDefinition erba verde: DeleteCoreDefinition erba verde: DeleteCoreDevice erba verde: DeleteDeployment erba verde: DeleteDeviceDefinition erba verde: DeleteFunctionDefinition erba verde: DeleteGroup erba verde: DeleteLoggerDefinition erba verde: DeleteResourceDefinition erba verde: DeleteSubscriptionDefinition erba verde: DescribeComponent erba verde: DisassociateRoleFromGroup erba verde: DisassociateServiceRoleFromAccount erba verde: GetAssociatedRole erba verde: GetBulkDeploymentStatus erba verde: GetComponent

Prefisso del servizio	Azioni
	erba verde: GetComponentVersionArtifact erba verde: GetConnectivityInfo erba verde: GetConnectorDefinition erba verde: GetConnectorDefinitionVersion erba verde: GetCoreDefinition erba verde: GetCoreDefinitionVersion erba verde: GetCoreDevice erba verde: GetDeployment erba verde: GetDeploymentStatus erba verde: GetDeviceDefinition erba verde: GetDeviceDefinitionVersion erba verde: GetFunctionDefinition erba verde: GetFunctionDefinitionVersion erba verde: GetGroup erba verde: GetGroupCertificateAuthority erba verde: GetGroupCertificateConfiguration erba verde: GetGroupVersion erba verde: GetLoggerDefinition erba verde: GetLoggerDefinitionVersion erba verde: GetResourceDefinition erba verde: GetResourceDefinitionVersion

Prefisso del servizio	Azioni
	erba verde: GetServiceRoleForAccount erba verde: GetSubscriptionDefinition erba verde: GetSubscriptionDefinitionVersion erba verde: GetThingRuntimeConfiguration erba verde: ListBulkDeploymentDetailedReports erba verde: ListBulkDeployments erba verde: ListClientDevicesAssociatedWithCoreDevice erba verde: ListComponentVersions erba verde: ListComponents erba verde: ListConnectorDefinitionVersions erba verde: ListConnectorDefinitions erba verde: ListCoreDefinitionVersions erba verde: ListCoreDefinitions erba verde: ListCoreDevices erba verde: ListDeployments erba verde: ListDeviceDefinitionVersions erba verde: ListDeviceDefinitions erba verde: ListEffectiveDeployments erba verde: ListFunctionDefinitionVersions erba verde: ListFunctionDefinitions erba verde: ListGroupCertificateAuthorities

Prefisso del servizio	Azioni
	erba verde: ListGroupVersions erba verde: ListGroups erba verde: ListInstalledComponents erba verde: ListLoggerDefinitionVersions erba verde: ListLoggerDefinitions erba verde: ListResourceDefinitionVersions erba verde: ListResourceDefinitions erba verde: ListSubscriptionDefinitionVersions erba verde: ListSubscriptionDefinitions erba verde: ResetDeployments erba verde: StartBulkDeployment erba verde: StopBulkDeployment erba verde: UpdateConnectivityInfo erba verde: UpdateConnectorDefinition erba verde: UpdateCoreDefinition erba verde: UpdateDeviceDefinition erba verde: UpdateFunctionDefinition erba verde: UpdateGroup erba verde: UpdateGroupCertificateConfiguration erba verde: UpdateLoggerDefinition erba verde: UpdateResourceDefinition

Prefisso del servizio	Azioni
	erba verde: UpdateSubscriptionDefinition erba verde: UpdateThingRuntimeConfiguration

Prefisso del servizio	Azioni
groundstation	stazione a terra: CancelContact
	stazione a terra: CreateConfig
	stazione a terra: CreateDataflowEndpointGroup
	stazione a terra: CreateDataflowEndpointGroupV2
	stazione a terra: CreateEphemeris
	stazione a terra: CreateMissionProfile
	stazione a terra: DeleteConfig
	stazione a terra: DeleteDataflowEndpointGroup
	stazione a terra: DeleteEphemeris
	stazione a terra: DeleteMissionProfile
	stazione a terra: DescribeContact
	stazione a terra: DescribeContactVersion
	stazione a terra: DescribeEphemeris
	stazione a terra: GetConfig
	stazione a terra: GetDataflowEndpointGroup
	stazione a terra: GetMinuteUsage
	stazione a terra: GetMissionProfile
	stazione a terra: GetSatellite
	stazione a terra: ListAntennas
	stazione a terra: ListConfigs
	stazione a terra: ListContacts

Prefisso del servizio	Azioni
	stazione a terra: ListDataflowEndpointGroups
	stazione a terra: ListEphemerides
	stazione a terra: ListGroundStationReservations
	stazione a terra: ListGroundStations
	stazione a terra: ListMissionProfiles
	stazione a terra: ListSatellites
	stazione a terra: RegisterAgent
	stazione a terra: ReserveContact
	stazione a terra: UpdateAgentStatus
	stazione a terra: UpdateConfig
	stazione a terra: UpdateEphemeris
	stazione a terra: UpdateMissionProfile

Prefisso del servizio	Azioni
guardduty	servizio di guardia: AcceptAdministratorInvitation servizio di guardia: AcceptInvitation servizio di guardia: ArchiveFindings servizio di guardia: CreateDetector servizio di guardia: CreateFilter guardduty:CreateIPSet servizio di guardia: CreateInvestigation servizio di guardia: CreateMalwareProtectionPlan servizio di guardia: CreateMembers servizio di guardia: CreatePublishingDestination servizio di guardia: CreateSampleFindings servizio di guardia: CreateThreatEntitySet servizio di guardia: CreateThreatIntelSet servizio di guardia: CreateTrustedEntitySet servizio di guardia: DeclineInvitations servizio di guardia: DeleteDetector servizio di guardia: DeleteFilter guardduty:DeleteIPSet servizio di guardia: DeleteInvitations servizio di guardia: DeleteMalwareProtectionPlan servizio di guardia: DeleteMembers

Prefisso del servizio	Azioni
	servizio di guardia: DeletePublishingDestination servizio di guardia: DeleteThreatEntitySet servizio di guardia: DeleteThreatIntelSet servizio di guardia: DeleteTrustedEntitySet servizio di guardia: DescribeMalwareScans servizio di guardia: DescribeOrganizationConfiguration servizio di guardia: DescribePublishingDestination servizio di guardia: DisableOrganizationAdminAccount servizio di guardia: DisassociateFromAdministratorAccount servizio di guardia: DisassociateFromMasterAccount servizio di guardia: DisassociateMembers servizio di guardia: EnableOrganizationAdminAccount servizio di guardia: GetAdministratorAccount servizio di guardia: GetCoverageStatistics servizio di guardia: GetDetector servizio di guardia: GetFilter servizio di guardia: GetFindings servizio di guardia: GetFindingsStatistics guardduty:GetIPSet servizio di guardia: GetInvestigation servizio di guardia: GetInvitationsCount

Prefisso del servizio	Azioni
	servizio di guardia: GetMalwareProtectionPlan servizio di guardia: GetMalwareScan servizio di guardia: GetMalwareScanSettings servizio di guardia: GetMasterAccount servizio di guardia: GetMemberDetectors servizio di guardia: GetMembers servizio di guardia: GetOrganizationStatistics servizio di guardia: GetRemainingFreeTrialDays servizio di guardia: GetThreatEntitySet servizio di guardia: GetThreatIntelSet servizio di guardia: GetTrustedEntitySet servizio di guardia: GetUsageStatistics servizio di guardia: InviteMembers servizio di guardia: ListCoverage servizio di guardia: ListDetectors servizio di guardia: ListFilters servizio di guardia: ListFindings guardduty:ListIPSets servizio di guardia: ListInvestigations servizio di guardia: ListInvitations servizio di guardia: ListMalwareProtectionPlans

Prefisso del servizio	Azioni
	servizio di guardia: ListMalwareScans servizio di guardia: ListMembers servizio di guardia: ListOrganizationAdminAccounts servizio di guardia: ListPublishingDestinations servizio di guardia: ListThreatEntitySets servizio di guardia: ListThreatIntelSets servizio di guardia: ListTrustedEntitySets servizio di guardia: StartMalwareScan servizio di guardia: StartMonitoringMembers servizio di guardia: StopMonitoringMembers servizio di guardia: UnarchiveFindings servizio di guardia: UpdateDetector servizio di guardia: UpdateFilter servizio di guardia: UpdateFindingsFeedback guardduty:UpdateIPSet servizio di guardia: UpdateMalwareProtectionPlan servizio di guardia: UpdateMalwareScanSettings servizio di guardia: UpdateMemberDetectors servizio di guardia: UpdateOrganizationConfiguration servizio di guardia: UpdatePublishingDestination servizio di guardia: UpdateThreatEntitySet

Prefisso del servizio	Azioni
	servizio di guardia: UpdateThreatIntelSet servizio di guardia: UpdateTrustedEntitySet

Prefisso del servizio	Azioni
healthlake	lago sanitario: CancelFHIRExportJobWithDelete
	lago sanitario: CreateDataTransformationProfile
	healthlake:CreateFHIRDatastore
	lago sanitario: CreateResource
	lago sanitario: DeleteDataTransformationProfile
	healthlake:DeleteFHIRDatastore
	lago sanitario: DeleteResource
	lago sanitario: DescribeDataTransformationJob
	healthlake:DescribeFHIRDatastore
	lago sanitario: DescribeFHIRExportJob
	lago sanitario: DescribeFHIRExportJobWithGet
	lago sanitario: DescribeFHIRImportJob
	lago sanitario: GetCapabilities
	lago sanitario: GetDataTransformationProfile
	lago sanitario: ListDataTransformationJobs
	lago sanitario: ListDataTransformationProfileVersions
	lago sanitario: ListDataTransformationProfiles
	healthlake:ListFHIRDatastores
	lago sanitario: ListFHIRExportJobs
	lago sanitario: ListFHIRImportJobs
	lago sanitario: PublishDataTransformationProfile

Prefisso del servizio	Azioni
	lago sanitario: ReadResource
	lago sanitario: SearchEverything
	lago sanitario: SearchWithGet
	lago sanitario: SearchWithPost
	lago sanitario: StartFHIRExportJob
	lago sanitario: StartFHIRExportJobWithPost
	lago sanitario: StartFHIRImportJob
	lago sanitario: UpdateDataTransformationProfile
	Healthlake: aggiorna il datastore Fhir
	Healthlake: UpdateProfileWithAgent
	lago sanitario: UpdateResource

Prefisso del servizio	Azioni
honeycode	codice del miele: BatchCreateTableRows codice del miele: BatchDeleteTableRows codice del miele: BatchUpdateTableRows codice del miele: BatchUpsertTableRows codice del miele: DescribeTableDataImportJob codice del miele: GetScreenData codice del miele: InvokeScreenAutomation codice del miele: ListTableColumns codice del miele: ListTableRows codice del miele: ListTables codice del miele: QueryTableRows codice del miele: StartTableDataImportJob

Prefisso del servizio	Azioni
iam	obiettivo: AddClientIDToOpenIDConnectProvider
	Io sono: AddRoleToInstanceProfile
	Io sono: AddUserToGroup
	Io sono: AttachGroupPolicy
	Io sono: AttachRolePolicy
	Io sono: AttachUserPolicy
	Io sono: ChangePassword
	Io sono: CreateAccessKey
	Io sono: CreateAccountAlias
	Io sono: CreateGroup
	Io sono: CreateInstanceProfile
	Io sono: CreateLoginProfile
	Io sono: CreateOpenIDConnectProvider
	Io sono: CreatePolicy
	Io sono: CreatePolicyVersion
	Io sono: CreateRole
	iam:CreateSAMLProvider
	Io sono: CreateServiceLinkedRole
	Io sono: CreateServiceSpecificCredential
	Io sono: CreateUser
	Io sono: CreateVirtualMFADevice

Prefisso del servizio	Azioni
	iam:DeactivateMFADevice
	Io sono: DeleteAccessKey
	Io sono: DeleteAccountAlias
	Io sono: DeleteAccountPasswordPolicy
	Io sono: DeleteCloudFrontPublicKey
	Io sono: DeleteGroup
	Io sono: DeleteGroupPolicy
	Io sono: DeleteInstanceProfile
	Io sono: DeleteLoginProfile
	Io sono: DeleteOpenIDConnectProvider
	Io sono: DeletePolicy
	Io sono: DeletePolicyVersion
	Io sono: DeleteRole
	Io sono: DeleteRolePermissionsBoundary
	Io sono: DeleteRolePolicy
	iam:DeleteSAMLProvider
	Io sono: DeleteSSHPublicKey
	Io sono: DeleteServerCertificate
	Io sono: DeleteServiceLinkedRole
	Io sono: DeleteServiceSpecificCredential
	Io sono: DeleteSigningCertificate

Prefisso del servizio	Azioni
	Io sono: DeleteUser
	Io sono: DeleteUserPermissionsBoundary
	Io sono: DeleteUserPolicy
	Io sono: DeleteVirtualMFADevice
	Io sono: DetachGroupPolicy
	Io sono: DetachRolePolicy
	Io sono: DetachUserPolicy
	Io sono: DisableOrganizationsRootCredentialsManagement
	sono: DisableOrganizationsRootSessions
	sono: DisableOutboundWebIdentityFederation
	iam:EnableMFADevice
	sono: EnableOrganizationsRootCredentialsManagement
	sono: EnableOrganizationsRootSessions
	sono: EnableOutboundWebIdentityFederation
	sono: GenerateCredentialReport
	sono: GenerateOrganizationsAccessReport
	sono: GenerateServiceLastAccessedDetails
	sono: GetAccessKeyLastUsed
	sono: GetAccountAuthorizationDetails
	sono: GetAccountEmailAddress
	sono: GetAccountName

Prefisso del servizio	Azioni
	sono: GetAccountPasswordPolicy
	sono: GetAccountSummary
	sono: GetCloudFrontPublicKey
	sono: GetContextKeysForCustomPolicy
	sono: GetContextKeysForPrincipalPolicy
	sono: GetCredentialReport
	sono: GetGroup
	sono: GetGroupPolicy
	sono: GetInstanceProfile
	sono: GetLoginProfile
	iam:GetMFADevice
	sono: GetOpenIDConnectProvider
	sono: GetOrganizationsAccessReport
	sono: GetOutboundWebIdentityFederationInfo
	sono: GetPolicy
	sono: GetPolicyVersion
	sono: GetRole
	sono: GetRolePolicy
	iam:GetSAMLProvider
	sono: GetSSHPublicKey
	sono: GetServerCertificate

Prefisso del servizio	Azioni
	sono: GetServiceLastAccessedDetails
	sono: GetServiceLastAccessedDetailsWithEntities
	sono: GetServiceLinkedRoleDeletionStatus
	sono: GetUser
	sono: GetUserPolicy
	sono: ListAccessKeys
	sono: ListAccountAliases
	sono: ListAttachedGroupPolicies
	sono: ListAttachedRolePolicies
	sono: ListAttachedUserPolicies
	sono: ListCloudFrontPublicKeys
	sono: ListDelegationRequests
	sono: ListEntitiesForPolicy
	sono: ListGroupPolicies
	sono: ListGroups
	sono: ListGroupsForUser
	sono: ListInstanceProfiles
	sono: ListInstanceProfilesForRole
	iam:ListMFADevices
	sono: ListOpenIDConnectProviders
	sono: ListOrganizationsFeatures

Prefisso del servizio	Azioni
	sono: ListPolicies
	sono: ListPoliciesGrantingServiceAccess
	sono: ListPolicyVersions
	sono: ListRolePolicies
	sono: ListRoles
	iam:ListSAMLProviders
	sono: ListSSHPublicKeys
	sono: ListSTSRegionalEndpointsStatus
	sono: ListServerCertificates
	sono: ListServiceSpecificCredentials
	sono: ListSigningCertificates
	sono: ListUserPolicies
	sono: ListUsers
	sono: ListVirtualMFADevices
	Io sono: PutGroupPolicy
	Io sono: PutRolePermissionsBoundary
	Io sono: PutRolePolicy
	Io sono: PutUserPermissionsBoundary
	Io sono: PutUserPolicy
	Io sono: RemoveClientIDFromOpenIDConnectProvider
	Io sono: RemoveRoleFromInstanceProfile

Prefisso del servizio	Azioni
	Io sono: RemoveUserFromGroup
	Io sono: ResetServiceSpecificCredential
	iam:ResyncMFADevice
	Io sono: SetDefaultPolicyVersion
	Io sono: SetSTSRegionalEndpointStatus
	Io sono: SetSecurityTokenServicePreferences
	Io sono: SimulateCustomPolicy
	Io sono: SimulatePrincipalPolicy
	Io sono: UpdateAccessKey
	Io sono: UpdateAccountEmailAddress
	Io sono: UpdateAccountName
	Io sono: UpdateAccountPasswordPolicy
	Io sono: UpdateAssumeRolePolicy
	Io sono: UpdateCloudFrontPublicKey
	Io sono: UpdateGroup
	Io sono: UpdateLoginProfile
	Io sono: UpdateOpenIDConnectProviderThumbprint
	Io sono: UpdateRole
	Io sono: UpdateRoleDescription
	iam:UpdateSAMLProvider
	Io sono: UpdateSSHPublicKey

Prefisso del servizio	Azioni
	Io sono: UpdateServerCertificate
	Io sono: UpdateServiceSpecificCredential
	Io sono: UpdateSigningCertificate
	Io sono: UpdateUser
	Io sono: UploadCloudFrontPublicKey
	Io sono: UploadSSHPublicKey
	Io sono: UploadServerCertificate
	Io sono: UploadSigningCertificate

Prefisso del servizio	Azioni
identitystore	archivio di identità: CreateGroup archivio di identità: CreateGroupMembership archivio di identità: CreateUser archivio di identità: DeleteGroup archivio di identità: DeleteGroupMembership archivio di identità: DeleteUser archivio di identità: DescribeGroup archivio di identità: DescribeGroupMembership archivio di identità: DescribeUser archivio di identità: GetGroupId archivio di identità: GetGroupMembershipId archivio di identità: GetUserId archivio di identità: IsMemberInGroups archivio di identità: ListGroupMemberships archivio di identità: ListGroupMembershipsForMember archivio di identità: ListGroups archivio di identità: ListUsers archivio di identità: UpdateGroup archivio di identità: UpdateUser

Prefisso del servizio	Azioni
imagebuilder	generatore di immagini: CancelImageCreation
	generatore di immagini: CancelLifecycleExecution
	generatore di immagini: CreateComponent
	generatore di immagini: CreateContainerRecipe
	generatore di immagini: CreateDistributionConfiguration
	generatore di immagini: CreateImage
	generatore di immagini: CreateImagePipeline
	generatore di immagini: CreateImageRecipe
	generatore di immagini: CreateInfrastructureConfiguration
	generatore di immagini: CreateLifecyclePolicy
	generatore di immagini: CreateWorkflow
	generatore di immagini: DeleteComponent
	generatore di immagini: DeleteContainerRecipe
	generatore di immagini: DeleteDistributionConfiguration
	generatore di immagini: DeleteImage
	generatore di immagini: DeleteImagePipeline
	generatore di immagini: DeleteImageRecipe
	generatore di immagini: DeleteInfrastructureConfiguration
	generatore di immagini: DeleteLifecyclePolicy
	generatore di immagini: DeleteWorkflow
	generatore di immagini: DistributeImage

Prefisso del servizio	Azioni
	generatore di immagini: GetComponentPolicy
	generatore di immagini: GetContainerRecipePolicy
	generatore di immagini: GetImagePolicy
	generatore di immagini: GetImageRecipePolicy
	generatore di immagini: GetLifecycleExecution
	generatore di immagini: GetLifecyclePolicy
	generatore di immagini: GetMarketplaceResource
	generatore di immagini: GetWorkflowExecution
	generatore di immagini: GetWorkflowStepExecution
	generatore di immagini: ImportComponent
	generatore di immagini: ImportDiskImage
	generatore di immagini: ImportVmImage
	generatore di immagini: ListComponentBuildVersions
	generatore di immagini: ListComponents
	generatore di immagini: ListContainerRecipes
	generatore di immagini: ListDistributionConfigurations
	generatore di immagini: ListImageBuildVersions
	generatore di immagini: ListImagePackages
	generatore di immagini: ListImagePipelineImages
	generatore di immagini: ListImagePipelines
	generatore di immagini: ListImageRecipes

Prefisso del servizio	Azioni
	generatore di immagini: ListImageScanFindingAggregations
	generatore di immagini: ListImageScanFindings
	generatore di immagini: ListImages
	generatore di immagini: ListInfrastructureConfigurations
	generatore di immagini: ListLifecycleExecutionResources
	generatore di immagini: ListLifecycleExecutions
	generatore di immagini: ListLifecyclePolicies
	generatore di immagini: ListWaitingWorkflowSteps
	generatore di immagini: ListWorkflowExecutions
	generatore di immagini: ListWorkflowStepExecutions
	generatore di immagini: ListWorkflows
	generatore di immagini: PutComponentPolicy
	generatore di immagini: PutContainerRecipePolicy
	generatore di immagini: PutImagePolicy
	generatore di immagini: PutImageRecipePolicy
	generatore di immagini: RetryImage
	generatore di immagini: SendWorkflowStepAction
	generatore di immagini: StartImagePipelineExecution
	generatore di immagini: StartResourceStateUpdate
	generatore di immagini: UpdateDistributionConfiguration
	generatore di immagini: UpdateImagePipeline

Prefisso del servizio	Azioni
	generatore di immagini: UpdateInfrastructureConfiguration

Prefisso del servizio	Azioni
inspector	ispettore: AddAttributesToFindings ispettore: CreateAssessmentTarget ispettore: CreateAssessmentTemplate ispettore: CreateExclusionsPreview ispettore: CreateResourceGroup ispettore: DeleteAssessmentRun ispettore: DeleteAssessmentTarget ispettore: DeleteAssessmentTemplate ispettore: DescribeAssessmentRuns ispettore: DescribeAssessmentTargets ispettore: DescribeAssessmentTemplates ispettore: DescribeCrossAccountAccessRole ispettore: DescribeExclusions ispettore: DescribeFindings ispettore: DescribeResourceGroups ispettore: DescribeRulesPackages ispettore: GetAssessmentReport ispettore: GetExclusionsPreview ispettore: GetTelemetryMetadata ispettore: ListAssessmentRunAgents ispettore: ListAssessmentRuns

Prefisso del servizio	Azioni
	ispettore: ListAssessmentTargets ispettore: ListAssessmentTemplates ispettore: ListEventSubscriptions ispettore: ListExclusions ispettore: ListFindings ispettore: ListRulesPackages ispettore: PreviewAgents ispettore: RegisterCrossAccountAccessRole ispettore: RemoveAttributesFromFindings ispettore: StartAssessmentRun ispettore: StopAssessmentRun ispettore: SubscribeToEvent ispettore: UnsubscribeFromEvent ispettore: UpdateAssessmentTarget

Prefisso del servizio	Azioni
inspector2	ispettore 2: AssociateMember
	ispettore 2: BatchGetAccountStatus
	ispettore 2: BatchGetCodeSnippet
	ispettore 2: BatchGetFindingDetails
	ispettore 2: BatchGetFreeTrialInfo
	ispettore 2: BatchGetMemberEc2DeepInspectionStatus
	ispettore 2: BatchUpdateMemberEc2DeepInspectionStatus
	ispettore 2: CancelFindingsReport
	ispettore 2: CancelSbomExport
	ispettore 2: CreateCisScanConfiguration
	ispettore 2: CreateCodeSecurityIntegration
	ispettore 2: CreateConnector
	ispettore 2: CreateFilter
	ispettore 2: CreateFindingsReport
	ispettore 2: CreateSbomExport
	ispettore 2: DeleteCisScanConfiguration
	ispettore 2: DeleteCodeSecurityIntegration
	ispettore 2: DeleteConnector
	ispettore 2: DeleteFilter
	ispettore 2: DescribeOrganizationConfiguration
	inspector2:Disable

Prefisso del servizio	Azioni
	ispettore 2: DisableDelegatedAdminAccount
	ispettore 2: DisassociateMember
	inspector2:Enable
	ispettore 2: EnableDelegatedAdminAccount
	ispettore 2: GetCisScanReport
	ispettore 2: GetCisScanResultDetails
	ispettore 2: GetClustersForImage
	ispettore 2: GetCodeSecurityIntegration
	ispettore 2: GetCodeSecurityScan
	ispettore 2: GetConfiguration
	ispettore 2: GetDelegatedAdminAccount
	ispettore 2: GetEc2DeepInspectionConfiguration
	ispettore 2: GetEncryptionKey
	ispettore 2: GetFindingsReportStatus
	ispettore 2: GetMember
	ispettore 2: GetSbomExport
	ispettore 2: ListAccountPermissions
	ispettore 2: ListCisScanConfigurations
	ispettore 2: ListCisScanResultsAggregatedByChecks
	ispettore 2: ListCisScanResultsAggregatedByTargetResource
	ispettore 2: ListCisScans

Prefisso del servizio	Azioni
	ispettore 2: ListCodeSecurityIntegrations
	ispettore 2: ListCodeSecurityScanConfigurations
	ispettore 2: ListConnectorScanConfigurations
	ispettore 2: ListConnectors
	ispettore 2: ListCoverage
	ispettore 2: ListCoverageStatistics
	ispettore 2: ListDelegatedAdminAccounts
	ispettore 2: ListFilters
	ispettore 2: ListFindingAggregations
	ispettore 2: ListFindings
	ispettore 2: ListMembers
	ispettore 2: ListUsageTotals
	ispettore 2: ResetEncryptionKey
	ispettore 2: SearchVulnerabilities
	ispettore 2: SendCisSessionHealth
	ispettore 2: SendCisSessionTelemetry
	ispettore 2: StartCisSession
	ispettore 2: StartCodeSecurityScan
	ispettore 2: StopCisSession
	ispettore 2: UpdateCisScanConfiguration
	ispettore 2: UpdateCodeSecurityIntegration

Prefisso del servizio	Azioni
	ispettore 2: UpdateConfiguration ispettore 2: UpdateConnector ispettore 2: UpdateEc2DeepInspectionConfiguration ispettore 2: UpdateEncryptionKey ispettore 2: UpdateFilter ispettore 2: UpdateOrgEc2DeepInspectionConfiguration ispettore 2: UpdateOrganizationConfiguration

Prefisso del servizio	Azioni
iot	IoT: AcceptCertificateTransfer IoT: AddThingToBillingGroup IoT: AddThingToThingGroup IoT: AssociateSbomWithPackageVersion IoT: AssociateTargetsWithJob IoT: AttachPolicy IoT: AttachPrincipalPolicy IoT: AttachSecurityProfile IoT: AttachThingPrincipal IoT: CancelAuditMitigationActionsTask IoT: CancelAuditTask IoT: CancelCertificateTransfer IoT: CancelDetectMitigationActionsTask IoT: CancelJob IoT: CancelJobExecution IoT: ClearDefaultAuthorizer IoT: ConfirmTopicRuleDestination IoT: CreateAuditSuppression IoT: CreateAuthorizer IoT: CreateBillingGroup IoT: CreateCertificateFromCsr

Prefisso del servizio	Azioni
	IoT: CreateCertificateProvider IoT: CreateCommand IoT: CreateCustomMetric IoT: CreateDimension IoT: CreateDomainConfiguration IoT: CreateDynamicThingGroup IoT: CreateFleetMetric IoT: CreateJob IoT: CreateJobTemplate IoT: CreateKeysAndCertificate IoT: CreateMitigationAction iot:CreateOTAUpdate IoT: CreatePackage IoT: CreatePackageVersion IoT: CreatePolicy IoT: CreatePolicyVersion IoT: CreateProvisioningClaim IoT: CreateProvisioningTemplate IoT: CreateProvisioningTemplateVersion IoT: CreateRoleAlias IoT: CreateScheduledAudit

Prefisso del servizio	Azioni
	IoT: CreateSecurityProfile IoT: CreateStream IoT: CreateThing IoT: CreateThingGroup IoT: CreateThingType IoT: CreateTopicRule IoT: CreateTopicRuleDestination IoT: DeleteAccountAuditConfiguration IoT: DeleteAuditSuppression IoT: DeleteAuthorizer IoT: DeleteBillingGroup iot:DeleteCACertificate IoT: DeleteCertificate IoT: DeleteCertificateProvider IoT: DeleteCommand IoT: DeleteCustomMetric IoT: DeleteDimension IoT: DeleteDomainConfiguration IoT: DeleteDynamicThingGroup IoT: DeleteFleetMetric IoT: DeleteJob

Prefisso del servizio	Azioni
	IoT: DeleteJobExecution IoT: DeleteJobTemplate IoT: DeleteMitigationAction iot:DeleteOTAUpdate IoT: DeletePackage IoT: DeletePackageVersion IoT: DeletePolicy IoT: DeletePolicyVersion IoT: DeleteProvisioningTemplate IoT: DeleteProvisioningTemplateVersion IoT: DeleteRegistrationCode IoT: DeleteRoleAlias IoT: DeleteScheduledAudit IoT: DeleteSecurityProfile IoT: DeleteStream IoT: DeleteThing IoT: DeleteThingGroup IoT: DeleteThingType IoT: DeleteTopicRule IoT: DeleteTopicRuleDestination IoT: DeleteV2LoggingLevel

Prefisso del servizio	Azioni
	IoT: DeprecateThingType IoT: DescribeAccountAuditConfiguration IoT: DescribeAuditFinding IoT: DescribeAuditMitigationActionsTask IoT: DescribeAuditSuppression IoT: DescribeAuditTask IoT: DescribeAuthorizer IoT: DescribeBillingGroup iot:DescribeCACertificate IoT: DescribeCertificate IoT: DescribeCertificateProvider IoT: DescribeCustomMetric IoT: DescribeDefaultAuthorizer IoT: DescribeDetectMitigationActionsTask IoT: DescribeDimension IoT: DescribeDomainConfiguration IoT: DescribeEncryptionConfiguration IoT: DescribeEndpoint IoT: DescribeEventConfigurations IoT: DescribeFleetMetric IoT: DescribeIndex

Prefisso del servizio	Azioni
	IoT: DescribeJob IoT: DescribeJobExecution IoT: DescribeJobTemplate IoT: DescribeManagedJobTemplate IoT: DescribeMitigationAction IoT: DescribeProvisioningTemplate IoT: DescribeProvisioningTemplateVersion IoT: DescribeRoleAlias IoT: DescribeScheduledAudit IoT: DescribeSecurityProfile IoT: DescribeStream IoT: DescribeThing IoT: DescribeThingGroup IoT: DescribeThingRegistrationTask IoT: DescribeThingType IoT: DetachPolicy IoT: DetachPrincipalPolicy IoT: DetachSecurityProfile IoT: DetachThingPrincipal IoT: DisableTopicRule IoT: DisassociateSbomFromPackageVersion

Prefisso del servizio	Azioni
	IoT: EnableTopicRule IoT: GetBehaviorModelTrainingSummaries IoT: GetBucketsAggregation IoT: GetCardinality IoT: GetCommand IoT: GetEffectivePolicies IoT: GetJobDocument IoT: GetLoggingOptions iot:GetOTAUpdate IoT: GetPackage IoT: GetPackageConfiguration IoT: GetPackageVersion IoT: GetPercentiles IoT: GetPolicy IoT: GetPolicyVersion IoT: GetRegistrationCode IoT: GetStatistics IoT: GetThingConnectivityData IoT: GetTopicRule IoT: GetTopicRuleDestination IoT: GetV2LoggingOptions

Prefisso del servizio	Azioni
	IoT: ListActiveViolations IoT: ListAttachedPolicies IoT: ListAuditFindings IoT: ListAuditMitigationActionsExecutions IoT: ListAuditMitigationActionsTasks IoT: ListAuditSuppressions IoT: ListAuditTasks IoT: ListAuthorizers IoT: ListBillingGroups iot:ListCACertificates IoT: ListCertificateProviders IoT: ListCertificates IoT: ListCertificatesByCA IoT: ListCommands IoT: ListCustomMetrics IoT: ListDetectMitigationActionsExecutions IoT: ListDetectMitigationActionsTasks IoT: ListDimensions IoT: ListDomainConfigurations IoT: ListFleetMetrics IoT: ListIndices

Prefisso del servizio	Azioni
	IoT: ListJobExecutionsForJob IoT: ListJobExecutionsForThing IoT: ListJobTemplates IoT: ListJobs IoT: ListManagedJobTemplates IoT: ListMetricValues IoT: ListMitigationActions iot:ListOTAUpdates IoT: ListOutgoingCertificates IoT: ListPackageVersions IoT: ListPackages IoT: ListPolicies IoT: ListPolicyPrincipals IoT: ListPolicyVersions IoT: ListPrincipalPolicies IoT: ListPrincipalThings IoT: ListPrincipalThingsV2 IoT: ListProvisioningTemplateVersions IoT: ListProvisioningTemplates IoT: ListRelatedResourcesForAuditFinding IoT: ListRoleAliases

Prefisso del servizio	Azioni
	IoT: ListSbomValidationResults IoT: ListScheduledAudits IoT: ListSecurityProfiles IoT: ListSecurityProfilesForTarget IoT: ListStreams IoT: ListTargetsForPolicy IoT: ListTargetsForSecurityProfile IoT: ListThingGroups IoT: ListThingGroupsForThing IoT: ListThingPrincipals IoT: ListThingPrincipalsV2 IoT: ListThingRegistrationTaskReports IoT: ListThingRegistrationTasks IoT: ListThingTypes IoT: ListThings IoT: ListThingsInBillingGroup IoT: ListThingsInThingGroup IoT: ListTopicRuleDestinations IoT: ListTopicRules IoT: ListV2LoggingLevels IoT: ListViolationEvents

Prefisso del servizio	Azioni
	IoT: PutVerificationStateOnViolation iot:RegisterCACertificate IoT: RegisterCertificate IoT: RegisterCertificateWithoutCA IoT: RegisterThing IoT: RejectCertificateTransfer IoT: RemoveThingFromBillingGroup IoT: RemoveThingFromThingGroup IoT: ReplaceTopicRule IoT: SearchIndex IoT: SetDefaultAuthorizer IoT: SetDefaultPolicyVersion IoT: SetLoggingOptions IoT: SetV2LoggingLevel IoT: SetV2LoggingOptions IoT: StartAuditMitigationActionsTask IoT: StartDetectMitigationActionsTask IoT: StartOnDemandAuditTask IoT: StartThingRegistrationTask IoT: StopThingRegistrationTask IoT: TestAuthorization

Prefisso del servizio	Azioni
	IoT: TestInvokeAuthorizer IoT: TransferCertificate IoT: UpdateAccountAuditConfiguration IoT: UpdateAuditSuppression IoT: UpdateAuthorizer IoT: UpdateBillingGroup iot:UpdateCACertificate IoT: UpdateCertificate IoT: UpdateCertificateProvider IoT: UpdateCommand IoT: UpdateCustomMetric IoT: UpdateDimension IoT: UpdateDomainConfiguration IoT: UpdateDynamicThingGroup IoT: UpdateEncryptionConfiguration IoT: UpdateEventConfigurations IoT: UpdateFleetMetric IoT: UpdateIndexingConfiguration IoT: UpdateJob IoT: UpdateMitigationAction IoT: UpdatePackage

Prefisso del servizio	Azioni
	IoT: UpdatePackageConfiguration IoT: UpdatePackageVersion IoT: UpdateProvisioningTemplate IoT: UpdateRoleAlias IoT: UpdateScheduledAudit IoT: UpdateSecurityProfile IoT: UpdateStream IoT: UpdateThing IoT: UpdateThingGroup IoT: UpdateThingGroupsForThing IoT: UpdateThingType IoT: UpdateTopicRuleDestination IoT: ValidateSecurityProfileBehaviors

Prefisso del servizio	Azioni
iotanalytics	analisi IoT: CancelPipelineReprocessing
	analisi IoT: CreateChannel
	analisi IoT: CreateDataset
	analisi IoT: CreateDatasetContent
	analisi IoT: CreateDatastore
	analisi IoT: CreatePipeline
	analisi IoT: DeleteChannel
	analisi IoT: DeleteDataset
	analisi IoT: DeleteDatasetContent
	analisi IoT: DeleteDatastore
	analisi IoT: DeletePipeline
	analisi IoT: DescribeChannel
	analisi IoT: DescribeDataset
	analisi IoT: DescribeDatastore
	analisi IoT: DescribeLoggingOptions
	analisi IoT: DescribePipeline
	analisi IoT: GetDatasetContent
	analisi IoT: ListChannels
	analisi IoT: ListDatasetContents
	analisi IoT: ListDatasets
	analisi IoT: ListDatastores

Prefisso del servizio	Azioni
	analisi IoT: ListPipelines analisi IoT: PutLoggingOptions analisi IoT: RunPipelineActivity analisi IoT: SampleChannelData analisi IoT: StartPipelineReprocessing analisi IoT: UpdateChannel analisi IoT: UpdateDataset analisi IoT: UpdateDatastore analisi IoT: UpdatePipeline
iotdeviceadvisor	consulente per dispositivi IoT: CreateSuiteDefinition consulente per dispositivi iot: DeleteSuiteDefinition consulente per dispositivi iot: GetEndpoint consulente per dispositivi iot: GetSuiteDefinition consulente per dispositivi iot: GetSuiteRun consulente per dispositivi iot: GetSuiteRunReport consulente per dispositivi iot: ListSuiteDefinitions consulente per dispositivi iot: ListSuiteRuns consulente per dispositivi iot: StartSuiteRun consulente per dispositivi iot: StopSuiteRun consulente per dispositivi iot: UpdateSuiteDefinition

Prefisso del servizio	Azioni
iotevents	eventi IoT: BatchAcknowledgeAlarm
	eventi IoT: BatchDeleteDetector
	eventi IoT: BatchDisableAlarm
	eventi IoT: BatchEnableAlarm
	eventi IoT: BatchResetAlarm
	eventi IoT: BatchSnoozeAlarm
	eventi IoT: BatchUpdateDetector
	eventi IoT: CreateAlarmModel
	eventi IoT: CreateDetectorModel
	eventi IoT: CreateInput
	eventi IoT: DeleteAlarmModel
	eventi IoT: DeleteDetectorModel
	eventi IoT: DeleteInput
	eventi IoT: DescribeAlarm
	eventi IoT: DescribeAlarmModel
	eventi IoT: DescribeDetector
	eventi IoT: DescribeDetectorModel
	eventi IoT: DescribeDetectorModelAnalysis
	eventi IoT: DescribeInput
	eventi IoT: DescribeLoggingOptions
	eventi IoT: GetDetectorModelAnalysisResults

Prefisso del servizio	Azioni
	eventi IoT: ListAlarmModelVersions eventi IoT: ListAlarmModels eventi IoT: ListAlarms eventi IoT: ListDetectorModelVersions eventi IoT: ListDetectorModels eventi IoT: ListDetectors eventi IoT: ListInputRoutings eventi IoT: ListInputs eventi IoT: PutLoggingOptions eventi IoT: StartDetectorModelAnalysis eventi IoT: UpdateAlarmModel eventi IoT: UpdateDetectorModel eventi IoT: UpdateInput
iotfleethub	hub iotfleet: CreateApplication hub iotfleet: DeleteApplication hub iotfleet: DescribeApplication hub iotfleet: ListApplications hub iotfleet: UpdateApplication

Prefisso del servizio	Azioni
iotsitewise	per quanto riguarda il sito IoT: AssociateAssets
	per quanto riguarda il sito IoT: AssociateTimeSeriesToAssetProperty
	per quanto riguarda il sito IoT: BatchAssociateProjectAssets
	per quanto riguarda il sito IoT: BatchDisassociateProjectAssets
	per quanto riguarda il sito IoT: CreateAccessPolicy
	per quanto riguarda il sito IoT: CreateAsset
	per quanto riguarda il sito IoT: CreateAssetModel
	per quanto riguarda il sito IoT: CreateAssetModelCompositeModel
	a livello di sito IoT: CreateBulkImportJob
	a livello di sito IoT: CreateComputationModel
	a livello di sito IoT: CreateDashboard
	a livello di sito IoT: CreateDataset
	a livello di sito IoT: CreateGateway
	a livello di sito IoT: CreatePortal
	a livello di sito IoT: CreateProject
	a livello di sito IoT: DeleteAccessPolicy
	a livello di sito IoT: DeleteAsset
	a livello di sito IoT: DeleteAssetModel
	a livello di sito IoT: DeleteAssetModelCompositeModel
	a livello di sito IoT: DeleteComputationModel

Prefisso del servizio	Azioni
	a livello di sito IoT: DeleteDashboard
	a livello di sito IoT: DeleteDataset
	a livello di sito IoT: DeleteGateway
	a livello di sito IoT: DeletePortal
	a livello di sito IoT: DeleteProject
	a livello di sito IoT: DeleteTimeSeries
	a livello di sito IoT: DescribeAccessPolicy
	a livello di sito IoT: DescribeAsset
	a livello di sito IoT: DescribeAssetCompositeModel
	a livello di sito IoT: DescribeAssetModel
	a livello di sito IoT: DescribeAssetModelCompositeModel
	a livello di sito IoT: DescribeAssetModelInterfaceRelationship
	a livello di sito IoT: DescribeAssetProperty
	a livello di sito IoT: DescribeBulkImportJob
	a livello di sito IoT: DescribeComputationModel
	a livello di sito IoT: DescribeComputationModelExecutionSummary
	a livello di sito IoT: DescribeDashboard
	a livello di sito IoT: DescribeDataset
	a livello di sito IoT: DescribeDefaultEncryptionConfiguration
	a livello di sito IoT: DescribeExecution
	a livello di sito IoT: DescribeGateway

Prefisso del servizio	Azioni
	a livello di sito IoT: DescribeGatewayCapabilityConfiguration
	a livello di sito IoT: DescribeLoggingOptions
	a livello di sito IoT: DescribePortal
	a livello di sito IoT: DescribeProject
	a livello di sito IoT: DescribeStorageConfiguration
	a livello di sito IoT: DescribeTimeSeries
	a livello di sito IoT: DisassociateAssets
	a livello di sito IoT: DisassociateTimeSeriesFromAssetProperty
	a livello di sito IoT: ExecuteAction
	a livello di sito IoT: ExecuteQuery
	a livello di sito IoT: ListAccessPolicies
	a livello di sito IoT: ListActions
	a livello di sito IoT: ListAssetModelCompositeModels
	a livello di sito IoT: ListAssetModelProperties
	a livello di sito IoT: ListAssetModels
	a livello di sito IoT: ListAssetProperties
	a livello di sito IoT: ListAssetRelationships
	a livello di sito IoT: ListAssets
	a livello di sito IoT: ListAssociatedAssets
	a livello di sito IoT: ListBulkImportJobs
	a livello di sito IoT: ListCompositionRelationships

Prefisso del servizio	Azioni
	a livello di sito IoT: ListComputationModelDataBindingUsages
	a livello di sito IoT: ListComputationModelResolveToResources
	a livello di sito IoT: ListComputationModels
	a livello di sito IoT: ListDashboards
	a livello di sito IoT: ListDatasets
	a livello di sito IoT: ListExecutions
	a livello di sito IoT: ListGateways
	a livello di sito IoT: ListInterfaceRelationships
	a livello di sito IoT: ListPortals
	a livello di sito IoT: ListProjectAssets
	a livello di sito IoT: ListProjects
	a livello di sito IoT: ListTimeSeries
	a livello di sito IoT: PutDefaultEncryptionConfiguration
	a livello di sito IoT: PutLoggingOptions
	a livello di sito IoT: PutStorageConfiguration
	a livello di sito IoT: UpdateAccessPolicy
	a livello di sito IoT: UpdateAsset
	a livello di sito IoT: UpdateAssetModel
	a livello di sito IoT: UpdateAssetModelCompositeModel
	a livello di sito IoT: UpdateAssetProperty
	a livello di sito IoT: UpdateComputationModel

Prefisso del servizio	Azioni
	a livello di sito IoT: UpdateDashboard
	a livello di sito IoT: UpdateDataset
	a livello di sito IoT: UpdateGateway
	a livello di sito IoT: UpdateGatewayCapabilityConfiguration
	a livello di sito IoT: UpdatePortal
	a livello di sito IoT: UpdateProject

Prefisso del servizio	Azioni
iottwinmaker	iottwinmaker: CancelMetadataTransferJob iottwinmaker: CreateComponentType iottwinmaker: CreateEntity iottwinmaker: CreateMetadataTransferJob iottwinmaker: CreateScene iottwinmaker: CreateSyncJob iottwinmaker: CreateWorkspace iottwinmaker: DeleteComponentType iottwinmaker: DeleteEntity iottwinmaker: DeleteScene iottwinmaker: DeleteSyncJob iottwinmaker: DeleteWorkspace iottwinmaker: ExecuteQuery iottwinmaker: GetMetadataTransferJob iottwinmaker: GetPricingPlan iottwinmaker: GetScene iottwinmaker: GetSyncJob iottwinmaker: ListComponentTypes iottwinmaker: ListComponents iottwinmaker: ListEntities iottwinmaker: ListMetadataTransferJobs

Prefisso del servizio	Azioni
	iottwinmaker: ListProperties iottwinmaker: ListScenes iottwinmaker: ListSyncJobs iottwinmaker: ListSyncResources iottwinmaker: ListWorkspaces iottwinmaker: UpdateComponentType iottwinmaker: UpdateEntity iottwinmaker: UpdatePricingPlan iottwinmaker: UpdateScene iottwinmaker: UpdateWorkspace

Prefisso del servizio	Azioni
iotwireless	IoT senza fili: AssociateAwsAccountWithPartnerAccount
	IoT wireless: AssociateMulticastGroupWithFuotaTask
	IoT wireless: AssociateWirelessDeviceWithFuotaTask
	IoT wireless: AssociateWirelessDeviceWithMulticastGroup
	IoT wireless: AssociateWirelessDeviceWithThing
	IoT wireless: AssociateWirelessGatewayWithCertificate
	IoT wireless: AssociateWirelessGatewayWithThing
	IoT wireless: CancelMulticastGroupSession
	IoT wireless: CreateDestination
	IoT wireless: CreateDeviceProfile
	IoT wireless: CreateFuotaTask
	IoT wireless: CreateMulticastGroup
	IoT wireless: CreateNetworkAnalyzerConfiguration
	IoT wireless: CreateServiceProfile
	IoT wireless: CreateWirelessDevice
	IoT wireless: CreateWirelessGateway
	IoT wireless: CreateWirelessGatewayTask
	IoT wireless: CreateWirelessGatewayTaskDefinition
	IoT wireless: DeleteDestination
	IoT wireless: DeleteDeviceProfile
	IoT wireless: DeleteFuotaTask

Prefisso del servizio	Azioni
	IoT wireless: DeleteMulticastGroup
	IoT wireless: DeleteNetworkAnalyzerConfiguration
	IoT wireless: DeleteQueuedMessages
	IoT wireless: DeleteServiceProfile
	IoT wireless: DeleteWirelessDevice
	IoT wireless: DeleteWirelessDeviceImportTask
	IoT wireless: DeleteWirelessGateway
	IoT wireless: DeleteWirelessGatewayTask
	IoT wireless: DeleteWirelessGatewayTaskDefinition
	IoT wireless: DeregisterWirelessDevice
	IoT wireless: DisassociateAwsAccountFromPartnerAccount
	IoT wireless: DisassociateMulticastGroupFromFuotaTask
	IoT wireless: DisassociateWirelessDeviceFromFuotaTask
	IoT wireless: DisassociateWirelessDeviceFromMulticastGroup
	IoT wireless: DisassociateWirelessDeviceFromThing
	IoT wireless: DisassociateWirelessGatewayFromCertificate
	IoT wireless: DisassociateWirelessGatewayFromThing
	IoT wireless: GetDestination
	IoT wireless: GetDeviceProfile
	IoT wireless: GetEventConfigurationByResourceTypes
	IoT wireless: GetFuotaTask

Prefisso del servizio	Azioni
	IoT wireless: GetLogLevelsByResourceTypes
	IoT wireless: GetMetricConfiguration
	IoT wireless: GetMetrics
	IoT wireless: GetMulticastGroup
	IoT wireless: GetMulticastGroupSession
	IoT wireless: GetNetworkAnalyzerConfiguration
	IoT wireless: GetPartnerAccount
	IoT wireless: GetPosition
	IoT wireless: GetPositionConfiguration
	IoT wireless: GetPositionEstimate
	IoT wireless: GetResourceEventConfiguration
	IoT wireless: GetResourceLogLevel
	IoT wireless: GetResourcePosition
	IoT wireless: GetServiceEndpoint
	IoT wireless: GetServiceProfile
	IoT wireless: GetWirelessDevice
	IoT wireless: GetWirelessDeviceImportTask
	IoT wireless: GetWirelessDeviceStatistics
	IoT wireless: GetWirelessGateway
	IoT wireless: GetWirelessGatewayCertificate
	IoT wireless: GetWirelessGatewayFirmwareInformation

Prefisso del servizio	Azioni
	IoT wireless: GetWirelessGatewayStatistics
	IoT wireless: GetWirelessGatewayTask
	IoT wireless: GetWirelessGatewayTaskDefinition
	IoT wireless: ListDestinations
	IoT wireless: ListDeviceProfiles
	IoT wireless: ListDevicesForWirelessDeviceImportTask
	IoT wireless: ListEventConfigurations
	IoT wireless: ListFuotaTasks
	IoT wireless: ListMulticastGroups
	IoT wireless: ListMulticastGroupsByFuotaTask
	IoT wireless: ListNetworkAnalyzerConfigurations
	IoT wireless: ListPartnerAccounts
	IoT wireless: ListPositionConfigurations
	IoT wireless: ListQueuedMessages
	IoT wireless: ListServiceProfiles
	IoT wireless: ListWirelessDeviceImportTasks
	IoT wireless: ListWirelessDevices
	IoT wireless: ListWirelessGatewayTaskDefinitions
	IoT wireless: ListWirelessGateways
	IoT wireless: PutPositionConfiguration
	IoT wireless: PutResourceLogLevel

Prefisso del servizio	Azioni
	IoT wireless: ResetAllResourceLogLevels
	IoT wireless: ResetResourceLogLevel
	IoT wireless: SendDataToMulticastGroup
	IoT wireless: SendDataToWirelessDevice
	IoT wireless: StartBulkAssociateWirelessDeviceWithMulticastGroup
	IoT wireless: StartBulkDisassociateWirelessDeviceFromMulticastGroup
	IoT wireless: StartFuotaTask
	IoT wireless: StartMulticastGroupSession
	IoT wireless: StartNetworkAnalyzerStream
	IoT wireless: StartSingleWirelessDeviceImportTask
	IoT wireless: StartWirelessDeviceImportTask
	IoT wireless: TestWirelessDevice
	IoT wireless: UpdateDestination
	IoT wireless: UpdateEventConfigurationByResourceTypes
	IoT wireless: UpdateFuotaTask
	IoT wireless: UpdateLogLevelsByResourceTypes
	IoT wireless: UpdateMetricConfiguration
	IoT wireless: UpdateMulticastGroup
	IoT wireless: UpdateNetworkAnalyzerConfiguration
	IoT wireless: UpdatePartnerAccount

Prefisso del servizio	Azioni
	IoT wireless: UpdatePosition
	IoT wireless: UpdateResourceEventConfiguration
	IoT wireless: UpdateResourcePosition
	IoT wireless: UpdateWirelessDevice
	IoT wireless: UpdateWirelessDeviceImportTask
	IoT wireless: UpdateWirelessGateway

Prefisso del servizio	Azioni
ivs	è: BatchGetChannel è: BatchGetStreamKey è: BatchStartViewerSessionRevocation è: CreateAdConfiguration è: CreateChannel è: CreateEncoderConfiguration è: CreateIngestConfiguration è: CreateParticipantToken è: CreatePlaybackRestrictionPolicy è: CreateRecordingConfiguration è: CreateStorageConfiguration è: CreateStreamKey è: DeleteAdConfiguration è: DeleteChannel è: DeleteEncoderConfiguration è: DeleteIngestConfiguration è: DeletePlaybackKeyPair è: DeletePlaybackRestrictionPolicy è: DeletePublicKey è: DeleteRecordingConfiguration è: DeleteStorageConfiguration

Prefisso del servizio	Azioni
	è: DeleteStreamKey è: DisconnectParticipant è: GetAdConfiguration è: GetChannel è: GetComposition è: GetEncoderConfiguration è: GetIngestConfiguration è: GetParticipant è: GetPlaybackKeyPair è: GetPlaybackRestrictionPolicy è: GetPublicKey è: GetRecordingConfiguration è: GetStorageConfiguration è: GetStream è: GetStreamKey è: GetStreamSession è: ImportPlaybackKeyPair è: ImportPublicKey è: InsertAdBreak è: ListAdConfigurations è: ListChannels

Prefisso del servizio	Azioni
	è: ListCompositions è: ListEncoderConfigurations è: ListIngestConfigurations è: ListParticipantEvents è: ListParticipantReplicas è: ListParticipants è: ListPlaybackKeyPairs è: ListPlaybackRestrictionPolicies è: ListPublicKeys è: ListRecordingConfigurations è: ListStorageConfigurations è: ListStreamKeys è: ListStreamSessions è: ListStreams è: PutMetadata è: StartComposition è: StartViewerSessionRevocation è: StopComposition è: StopStream è: UpdateAdConfiguration è: UpdateChannel

Prefisso del servizio	Azioni
	è: UpdateIngestConfiguration è: UpdatePlaybackRestrictionPolicy
ivschat	visto: CreateChatToken vista chat: CreateLoggingConfiguration vista chat: CreateRoom vista chat: DeleteLoggingConfiguration vista chat: DeleteMessage vista chat: DeleteRoom vista chat: DisconnectUser vista chat: GetLoggingConfiguration vista chat: GetRoom vista chat: ListLoggingConfigurations vista chat: ListRooms vista chat: SendEvent vista chat: UpdateLoggingConfiguration vista chat: UpdateRoom

Prefisso del servizio	Azioni
kafka	kafka: BatchAssociateScramSecret caffè: BatchDisassociateScramSecret caffè: CreateCluster kafka: V2 CreateCluster kafka: CreateConfiguration caffè: CreateReplicator caffè: CreateVpcConnection caffè: DeleteCluster caffè: DeleteClusterPolicy caffè: DeleteConfiguration caffè: DeleteReplicator caffè: DeleteVpcConnection caffè: DescribeCluster caffè: DescribeClusterOperation caffè: DescribeClusterOperationV2 kafka: V2 DescribeCluster kafka: DescribeConfiguration caffè: DescribeConfigurationRevision caffè: DescribeVpcConnection caffè: GetBootstrapBrokers caffè: GetClusterPolicy

Prefisso del servizio	Azioni
	caffè: GetCompatibleKafkaVersions caffè: ListClientVpcConnections caffè: ListClusterOperations caffè: ListClusterOperationsV2 caffè: ListClusters kafka: V2 ListClusters kafka: ListConfigurationRevisions caffè: ListConfigurations caffè: ListKafkaVersions caffè: ListNodes caffè: ListReplicators caffè: ListScramSecrets caffè: ListVpcConnections caffè: PutClusterPolicy caffè: RebootBroker caffè: RejectClientVpcConnection caffè: UpdateBrokerCount caffè: UpdateBrokerStorage caffè: UpdateBrokerType caffè: UpdateClusterConfiguration caffè: UpdateClusterKafkaVersion

Prefisso del servizio	Azioni
	caffè: UpdateConfiguration caffè: UpdateConnectivity caffè: UpdateMonitoring caffè: UpdateRebalancing caffè: UpdateReplicationInfo caffè: UpdateSecurity caffè: UpdateStorage
kafkaconnect	connessione kafka: CreateConnector connessione kafka: CreateCustomPlugin connessione kafka: CreateWorkerConfiguration connessione kafka: DeleteConnector connessione kafka: DeleteCustomPlugin connessione kafka: DeleteWorkerConfiguration connessione kafka: DescribeConnector connessione kafka: DescribeCustomPlugin connessione kafka: DescribeWorkerConfiguration connessione kafka: ListConnectorOperations connessione kafka: ListConnectors connessione kafka: ListCustomPlugins connessione kafka: ListWorkerConfigurations connessione kafka: UpdateConnector

Prefisso del servizio	Azioni
kendra	kendra: AssociateEntitiesToExperience kendra: AssociatePersonasToEntities kendra: BatchDeleteDocument kendra: BatchDeleteFeaturedResultsSet kendra: BatchGetDocumentStatus kendra: BatchPutDocument kendra: ClearQuerySuggestions kendra: CreateAccessControlConfiguration kendra: CreateDataSource kendra: CreateExperience kendra: CreateFaq kendra: CreateFeaturedResultsSet kendra: CreateIndex kendra: CreateQuerySuggestionsBlockList kendra: CreateThesaurus kendra: DeleteDataSource kendra: DeleteExperience kendra: DeleteFaq kendra: DeleteIndex kendra: DeletePrincipalMapping kendra: DeleteQuerySuggestionsBlockList

Prefisso del servizio	Azioni
	kendra: DeleteThesaurus kendra: DescribeAccessControlConfiguration kendra: DescribeDataSource kendra: DescribeExperience kendra: DescribeFaq kendra: DescribeFeaturedResultsSet kendra: DescribeIndex kendra: DescribePrincipalMapping kendra: DescribeQuerySuggestionsBlockList kendra: DescribeQuerySuggestionsConfig kendra: DescribeThesaurus kendra: DisassociateEntitiesFromExperience kendra: DisassociatePersonasFromEntities kendra: GetQuerySuggestions kendra: GetSnapshots kendra: ListAccessControlConfigurations kendra: ListDataSourceSyncJobs kendra: ListDataSources kendra: ListEntityPersonas kendra: ListExperienceEntities kendra: ListExperiences

Prefisso del servizio	Azioni
	kendra: ListFaqs kendra: ListFeaturedResultsSets kendra: ListGroupsOlderThanOrderingId kendra: ListIndices kendra: ListQuerySuggestionsBlockLists kendra: ListThesauri kendra: PutPrincipalMapping kendra:Query kendra:Retrieve kendra: StartDataSourceSyncJob kendra: StopDataSourceSyncJob kendra: SubmitFeedback kendra: UpdateDataSource kendra: UpdateExperience kendra: UpdateFeaturedResultsSet kendra: UpdateIndex kendra: UpdateQuerySuggestionsBlockList kendra: UpdateQuerySuggestionsConfig kendra: UpdateThesaurus

Prefisso del servizio	Azioni
kinesis	cinesi: CreateStream
	cinesi: DecreaseStreamRetentionPeriod
	cinesi: DeleteStream
	cinesi: DeregisterStreamConsumer
	cinesi: DescribeAccountSettings
	cinesi: DescribeLimits
	cinesi: DescribeStream
	cinesi: DescribeStreamConsumer
	cinesi: DescribeStreamSummary
	cinesi: DisableEnhancedMonitoring
	cinesi: EnableEnhancedMonitoring
	cinesi: IncreaseStreamRetentionPeriod
	cinesi: ListShards
	cinesi: ListStreamConsumers
	cinesi: ListStreams
	cinesi: MergeShards
	cinesi: RegisterStreamConsumer
	cinesi: SplitShard
	cinesi: StartStreamEncryption
	cinesi: StopStreamEncryption
	cinesi: UpdateAccountSettings

Prefisso del servizio	Azioni
	cinesi: UpdateShardCount cinesi: UpdateStreamMode

Prefisso del servizio	Azioni
kinesisanalytics	analisi della cinesi: AddApplicationCloudWatchLoggingOption kinesisanalytics: AddApplicationInput kinesisanalytics: AddApplicationInputProcessingConfiguration kinesisanalytics: AddApplicationOutput kinesisanalytics: AddApplicationReferenceDataSource kinesisanalytics: AddApplicationVpcConfiguration kinesisanalytics: CreateApplication kinesisanalytics: CreateApplicationPresignedUrl kinesisanalytics: CreateApplicationSnapshot kinesisanalytics: DeleteApplication kinesisanalytics: DeleteApplicationCloudWatchLoggingOption kinesisanalytics: DeleteApplicationInputProcessingConfiguration kinesisanalytics: DeleteApplicationOutput kinesisanalytics: DeleteApplicationReferenceDataSource kinesisanalytics: DeleteApplicationSnapshot kinesisanalytics: DeleteApplicationVpcConfiguration kinesisanalytics: DescribeApplication kinesisanalytics: DescribeApplicationOperation kinesisanalytics: DescribeApplicationSnapshot kinesisanalytics: DescribeApplicationVersion kinesisanalytics: DiscoverInputSchema

Prefisso del servizio	Azioni
	kinesisanalytics: ListApplicationOperations kinesisanalytics: ListApplicationSnapshots kinesisanalytics: ListApplicationVersions kinesisanalytics: ListApplications kinesisanalytics: RollbackApplication kinesisanalytics: StartApplication kinesisanalytics: StopApplication kinesisanalytics: UpdateApplication kinesisanalytics: UpdateApplicationMaintenanceConfiguration

Prefisso del servizio	Azioni
kms	kms:CancelKeyDeletion
	kms:ConnectCustomKeyStore
	kms:CreateAlias
	kms:CreateCustomKeyStore
	kms:CreateGrant
	kms:CreateKey
	kms:Decrypt
	kms:DeleteAlias
	kms:DeleteCustomKeyStore
	kms:DeleteImportedKeyMaterial
	kms:DeriveSharedSecret
	kms:DescribeCustomKeyStores
	kms:DescribeKey
	kms:DisableKey
	kms:DisableKeyRotation
	kms:DisconnectCustomKeyStore
	kms:EnableKey
	kms:EnableKeyRotation
	kms:Encrypt
	kms:GenerateDataKey
	kms:GenerateDataKeyPair

Prefisso del servizio	Azioni
	kms:GenerateDataKeyPairWithoutPlaintext
	kms:GenerateDataKeyWithoutPlaintext
	kms:GenerateMac
	kms:GenerateRandom
	kms:GetKeyLastUsage
	kms:GetKeyPolicy
	kms:GetKeyRotationStatus
	kms:GetParametersForImport
	kms:GetPublicKey
	kms:ImportKeyMaterial
	kms:ListAliases
	kms:ListGrants
	kms:ListKeyPolicies
	kms:ListKeyRotations
	kms:ListKeys
	kms:ListRetirableGrants
	kms:ReplicateKey
	kms:RetireGrant
	kms:RevokeGrant
	kms:RotateKeyOnDemand
	kms:ScheduleKeyDeletion

Prefisso del servizio	Azioni
	kms:Sign kms:UpdateAlias kms:UpdateCustomKeyStore kms:UpdateKeyDescription kms:UpdatePrimaryRegion kms:Verify kms:VerifyMac

Prefisso del servizio	Azioni
lambda	lambda: AddLayerVersionPermission lambda: AddPermission lambda: CreateAlias lambda: CreateCodeSigningConfig lambda: CreateEventSourceMapping lambda: CreateFunction lambda: CreateFunctionUrlConfig lambda: DeleteAlias lambda: DeleteCapacityProvider lambda: DeleteCodeSigningConfig lambda: DeleteEventSourceMapping lambda: DeleteFunction lambda: DeleteFunctionCodeSigningConfig lambda: DeleteFunctionConcurrency lambda: DeleteFunctionEventInvokeConfig lambda: DeleteFunctionUrlConfig lambda: DeleteLayerVersion lambda: DeleteProvisionedConcurrencyConfig lambda: GetAccountSettings lambda: GetAlias lambda: GetCapacityProvider

Prefisso del servizio	Azioni
	lambda: GetCodeSigningConfig lambda: GetEventSourceMapping lambda: GetFunction lambda: GetFunctionCodeSigningConfig lambda: GetFunctionConcurrency lambda: GetFunctionConfiguration lambda: GetFunctionEventInvokeConfig lambda: GetFunctionRecursionConfig lambda: GetFunctionScalingConfig lambda: GetFunctionUrlConfig lambda: GetLayerVersion lambda: GetLayerVersionPolicy lambda: GetPolicy lambda: GetProvisionedConcurrencyConfig lambda: GetRuntimeManagementConfig lambda: ListAliases lambda: ListCapacityProviders lambda: ListCodeSigningConfigs lambda: ListDurableExecutionsByFunction lambda: ListEventSourceMappings lambda: ListFunctionEventInvokeConfigs

Prefisso del servizio	Azioni
	lambda: ListFunctionUrlConfigs lambda: ListFunctions lambda: ListFunctionsByCodeSigningConfig lambda: ListLayerVersions lambda: ListLayers lambda: ListProvisionedConcurrencyConfigs lambda: ListVersionsByFunction lambda: PublishLayerVersion lambda: PublishVersion lambda: PutFunctionCodeSigningConfig lambda: PutFunctionConcurrency lambda: PutFunctionEventInvokeConfig lambda: PutFunctionRecursionConfig lambda: PutFunctionScalingConfig lambda: PutProvisionedConcurrencyConfig lambda: PutRuntimeManagementConfig lambda: RemoveLayerVersionPermission lambda: RemovePermission lambda: UpdateAlias lambda: UpdateCapacityProvider lambda: UpdateCodeSigningConfig

Prefisso del servizio	Azioni
	lambda: UpdateEventSourceMapping lambda: UpdateFunctionCode lambda: UpdateFunctionConfiguration lambda: UpdateFunctionEventInvokeConfig lambda: UpdateFunctionUrlConfig

Prefisso del servizio	Azioni
lex	lex: BatchCreateCustomVocabularyItem
	lex: BatchDeleteCustomVocabularyItem
	lex: BatchUpdateCustomVocabularyItem
	lex: BuildBotLocale
	lex: CreateBotAlias
	lex: CreateBotReplica
	lex: CreateBotVersion
	lex: CreateExport
	lex: CreateIntentVersion
	lex: CreateResourcePolicy
	lex: CreateSlotTypeVersion
	lex: CreateTestSetDiscrepancyReport
	lex: CreateUploadUrl
	lex: DeleteBot
	lex: DeleteBotAnalyzerRecommendation
	lex: DeleteBotChannelAssociation
	lex: DeleteBotReplica
	lex: DeleteExport
	lex: DeleteImport
	lex: DeleteIntentVersion
	lex: DeleteResourcePolicy

Prefisso del servizio	Azioni
	lex: DeleteSlotTypeVersion
	lex: DeleteTestSet
	lex: DeleteUtterances
	lex: DescribeBotAlias
	lex: DescribeBotAnalyzerRecommendation
	lex: DescribeBotRecommendation
	lex: DescribeBotReplica
	lex: DescribeBotResourceGeneration
	lex: DescribeBotVersion
	lex: DescribeCustomVocabularyMetadata
	lex: DescribeExport
	lex: DescribeImport
	lex: DescribeResourcePolicy
	lex: DescribeTestExecution
	lex: DescribeTestSet
	lex: DescribeTestSetDiscrepancyReport
	lex: DescribeTestSetGeneration
	lex: GenerateBotElement
	lex: GetBot
	lex: GetBotAlias
	lex: GetBotAliases

Prefisso del servizio	Azioni
	lex: GetBotChannelAssociation
	lex: GetBotChannelAssociations
	lex: GetBotVersions
	lex: GetBots
	lex: GetBuiltinIntent
	lex: GetBuiltinIntents
	lex: GetBuiltinSlotTypes
	lex: GetExport
	lex: GetImport
	lex: GetIntent
	lex: GetIntentVersions
	lex: GetIntents
	lex: GetMigration
	lex: GetMigrations
	lex: GetSlotType
	lex: GetSlotTypeVersions
	lex: GetSlotTypes
	lex: GetTestExecutionArtifactsUrl
	lex: GetUtterancesView
	lex: ListBotAliasReplicas
	lex: ListBotAliases

Prefisso del servizio	Azioni
	lex: ListBotRecommendations
	lex: ListBotReplicas
	lex: ListBotResourceGenerations
	lex: ListBotVersionReplicas
	lex: ListBotVersions
	lex: ListBots
	lex: ListBuiltInIntents
	lex: ListBuiltInSlotTypes
	lex: ListCustomVocabularyItems
	lex: ListExports
	lex: ListImports
	lex: ListIntentMetrics
	lex: ListIntentPaths
	lex: ListRecommendedIntents
	lex: ListSessionAnalyticsData
	lex: ListSessionMetrics
	lex: ListTestExecutionResultItems
	lex: ListTestExecutions
	lex: ListTestSets
	lex: PutBot
	lex: PutBotAlias

Prefisso del servizio	Azioni
	lex: PutIntent
	lex: PutSlotType
	lex: SearchAssociatedTranscripts
	lex: StartBotAnalyzer
	lex: StartBotRecommendation
	lex: StartImport
	lex: StartMigration
	lex: StartTestExecution
	lex: StartTestSetGeneration
	lex: StopBotAnalyzer
	lex: StopBotRecommendation
	lex: UpdateBotAlias
	lex: UpdateBotRecommendation
	lex: UpdateExport
	lex: UpdateResourcePolicy

Prefisso del servizio	Azioni
license-manager-linux-subscriptions	abbonamenti license-manager-linux-: DeregisterSubscriptionProvider abbonamenti license-manager-linux-: GetRegisteredSubscriptionProvider abbonamenti license-manager-linux-: GetServiceSettings abbonamenti license-manager-linux-: ListLinuxSubscriptionInstances abbonamenti license-manager-linux-: ListLinuxSubscriptions abbonamenti license-manager-linux-: ListRegisteredSubscriptionProviders abbonamenti license-manager-linux-: RegisterSubscriptionProvider abbonamenti license-manager-linux-: UpdateServiceSettings

Prefisso del servizio	Azioni
lightsail	vela leggera: AllocateStaticIp
	vela leggera: AttachCertificateToDistribution
	vela leggera: AttachDisk
	vela leggera: AttachInstancesToLoadBalancer
	vela leggera: AttachLoadBalancerTlsCertificate
	vela leggera: AttachStaticIp
	vela leggera: CloseInstancePublicPorts
	vela leggera: CopySnapshot
	vela leggera: CreateBucket
	vela leggera: CreateBucketAccessKey
	vela leggera: CreateCertificate
	vela leggera: CreateCloudFormationStack
	vela leggera: CreateContactMethod
	vela leggera: CreateContainerService
	vela leggera: CreateContainerServiceDeployment
	vela leggera: CreateContainerServiceRegistryLogin
	vela leggera: CreateDisk
	vela leggera: CreateDiskFromSnapshot
	vela leggera: CreateDiskSnapshot
	vela leggera: CreateDistribution
	vela leggera: CreateDomain

Prefisso del servizio	Azioni
	vela leggera: CreateGUISessionAccessDetails
	vela leggera: CreateInstanceSnapshot
	vela leggera: CreateInstances
	vela leggera: CreateInstancesFromSnapshot
	vela leggera: CreateKeyPair
	vela leggera: CreateLoadBalancer
	vela leggera: CreateLoadBalancerTlsCertificate
	vela leggera: CreateRelationalDatabase
	vela leggera: CreateRelationalDatabaseFromSnapshot
	vela leggera: CreateRelationalDatabaseSnapshot
	vela leggera: DeleteAlarm
	vela leggera: DeleteAutoSnapshot
	vela leggera: DeleteBucket
	vela leggera: DeleteBucketAccessKey
	vela leggera: DeleteCertificate
	vela leggera: DeleteContactMethod
	vela leggera: DeleteContainerImage
	vela leggera: DeleteContainerService
	vela leggera: DeleteDisk
	vela leggera: DeleteDiskSnapshot
	vela leggera: DeleteDistribution

Prefisso del servizio	Azioni
	vela leggera: DeleteDomain
	vela leggera: DeleteDomainEntry
	vela leggera: DeleteInstance
	vela leggera: DeleteInstanceSnapshot
	vela leggera: DeleteKeyPair
	vela leggera: DeleteKnownHostKeys
	vela leggera: DeleteLoadBalancer
	vela leggera: DeleteLoadBalancerTlsCertificate
	vela leggera: DeleteRelationalDatabase
	vela leggera: DeleteRelationalDatabaseSnapshot
	vela leggera: DetachCertificateFromDistribution
	vela leggera: DetachDisk
	vela leggera: DetachInstancesFromLoadBalancer
	vela leggera: DetachStaticIp
	vela leggera: DisableAddOn
	vela leggera: DownloadDefaultKeyPair
	vela leggera: EnableAddOn
	vela leggera: ExportSnapshot
	vela leggera: GetActiveNames
	vela leggera: GetAlarms
	vela leggera: GetAutoSnapshots

Prefisso del servizio	Azioni
	vela leggera: GetBlueprints
	vela leggera: GetBucketAccessKeys
	vela leggera: GetBucketBundles
	vela leggera: GetBucketMetricData
	vela leggera: GetBuckets
	vela leggera: GetBundles
	vela leggera: GetCertificates
	vela leggera: GetCloudFormationStackRecords
	vela leggera: GetContactMethods
	vela leggera: GetContainerAPIMetadata
	vela leggera: GetContainerImages
	vela leggera: GetContainerLog
	vela leggera: GetContainerServiceDeployments
	vela leggera: GetContainerServiceMetricData
	vela leggera: GetContainerServicePowers
	vela leggera: GetContainerServices
	vela leggera: GetCostEstimate
	vela leggera: GetDisk
	vela leggera: GetDiskSnapshot
	vela leggera: GetDiskSnapshots
	vela leggera: GetDisks

Prefisso del servizio	Azioni
	vela leggera: GetDistributionBundles
	vela leggera: GetDistributionLatestCacheReset
	vela leggera: GetDistributionMetricData
	vela leggera: GetDistributions
	vela leggera: GetDomain
	vela leggera: GetExportSnapshotRecords
	vela leggera: GetInstance
	vela leggera: GetInstanceMetricData
	vela leggera: GetInstancePortStates
	vela leggera: GetInstanceSnapshot
	vela leggera: GetInstanceSnapshots
	vela leggera: GetInstanceState
	vela leggera: GetInstances
	vela leggera: GetKeyPair
	vela leggera: GetKeyPairs
	vela leggera: GetLoadBalancer
	vela leggera: GetLoadBalancerMetricData
	vela leggera: GetLoadBalancerTlsCertificates
	vela leggera: GetLoadBalancerTlsPolicies
	vela leggera: GetLoadBalancers
	vela leggera: GetOperation

Prefisso del servizio	Azioni
	vela leggera: GetOperations
	vela leggera: GetOperationsForResource
	vela leggera: GetRegions
	vela leggera: GetRelationalDatabase
	vela leggera: GetRelationalDatabaseBlueprints
	vela leggera: GetRelationalDatabaseBundles
	vela leggera: GetRelationalDatabaseEvents
	vela leggera: GetRelationalDatabaseLogEvents
	vela leggera: GetRelationalDatabaseLogStreams
	vela leggera: GetRelationalDatabaseMasterUserPassword
	vela leggera: GetRelationalDatabaseMetricData
	vela leggera: GetRelationalDatabaseParameters
	vela leggera: GetRelationalDatabaseSnapshot
	vela leggera: GetRelationalDatabaseSnapshots
	vela leggera: GetRelationalDatabases
	vela leggera: GetSetupHistory
	vela leggera: GetStaticIp
	vela leggera: GetStaticIps
	vela leggera: ImportKeyPair
	vela leggera: IsVpcPeered
	vela leggera: OpenInstancePublicPorts

Prefisso del servizio	Azioni
	vela leggera: PeerVpc
	vela leggera: PutAlarm
	vela leggera: PutInstancePublicPorts
	vela leggera: RebootInstance
	vela leggera: RebootRelationalDatabase
	vela leggera: RegisterContainerImage
	vela leggera: ReleaseStaticIp
	vela leggera: ResetDistributionCache
	vela leggera: SendContactMethodVerification
	vela leggera: SetIpAddressType
	vela leggera: SetResourceAccessForBucket
	vela leggera: SetupInstanceHttps
	lightsail:StartGUISession
	vela leggera: StartInstance
	vela leggera: StartRelationalDatabase
	lightsail:StopGUISession
	vela leggera: StopInstance
	vela leggera: StopRelationalDatabase
	vela leggera: TestAlarm
	vela leggera: UnpeerVpc
	vela leggera: UpdateBucket

Prefisso del servizio	Azioni
	vela leggera: UpdateBucketBundle
	vela leggera: UpdateContainerService
	vela leggera: UpdateDistribution
	vela leggera: UpdateDistributionBundle
	vela leggera: UpdateDomainEntry
	vela leggera: UpdateInstanceMetadataOptions
	vela leggera: UpdateLoadBalancerAttribute
	vela leggera: UpdateRelationalDatabase
	vela leggera: UpdateRelationalDatabaseParameters

Prefisso del servizio	Azioni
log	registri: AssociateKmsKey registri: AssociateSourceToS3TableIntegration registri: CancelExportTask registri: CancelImportTask registri: CreateDelivery registri: CreateExportTask registri: CreateLogAnomalyDetector registri: CreateLogGroup registri: CreateLogStream registri: CreateLookupTable registri: DeleteDataProtectionPolicy registri: DeleteDelivery registri: DeleteDeliveryDestination registri: DeleteDeliveryDestinationPolicy registri: DeleteDeliverySource registri: DeleteDestination registri: DeleteIndexPolicy registri: DeleteIntegration registri: DeleteLogAnomalyDetector registri: DeleteLogGroup registri: DeleteLogStream

Prefisso del servizio	Azioni
	registri: DeleteLookupTable registri: DeleteMetricFilter registri: DeleteQueryDefinition registri: DeleteResourcePolicy registri: DeleteRetentionPolicy registri: DeleteScheduledQuery registri: DeleteSubscriptionFilter registri: DeleteSyslogConfiguration registri: DeleteTransformer registri: DescribeAccountPolicies registri: DescribeConfigurationTemplates registri: DescribeDeliveries registri: DescribeDeliveryDestinations registri: DescribeDeliverySources registri: DescribeDestinations registri: DescribeExportTasks registri: DescribeFieldIndexes registri: DescribeImportTaskBatches registri: DescribeImportTasks registri: DescribeIndexPolicies registri: DescribeLogGroups

Prefisso del servizio	Azioni
	registri: DescribeLogStreams registri: DescribeLookupTables registri: DescribeMetricFilters registri: DescribeQueries registri: DescribeQueryDefinitions registri: DescribeResourcePolicies registri: DescribeSubscriptionFilters registri: DisassociateKmsKey registri: DisassociateSourceFromS3TableIntegration registri: GetDataProtectionPolicy registri: GetDelivery registri: GetDeliveryDestination registri: GetDeliveryDestinationPolicy registri: GetDeliverySource registri: GetIntegration registri: GetLogAnomalyDetector registri: GetLogFields registri: GetLogGroupFields registri: GetLogRecord registri: GetLookupTable registri: GetQueryResults

Prefisso del servizio	Azioni
	registri: GetScheduledQuery registri: GetScheduledQueryHistory registri: GetStorageTierPolicy registri: GetTransformer registri: ListAnomalies registri: ListIntegrations registri: ListLogAnomalyDetectors registri: ListLogGroupsForQuery registri: ListScheduledQueries registri: ListSourcesForS3TableIntegration registri: ListSyslogConfigurations registri: PutBearerTokenAuthentication registri: PutDataProtectionPolicy registri: PutDeliveryDestination registri: PutDeliveryDestinationPolicy registri: PutDeliverySource registri: PutDestination registri: PutDestinationPolicy registri: PutIndexPolicy registri: PutIntegration registri: PutLogGroupDeletionProtection

Prefisso del servizio	Azioni
	registri: PutMetricFilter registri: PutQueryDefinition registri: PutResourcePolicy registri: PutRetentionPolicy registri: PutStorageTierPolicy registri: PutSubscriptionFilter registri: PutSyslogConfiguration registri: PutTransformer registri: StartLiveTail registri: StartQuery registri: StopQuery registri: TestMetricFilter registri: TestTransformer registri: UpdateAnomaly registri: UpdateDeliveryConfiguration registri: UpdateLogAnomalyDetector registri: UpdateLookupTable

Prefisso del servizio	Azioni
lookoutequipment	attrezzatura di avvistamento: CreateDataset
	attrezzatura di avvistamento: CreateInferenceScheduler
	attrezzatura di avvistamento: CreateLabel
	attrezzatura di avvistamento: CreateLabelGroup
	attrezzatura di avvistamento: CreateModel
	attrezzatura di avvistamento: DeleteDataset
	attrezzatura di avvistamento: DeleteInferenceScheduler
	attrezzatura di avvistamento: DeleteLabel
	attrezzatura di avvistamento: DeleteLabelGroup
	attrezzatura di avvistamento: DeleteModel
	attrezzatura di avvistamento: DeleteResourcePolicy
	attrezzatura di avvistamento: DeleteRetrainingScheduler
	attrezzatura di avvistamento: DescribeDataIngestionJob
	attrezzatura di avvistamento: DescribeDataset
	attrezzatura di avvistamento: DescribeInferenceScheduler
	attrezzatura di avvistamento: DescribeLabelGroup
	attrezzatura di avvistamento: DescribeModel
	attrezzatura di avvistamento: DescribeModelVersion
	attrezzatura di avvistamento: DescribeResourcePolicy
	attrezzatura di avvistamento: DescribeRetrainingScheduler
	lookoutequipment:Describelabel

Prefisso del servizio	Azioni
	attrezzatura di avvistamento: ImportDataset
	attrezzatura di avvistamento: ImportModelVersion
	attrezzatura di avvistamento: ListDataIngestionJobs
	attrezzatura di avvistamento: ListDatasets
	attrezzatura di avvistamento: ListInferenceEvents
	attrezzatura di avvistamento: ListInferenceExecutions
	attrezzatura di avvistamento: ListInferenceSchedulers
	attrezzatura di avvistamento: ListLabelGroups
	attrezzatura di avvistamento: ListLabels
	attrezzatura di avvistamento: ListModelVersions
	attrezzatura di avvistamento: ListModels
	attrezzatura di avvistamento: ListRetrainingSchedulers
	attrezzatura di avvistamento: ListSensorStatistics
	attrezzatura di avvistamento: PutResourcePolicy
	attrezzatura di avvistamento: StartDataIngestionJob
	attrezzatura di avvistamento: StartInferenceScheduler
	attrezzatura di avvistamento: StartRetrainingScheduler
	attrezzatura di avvistamento: StopInferenceScheduler
	attrezzatura di avvistamento: StopRetrainingScheduler
	attrezzatura di avvistamento: UpdateActiveModelVersion
	attrezzatura di avvistamento: UpdateInferenceScheduler

Prefisso del servizio	Azioni
	attrezzatura di avvistamento: UpdateLabelGroup attrezzatura di avvistamento: UpdateModel attrezzatura di avvistamento: UpdateRetrainingScheduler

Prefisso del servizio	Azioni
lookoutmetrics	metriche di osservazione: ActivateAnomalyDetector metriche di attenzione: BackTestAnomalyDetector metriche di attenzione: CreateAlert metriche di attenzione: CreateAnomalyDetector metriche di attenzione: CreateMetricSet metriche di attenzione: DeactivateAnomalyDetector metriche di attenzione: DeleteAlert metriche di attenzione: DeleteAnomalyDetector metriche di attenzione: DescribeAlert metriche di attenzione: DescribeAnomalyDetectionExecutions metriche di attenzione: DescribeAnomalyDetector metriche di attenzione: DescribeMetricSet metriche di attenzione: DetectMetricSetConfig metriche di attenzione: GetAnomalyGroup metriche di attenzione: GetDataQualityMetrics metriche di attenzione: GetFeedback metriche di attenzione: GetSampleData metriche di attenzione: ListAlerts metriche di attenzione: ListAnomalyDetectors metriche di attenzione: ListAnomalyGroupRelatedMetrics metriche di attenzione: ListAnomalyGroupSummaries

Prefisso del servizio	Azioni
	metriche di attenzione: ListAnomalyGroupTimeSeries metriche di attenzione: ListMetricSets metriche di attenzione: PutFeedback metriche di attenzione: UpdateAlert metriche di attenzione: UpdateAnomalyDetector metriche di attenzione: UpdateMetricSet

Prefisso del servizio	Azioni
m2	m2: CancelBatchJobExecution
	m2: CreateApplication
	m2: CreateDataSetExportTask
	m2: CreateDataSetImportTask
	m2: CreateDeployment
	m2: CreateEnvironment
	m2: DeleteApplication
	m2: DeleteApplicationFromEnvironment
	m2: DeleteEnvironment
	m2: GetApplication
	m2: GetApplicationVersion
	m2: GetBatchJobExecution
	m2: GetDataSetDetails
	m2: GetDataSetExportTask
	m2: GetDataSetImportTask
	m2: GetDeployment
	m2: GetEnvironment
	m2: GetSignedBluinsightsUrl
	m2: ListApplicationVersions
	m2: ListApplications
	m2: ListBatchJobDefinitions

Prefisso del servizio	Azioni
	m2: ListBatchJobExecutions
	m2: ListBatchJobRestartPoints
	m2: ListDataSetExportHistory
	m2: ListDataSetImportHistory
	m2: ListDataSets
	m2: ListDeployments
	m2: ListEngineVersions
	m2: ListEnvironments
	m2: StartApplication
	m2: StartBatchJob
	m2: StopApplication
	m2: UpdateApplication
	m2: UpdateEnvironment

Prefisso del servizio	Azioni
managedblockchain	blockchain gestita: CreateAccessor
	blockchain gestita: CreateMember
	blockchain gestita: CreateNetwork
	blockchain gestita: CreateNode
	blockchain gestita: CreateProposal
	blockchain gestita: DeleteAccessor
	blockchain gestita: DeleteMember
	blockchain gestita: DeleteNode
	blockchain gestita: GetAccessor
	blockchain gestita: GetMember
	blockchain gestita: GetNetwork
	blockchain gestita: GetNode
	blockchain gestita: GetProposal
	blockchain gestita: InvokeRpcPolygonMainnet
	blockchain gestita: InvokeRpcPolygonMumbaiTestnet
	blockchain gestita: ListAccessors
	blockchain gestita: ListInvitations
	blockchain gestita: ListMembers
	blockchain gestita: ListNetworks
	blockchain gestita: ListNodes
	blockchain gestita: ListProposalVotes

Prefisso del servizio	Azioni
	blockchain gestita: ListProposals blockchain gestita: RejectInvitation blockchain gestita: UpdateMember blockchain gestita: UpdateNode blockchain gestita: VoteOnProposal

Prefisso del servizio	Azioni
mediaconnect	connessione multimediale: AddBridgeOutputs
	connessione multimediale: AddBridgeSources
	connessione multimediale: AddFlowMediaStreams
	connessione multimediale: AddFlowOutputs
	connessione multimediale: AddFlowSources
	connessione multimediale: AddFlowVpcInterfaces
	connessione multimediale: CreateBridge
	connessione multimediale: CreateFlow
	connessione multimediale: CreateGateway
	connessione multimediale: DeleteBridge
	connessione multimediale: DeleteFlow
	connessione multimediale: DeleteGateway
	connessione multimediale: DeleteRouterInput
	connessione multimediale: DeleteRouterNetworkInterface
	connessione multimediale: DeleteRouterOutput
	connessione multimediale: DeregisterGatewayInstance
	connessione multimediale: DescribeBridge
	connessione multimediale: DescribeFlow
	connessione multimediale: DescribeFlowSourceMetadata
	connessione multimediale: DescribeFlowSourceThumbnail
	connessione multimediale: DescribeGateway

Prefisso del servizio	Azioni
	connessione multimediale: DescribeGatewayInstance
	connessione multimediale: DescribeOffering
	connessione multimediale: DescribeReservation
	connessione multimediale: GetRouterInput
	connessione multimediale: GetRouterInputSourceMetadata
	connessione multimediale: GetRouterInputThumbnail
	connessione multimediale: GetRouterNetworkInterface
	connessione multimediale: GetRouterOutput
	connessione multimediale: GrantFlowEntitlements
	connessione multimediale: ListBridges
	connessione multimediale: ListEntitlements
	connessione multimediale: ListFlows
	connessione multimediale: ListGatewayInstances
	connessione multimediale: ListGateways
	connessione multimediale: ListOfferings
	connessione multimediale: ListReservations
	connessione multimediale: ListRouterInputs
	connessione multimediale: ListRouterNetworkInterfaces
	connessione multimediale: ListRouterOutputs
	connessione multimediale: PurchaseOffering
	connessione multimediale: RemoveBridgeOutput

Prefisso del servizio	Azioni
	connessione multimediale: RemoveBridgeSource
	connessione multimediale: RemoveFlowMediaStream
	connessione multimediale: RemoveFlowOutput
	connessione multimediale: RemoveFlowSource
	connessione multimediale: RemoveFlowVpcInterface
	connessione multimediale: RestartRouterInput
	connessione multimediale: RestartRouterOutput
	connessione multimediale: RevokeFlowEntitlement
	connessione multimediale: StartFlow
	connessione multimediale: StartRouterInput
	connessione multimediale: StartRouterOutput
	connessione multimediale: StopFlow
	connessione multimediale: StopRouterInput
	connessione multimediale: StopRouterOutput
	connessione multimediale: TakeRouterInput
	connessione multimediale: UpdateBridge
	connessione multimediale: UpdateBridgeOutput
	connessione multimediale: UpdateBridgeSource
	connessione multimediale: UpdateBridgeState
	connessione multimediale: UpdateFlow
	connessione multimediale: UpdateFlowEntitlement

Prefisso del servizio	Azioni
	connessione multimediale: UpdateFlowMediaStream connessione multimediale: UpdateGatewayInstance

Prefisso del servizio	Azioni
mediaconvert	conversione multimediale: AssociateCertificate conversione multimediale: CancelJob conversione multimediale: CreateJob conversione multimediale: CreateJobTemplate conversione multimediale: CreatePreset conversione multimediale: CreateQueue conversione multimediale: CreateResourceShare conversione multimediale: DeleteJobTemplate conversione multimediale: DeletePolicy conversione multimediale: DeletePreset conversione multimediale: DeleteQueue conversione multimediale: DescribeEndpoints conversione multimediale: DisassociateCertificate conversione multimediale: GetJob conversione multimediale: GetJobTemplate conversione multimediale: GetPolicy conversione multimediale: GetPreset conversione multimediale: GetQueue conversione multimediale: ListJobTemplates conversione multimediale: ListJobs conversione multimediale: ListPresets

Prefisso del servizio	Azioni
	conversione multimediale: ListQueues conversione multimediale: ListVersions mediaconvert:Probe conversione multimediale: PutPolicy conversione multimediale: SearchJobs conversione di file multimediali: UpdateJobTemplate conversione di file multimediali: UpdatePreset conversione di file multimediali: UpdateQueue

Prefisso del servizio	Azioni
medialive	media live: AcceptInputDeviceTransfer
	media in diretta: BatchDelete
	media in diretta: BatchStart
	media in diretta: BatchStop
	media in diretta: BatchUpdateSchedule
	media in diretta: CancelInputDeviceTransfer
	media in diretta: ClaimDevice
	media in diretta: CreateChannel
	media in diretta: CreateChannelPlacementGroup
	media in diretta: CreateCloudWatchAlarmTemplate
	media in diretta: CreateCloudWatchAlarmTemplateGroup
	media in diretta: CreateCluster
	media in diretta: CreateEventBridgeRuleTemplate
	media in diretta: CreateEventBridgeRuleTemplateGroup
	media in diretta: CreateInput
	media in diretta: CreateInputSecurityGroup
	media in diretta: CreateMultiplex
	media in diretta: CreateMultiplexProgram
	media in diretta: CreateNetwork
	media in diretta: CreateNode
	media in diretta: CreateNodeRegistrationScript

Prefisso del servizio	Azioni
	media in diretta: CreatePartnerInput
	media in diretta: CreateSdiSource
	media in diretta: CreateSignalMap
	media in diretta: DeleteChannel
	media in diretta: DeleteChannelPlacementGroup
	media in diretta: DeleteCloudWatchAlarmTemplate
	media in diretta: DeleteCloudWatchAlarmTemplateGroup
	media in diretta: DeleteCluster
	media in diretta: DeleteEventBridgeRuleTemplate
	media in diretta: DeleteEventBridgeRuleTemplateGroup
	media in diretta: DeleteInput
	media in diretta: DeleteInputSecurityGroup
	media in diretta: DeleteMultiplex
	media in diretta: DeleteMultiplexProgram
	media in diretta: DeleteNetwork
	media in diretta: DeleteNode
	media in diretta: DeleteReservation
	media in diretta: DeleteSchedule
	media in diretta: DeleteSdiSource
	media in diretta: DeleteSignalMap
	media in diretta: DescribeAccountConfiguration

Prefisso del servizio	Azioni
	media in diretta: DescribeChannel
	media in diretta: DescribeChannelPlacementGroup
	media in diretta: DescribeCluster
	media in diretta: DescribeInput
	media in diretta: DescribeInputDevice
	media in diretta: DescribeInputDeviceThumbnail
	media in diretta: DescribeInputSecurityGroup
	media in diretta: DescribeMultiplex
	media in diretta: DescribeMultiplexProgram
	media in diretta: DescribeNetwork
	media in diretta: DescribeNode
	media in diretta: DescribeOffering
	media in diretta: DescribeReservation
	media in diretta: DescribeSchedule
	media in diretta: DescribeSdiSource
	media in diretta: DescribeThumbnails
	media in diretta: GetCloudWatchAlarmTemplate
	media in diretta: GetCloudWatchAlarmTemplateGroup
	media in diretta: GetEventBridgeRuleTemplate
	media in diretta: GetEventBridgeRuleTemplateGroup
	media in diretta: GetSignalMap

Prefisso del servizio	Azioni
	media in diretta: ListAlerts
	media in diretta: ListChannelPlacementGroups
	media in diretta: ListChannels
	media in diretta: ListCloudWatchAlarmTemplateGroups
	media in diretta: ListCloudWatchAlarmTemplates
	media in diretta: ListClusterAlerts
	media in diretta: ListClusters
	media in diretta: ListEventBridgeRuleTemplateGroups
	media in diretta: ListEventBridgeRuleTemplates
	media in diretta: ListInputDeviceTransfers
	media in diretta: ListInputDevices
	media in diretta: ListInputSecurityGroups
	media in diretta: ListInputs
	media in diretta: ListMultiplexAlerts
	media in diretta: ListMultiplexPrograms
	media in diretta: ListMultiplexes
	media in diretta: ListNetworks
	media in diretta: ListNodes
	media in diretta: ListOfferings
	media in diretta: ListReservations
	media in diretta: ListSdiSources

Prefisso del servizio	Azioni
	media in diretta: ListSignalMaps
	media in diretta: ListVersions
	media in diretta: PurchaseOffering
	media in diretta: RebootInputDevice
	media in diretta: RejectInputDeviceTransfer
	media in diretta: RestartChannelPipelines
	media in diretta: StartChannel
	media in diretta: StartDeleteMonitorDeployment
	media in diretta: StartInputDevice
	media in diretta: StartInputDeviceMaintenanceWindow
	media in diretta: StartMonitorDeployment
	media in diretta: StartMultiplex
	media in diretta: StartUpdateSignalMap
	media in diretta: StopChannel
	media in diretta: StopInputDevice
	media in diretta: StopMultiplex
	media in diretta: TransferInputDevice
	media in diretta: UpdateAccountConfiguration
	media in diretta: UpdateChannel
	media in diretta: UpdateChannelClass
	media in diretta: UpdateChannelPlacementGroup

Prefisso del servizio	Azioni
	media in diretta: UpdateCloudWatchAlarmTemplate
	media in diretta: UpdateCloudWatchAlarmTemplateGroup
	media in diretta: UpdateCluster
	media in diretta: UpdateEventBridgeRuleTemplate
	media in diretta: UpdateEventBridgeRuleTemplateGroup
	media in diretta: UpdateInput
	media in diretta: UpdateInputDevice
	media in diretta: UpdateInputSecurityGroup
	media in diretta: UpdateMultiplex
	media in diretta: UpdateMultiplexProgram
	media in diretta: UpdateNetwork
	media in diretta: UpdateNode
	media in diretta: UpdateNodeState
	media in diretta: UpdateReservation
	media in diretta: UpdateSdiSource

Prefisso del servizio	Azioni
mediatailor	media tailor: ConfigureLogsForPlaybackConfiguration mediatailor: CreateChannel mediatailor: CreateLiveSource mediatailor: CreatePrefetchSchedule mediatailor: CreateProgram mediatailor: CreateSourceLocation mediatailor: CreateVodSource mediatailor: DeleteChannel mediatailor: DeleteChannelPolicy mediatailor: DeleteLiveSource mediatailor: DeletePlaybackConfiguration mediatailor: DeletePrefetchSchedule mediatailor: DeleteProgram mediatailor: DeleteSourceLocation mediatailor: DeleteVodSource mediatailor: DescribeChannel mediatailor: DescribeLiveSource mediatailor: DescribeProgram mediatailor: DescribeSourceLocation mediatailor: DescribeVodSource mediatailor: GetChannelPolicy

Prefisso del servizio	Azioni
	mediatailor: GetChannelSchedule mediatailor: GetPlaybackConfiguration mediatailor: GetPrefetchSchedule mediatailor: ListAlerts mediatailor: ListChannels mediatailor: ListLiveSources mediatailor: ListPlaybackConfigurations mediatailor: ListPrefetchSchedules mediatailor: ListSourceLocations mediatailor: ListVodSources mediatailor: PutChannelPolicy mediatailor: PutPlaybackConfiguration mediatailor: StartChannel mediatailor: StopChannel mediatailor: UpdateChannel mediatailor: UpdateLiveSource mediatailor: UpdateProgram mediatailor: UpdateSourceLocation mediatailor: UpdateVodSource

Prefisso del servizio	Azioni
db di memoria	db di memoria: BatchUpdateCluster
	db di memoria: CopySnapshot
	db di memoria: CreateAcl
	db di memoria: CreateCluster
	db di memoria: CreateMultiRegionCluster
	db di memoria: CreateParameterGroup
	db di memoria: CreateSnapshot
	db di memoria: CreateSubnetGroup
	db di memoria: CreateUser
	db di memoria: DeleteAcl
	db di memoria: DeleteCluster
	db di memoria: DeleteMultiRegionCluster
	db di memoria: DeleteParameterGroup
	db di memoria: DeleteSnapshot
	db di memoria: DeleteSubnetGroup
	db di memoria: DeleteUser
	db di memoria: DescribeAcls
	db di memoria: DescribeClusters
	db di memoria: DescribeEngineVersions
	db di memoria: DescribeEvents
	db di memoria: DescribeMultiRegionClusters

Prefisso del servizio	Azioni
	db di memoria: DescribeMultiRegionParameterGroups
	db di memoria: DescribeMultiRegionParameters
	db di memoria: DescribeParameterGroups
	db di memoria: DescribeParameters
	db di memoria: DescribeReservedNodes
	db di memoria: DescribeReservedNodesOfferings
	db di memoria: DescribeServiceUpdates
	db di memoria: DescribeSnapshots
	db di memoria: DescribeSubnetGroups
	db di memoria: DescribeUsers
	db di memoria: FailoverShard
	db di memoria: ListAllowedMultiRegionClusterUpdates
	db di memoria: ListAllowedNodeTypeUpdates
	db di memoria: PurchaseReservedNodesOffering
	db di memoria: ResetParameterGroup
	db di memoria: UpdateAcl
	db di memoria: UpdateCluster
	db di memoria: UpdateMultiRegionCluster
	db di memoria: UpdateParameterGroup
	db di memoria: UpdateSubnetGroup
	db di memoria: UpdateUser

Prefisso del servizio	Azioni
mgh	mg: AssociateCreatedArtifact mg: AssociateDiscoveredResource mg: AssociateSourceResource mg: CreateHomeRegionControl mg: CreateProgressUpdateStream mg: DeleteHomeRegionControl mg: DeleteProgressUpdateStream mg: DescribeApplicationState mg: DescribeHomeRegionControls mg: DescribeMigrationTask mg: DisassociateCreatedArtifact mg: DisassociateDiscoveredResource mg: DisassociateSourceResource mg: GetHomeRegion mg: ImportMigrationTask mg: ListApplicationStates mg: ListCreatedArtifacts mg: ListDiscoveredResources mg: ListMigrationTaskUpdates mg: ListMigrationTasks mg: ListProgressUpdateStreams

Prefisso del servizio	Azioni
	mg: ListSourceResources mg: NotifyApplicationState mg: NotifyMigrationTaskState mg: PutResourceAttributes

Prefisso del servizio	Azioni
mgn	mgn: ArchiveApplication
	mgn: ArchiveWave
	mgn: AssociateApplications
	mgn: AssociateSourceServers
	mgn: ChangeServerLifeCycleState
	mgn: CreateApplication
	mgn: CreateConnector
	mgn: CreateLaunchConfigurationTemplate
	mgn: CreateNetworkMigrationDefinition
	mgn: CreateReplicationConfigurationTemplate
	mgn: CreateWave
	mgn: DeleteApplication
	mgn: DeleteConnector
	mgn: DeleteJob
	mgn: DeleteLaunchConfigurationTemplate
	mgn: DeleteNetworkMigrationDefinition
	mgn: DeleteReplicationConfigurationTemplate
	mgn: DeleteSourceServer
	mgn: DeleteVcenterClient
	mgn: DeleteWave
	mgn: DescribeJobLogItems

Prefisso del servizio	Azioni
	mgn: DescribeJobs
	mgn: DescribeLaunchConfigurationTemplates
	mgn: DescribeReplicationConfigurationTemplates
	mgn: DescribeVcenterClients
	mgn: DisassociateApplications
	mgn: DisassociateSourceServers
	mgn: DisconnectFromService
	mgn: FinalizeCutover
	mgn: GetNetworkMigrationDefinition
	mgn: GetNetworkMigrationMapperSegmentConstruct
	mgn: GetReplicationConfiguration
	mgn: InitializeService
	mgn: ListConnectors
	mgn: ListExportErrors
	mgn: ListExports
	mgn: ListImportErrors
	mgn: ListImportFileEnrichments
	mgn: ListImports
	mgn: ListManagedAccounts
	mgn: ListNetworkMigrationAnalyses
	mgn: ListNetworkMigrationAnalysisResults

Prefisso del servizio	Azioni
	mgn: ListNetworkMigrationCodeGenerationSegments
	mgn: ListNetworkMigrationCodeGenerations
	mgn: ListNetworkMigrationDefinitions
	mgn: ListNetworkMigrationDeployedStacks
	mgn: ListNetworkMigrationDeployments
	mgn: ListNetworkMigrationExecutions
	mgn: ListNetworkMigrationMapperSegmentConstructs
	mgn: ListNetworkMigrationMapperSegments
	mgn: ListNetworkMigrationMappingUpdates
	mgn: ListNetworkMigrationMappings
	mgn: ListSourceServerActions
	mgn: ListTemplateActions
	mgn: MarkAsArchived
	mgn: PauseReplication
	mgn: PutSourceServerAction
	mgn: PutTemplateAction
	mgn: RemoveSourceServerAction
	mgn: RemoveTemplateAction
	mgn: ResumeReplication
	mgn: RetryDataReplication
	mgn: StartCutover

Prefisso del servizio	Azioni
	mgn: StartExport
	mgn: StartImport
	mgn: StartImportFileEnrichment
	mgn: StartNetworkMigrationCodeGeneration
	mgn: StartNetworkMigrationMapping
	mgn: StartNetworkMigrationMappingUpdate
	mgn: StartReplication
	mgn: StartTest
	mgn: StopReplication
	mgn: TerminateTargetInstances
	mgn: UnarchiveApplication
	mgn: UnarchiveWave
	mgn: UpdateApplication
	mgn: UpdateConnector
	mgn: UpdateLaunchConfigurationTemplate
	mgn: UpdateNetworkMigrationDefinition
	mgn: UpdateNetworkMigrationMapperSegment
	mgn: UpdateReplicationConfiguration
	mgn: UpdateReplicationConfigurationTemplate
	mgn: UpdateSourceServer
	mgn: UpdateSourceServerReplicationType

Prefisso del servizio	Azioni
	mgn: UpdateWave

Prefisso del servizio	Azioni
migrationhub-strategy	strategia migrationhub: GetAntiPattern migrationhub-strategy: GetApplicationComponentDetails migrationhub-strategy: GetApplicationComponentStrategies migrationhub-strategy: GetAssessment migrationhub-strategy: GetImportFileTask migrationhub-strategy: GetLatestAssessmentId migrationhub-strategy: GetMessage migrationhub-strategy: GetPortfolioPreferences migrationhub-strategy: GetPortfolioSummary migrationhub-strategy: GetRecommendationReportDetails migrationhub-strategy: GetServerDetails migrationhub-strategy: GetServerStrategies migrationhub-strategy: ListAnalyzableServers migrationhub-strategy: ListAntiPatterns migrationhub-strategy: ListApplicationComponents migrationhub-strategy: ListCollectors migrationhub-strategy: ListImportFileTask migrationhub-strategy: ListJarArtifacts migrationhub-strategy: ListServers migrationhub-strategy: PutLogData migrationhub-strategy: PutMetricData

Prefisso del servizio	Azioni
	migrationhub-strategy: PutPortfolioPreferences migrationhub-strategy: RegisterCollector migrationhub-strategy: SendMessage migrationhub-strategy: StartAssessment migrationhub-strategy: StartImportFileTask migrationhub-strategy: StartRecommendationReportGeneration migrationhub-strategy: StopAssessment migrationhub-strategy: UpdateApplicationComponentConfig migrationhub-strategy: UpdateCollectorConfiguration migrationhub-strategy: UpdateServerConfig

Prefisso del servizio	Azioni
mobiletargeting	targeting mobile: CreateApp
	targeting mobile: CreateCampaign
	targeting mobile: CreateEmailTemplate
	targeting mobile: CreateExportJob
	targeting mobile: CreateImportJob
	targeting mobile: CreateInAppTemplate
	targeting mobile: CreateJourney
	targeting mobile: CreatePushTemplate
	targeting mobile: CreateRecommenderConfiguration
	targeting mobile: CreateSegment
	targeting mobile: CreateSmsTemplate
	targeting mobile: CreateVoiceTemplate
	targeting mobile: DeleteAdmChannel
	targeting mobile: DeleteApnsChannel
	targeting mobile: DeleteApnsSandboxChannel
	targeting mobile: DeleteApnsVoipChannel
	targeting mobile: DeleteApnsVoipSandboxChannel
	targeting mobile: DeleteApp
	targeting mobile: DeleteBaiduChannel
	targeting mobile: DeleteCampaign
	targeting mobile: DeleteEmailChannel

Prefisso del servizio	Azioni
	targeting mobile: DeleteEmailTemplate
	targeting mobile: DeleteEndpoint
	targeting mobile: DeleteEventStream
	targeting mobile: DeleteGcmChannel
	targeting mobile: DeleteInAppTemplate
	targeting mobile: DeleteJourney
	targeting mobile: DeletePushTemplate
	targeting mobile: DeleteRecommenderConfiguration
	targeting mobile: DeleteSegment
	targeting mobile: DeleteSmsChannel
	targeting mobile: DeleteSmsTemplate
	targeting mobile: DeleteUserEndpoints
	targeting mobile: DeleteVoiceChannel
	targeting mobile: DeleteVoiceTemplate
	targeting mobile: GetAdmChannel
	targeting mobile: GetApnsChannel
	targeting mobile: GetApnsSandboxChannel
	targeting mobile: GetApnsVoipChannel
	targeting mobile: GetApnsVoipSandboxChannel
	targeting mobile: GetApp
	targeting mobile: GetApplicationDateRangeKpi

Prefisso del servizio	Azioni
	targeting mobile: GetApplicationSettings
	targeting mobile: GetApps
	targeting mobile: GetBaiduChannel
	targeting mobile: GetCampaign
	targeting mobile: GetCampaignActivities
	targeting mobile: GetCampaignDateRangeKpi
	targeting mobile: GetCampaignVersion
	targeting mobile: GetCampaignVersions
	targeting mobile: GetCampaigns
	targeting mobile: GetChannels
	targeting mobile: GetEmailChannel
	targeting mobile: GetEmailTemplate
	targeting mobile: GetEndpoint
	targeting mobile: GetEventStream
	targeting mobile: GetExportJob
	targeting mobile: GetExportJobs
	targeting mobile: GetGcmChannel
	targeting mobile: GetImportJob
	targeting mobile: GetImportJobs
	targeting mobile: GetInAppMessages
	targeting mobile: GetInAppTemplate

Prefisso del servizio	Azioni
	targeting mobile: GetJourney
	targeting mobile: GetJourneyDateRangeKpi
	targeting mobile: GetJourneyExecutionActivityMetrics
	targeting mobile: GetJourneyExecutionMetrics
	targeting mobile: GetJourneyRunExecutionActivityMetrics
	targeting mobile: GetJourneyRunExecutionMetrics
	targeting mobile: GetJourneyRuns
	targeting mobile: GetPushTemplate
	targeting mobile: GetRecommenderConfiguration
	targeting mobile: GetRecommenderConfigurations
	targeting mobile: GetSegment
	targeting mobile: GetSegmentExportJobs
	targeting mobile: GetSegmentImportJobs
	targeting mobile: GetSegmentVersion
	targeting mobile: GetSegmentVersions
	targeting mobile: GetSegments
	targeting mobile: GetSmsChannel
	targeting mobile: GetSmsTemplate
	targeting mobile: GetUserEndpoints
	targeting mobile: GetVoiceChannel
	targeting mobile: GetVoiceTemplate

Prefisso del servizio	Azioni
	targeting mobile: ListJourneys
	targeting mobile: ListTemplateVersions
	targeting mobile: ListTemplates
	targeting mobile: PhoneNumberValidate
	targeting mobile: PutEventStream
	targeting mobile: RemoveAttributes
	targeting mobile: UpdateAdmChannel
	targeting mobile: UpdateApnsChannel
	targeting mobile: UpdateApnsSandboxChannel
	targeting mobile: UpdateApnsVoipChannel
	targeting mobile: UpdateApnsVoipSandboxChannel
	targeting mobile: UpdateApplicationSettings
	targeting mobile: UpdateBaiduChannel
	targeting mobile: UpdateCampaign
	targeting mobile: UpdateEmailChannel
	targeting mobile: UpdateEmailTemplate
	targeting mobile: UpdateEndpoint
	targeting mobile: UpdateEndpointsBatch
	targeting mobile: UpdateGcmChannel
	targeting mobile: UpdateInAppTemplate
	targeting mobile: UpdateJourney

Prefisso del servizio	Azioni
	targeting mobile: UpdateJourneyState
	targeting mobile: UpdatePushTemplate
	targeting mobile: UpdateRecommenderConfiguration
	targeting mobile: UpdateSegment
	targeting mobile: UpdateSmsChannel
	targeting mobile: UpdateSmsTemplate
	targeting mobile: UpdateTemplateActiveVersion
	targeting mobile: UpdateVoiceChannel
	targeting mobile: UpdateVoiceTemplate
	mobiletargeting:VerifyOTPMessage

Prefisso del servizio	Azioni
mq	mq: CreateBroker mq: CreateConfiguration mq: CreateUser mq: DeleteBroker mq: DeleteConfiguration mq: DeleteUser mq: DescribeBroker mq: DescribeBrokerEngineTypes mq: DescribeBrokerInstanceOptions mq: DescribeConfiguration mq: DescribeConfigurationRevision mq: DescribeSharedResources mq: DescribeUser mq: ListBrokers mq: ListConfigurationRevisions mq: ListConfigurations mq: ListUsers mq:Promote mq: RebootBroker mq: UpdateBroker mq: UpdateConfiguration

Prefisso del servizio	Azioni
	mq: UpdateUser

Prefisso del servizio	Azioni
networkmanager	gestore di rete: AcceptAttachment
	gestore di rete: AssociateConnectPeer
	gestore di rete: AssociateCustomerGateway
	gestore di rete: AssociateLink
	gestore di rete: AssociateTransitGatewayConnectPeer
	gestore di rete: CreateConnectAttachment
	gestore di rete: CreateConnectPeer
	gestore di rete: CreateConnection
	gestore di rete: CreateCoreNetwork
	gestore di rete: CreateDevice
	gestore di rete: CreateDirectConnectGatewayAttachment
	gestore di rete: CreateGlobalNetwork
	gestore di rete: CreateLink
	gestore di rete: CreateSite
	gestore di rete: CreateSiteToSiteVpnAttachment
	gestore di rete: CreateTransitGatewayPeering
	gestore di rete: CreateTransitGatewayRouteTableAttachment
	gestore di rete: CreateVpcAttachment
	gestore di rete: DeleteAttachment
	gestore di rete: DeleteConnectPeer
	gestore di rete: DeleteConnection

Prefisso del servizio	Azioni
	gestore di rete: DeleteCoreNetwork
	gestore di rete: DeleteCoreNetworkPolicyVersion
	gestore di rete: DeleteDevice
	gestore di rete: DeleteGlobalNetwork
	gestore di rete: DeleteLink
	gestore di rete: DeletePeering
	gestore di rete: DeleteResourcePolicy
	gestore di rete: DeleteSite
	gestore di rete: DeregisterTransitGateway
	gestore di rete: DescribeGlobalNetworks
	gestore di rete: DisassociateConnectPeer
	gestore di rete: DisassociateCustomerGateway
	gestore di rete: DisassociateLink
	gestore di rete: DisassociateTransitGatewayConnectPeer
	gestore di rete: ExecuteCoreNetworkChangeSet
	gestore di rete: GetConnectAttachment
	gestore di rete: GetConnectPeer
	gestore di rete: GetConnectPeerAssociations
	gestore di rete: GetConnections
	gestore di rete: GetCoreNetwork
	gestore di rete: GetCoreNetworkChangeEvents

Prefisso del servizio	Azioni
	gestore di rete: GetCoreNetworkChangeSet
	gestore di rete: GetCoreNetworkPolicy
	gestore di rete: GetCustomerGatewayAssociations
	gestore di rete: GetDevices
	gestore di rete: GetLinkAssociations
	gestore di rete: GetLinks
	gestore di rete: GetNetworkResourceCounts
	gestore di rete: GetNetworkResourceRelationships
	gestore di rete: GetNetworkResources
	gestore di rete: GetNetworkRoutes
	gestore di rete: GetNetworkTelemetry
	gestore di rete: GetResourcePolicy
	gestore di rete: GetRouteAnalysis
	gestore di rete: GetSiteToSiteVpnAttachment
	gestore di rete: GetSites
	gestore di rete: GetTransitGatewayConnectPeerAssociations
	gestore di rete: GetTransitGatewayPeering
	gestore di rete: GetTransitGatewayRegistrations
	gestore di rete: GetTransitGatewayRouteTableAttachment
	gestore di rete: GetVpcAttachment
	gestore di rete: ListAttachmentRoutingPolicyAssociations

Prefisso del servizio	Azioni
	gestore di rete: ListAttachments
	gestore di rete: ListConnectPeers
	gestore di rete: ListCoreNetworkPolicyVersions
	gestore di rete: ListCoreNetworkPrefixListAssociations
	gestore di rete: ListCoreNetworkRoutingInformation
	gestore di rete: ListCoreNetworks
	gestore di rete: ListOrganizationServiceAccessStatus
	gestore di rete: ListPeerings
	gestore di rete: PutAttachmentRoutingPolicyLabel
	gestore di rete: PutCoreNetworkPolicy
	gestore di rete: PutResourcePolicy
	gestore di rete: RegisterTransitGateway
	gestore di rete: RejectAttachment
	gestore di rete: RemoveAttachmentRoutingPolicyLabel
	gestore di rete: RestoreCoreNetworkPolicyVersion
	gestore di rete: StartOrganizationServiceAccessUpdate
	gestore di rete: StartRouteAnalysis
	gestore di rete: UpdateConnection
	gestore di rete: UpdateCoreNetwork
	gestore di rete: UpdateDevice
	gestore di rete: UpdateDirectConnectGatewayAttachment

Prefisso del servizio	Azioni
	gestore di rete: UpdateGlobalNetwork gestore di rete: UpdateLink gestore di rete: UpdateNetworkResourceMetadata gestore di rete: UpdateSite gestore di rete: UpdateVpcAttachment

Prefisso del servizio	Azioni
nimble	agile: AcceptEulas agile: CreateLaunchProfile agile: CreateStreamingImage agile: CreateStreamingSession agile: CreateStreamingSessionStream agile: CreateStudio agile: CreateStudioComponent agile: DeleteLaunchProfile agile: DeleteLaunchProfileMember agile: DeleteStreamingImage agile: DeleteStreamingSession agile: DeleteStudio agile: DeleteStudioComponent agile: DeleteStudioMember agile: GetEula agile: GetLaunchProfileDetails agile: GetStreamingImage agile: GetStreamingSession agile: GetStreamingSessionBackup agile: GetStreamingSessionStream agile: GetStudio

Prefisso del servizio	Azioni
	agile: GetStudioComponent agile: GetStudioMember agile: ListEulas agile: ListLaunchProfileMembers agile: ListLaunchProfiles agile: ListStreamingImages agile: ListStreamingSessionBackups agile: ListStreamingSessions agile: ListStudioComponents agile: ListStudioMembers agile: ListStudios agile: PutLaunchProfileMembers agile: PutStudioMembers agile: StartStreamingSession agile: StartStudioSSOConfigurationRepair agile: StopStreamingSession agile: UpdateLaunchProfile agile: UpdateLaunchProfileMember agile: UpdateStreamingImage agile: UpdateStudio agile: UpdateStudioComponent

Prefisso del servizio	Azioni
omics	fumetti: AbortMultipartReadSetUpload
	fumetti: AcceptShare
	fumetti: BatchDeleteReadSet
	fumetti: CancelAnnotationImportJob
	fumetti: CancelRun
	fumetti: CancelRunBatch
	fumetti: CancelVariantImportJob
	fumetti: CompleteMultipartReadSetUpload
	fumetti: CreateAnnotationStore
	fumetti: CreateAnnotationStoreVersion
	fumetti: CreateConfiguration
	fumetti: CreateMultipartReadSetUpload
	fumetti: CreateReferenceStore
	fumetti: CreateRunGroup
	fumetti: CreateSequenceStore
	fumetti: CreateShare
	fumetti: CreateVariantStore
	fumetti: CreateWorkflow
	fumetti: CreateWorkflowVersion
	fumetti: DeleteAnnotationStore
	fumetti: DeleteAnnotationStoreVersions

Prefisso del servizio	Azioni
	fumetti: DeleteBatch
	fumetti: DeleteConfiguration
	fumetti: DeleteReference
	fumetti: DeleteReferenceStore
	fumetti: DeleteRun
	fumetti: DeleteRunBatch
	fumetti: DeleteRunGroup
	fumetti: DeleteSequenceStore
	fumetti: DeleteShare
	fumetti: DeleteVariantStore
	fumetti: DeleteWorkflow
	fumetti: DeleteWorkflowVersion
	fumetti: GetAnnotationImportJob
	fumetti: GetAnnotationStore
	fumetti: GetAnnotationStoreVersion
	fumetti: GetBatch
	fumetti: GetConfiguration
	fumetti: GetReadSet
	fumetti: GetReadSetActivationJob
	fumetti: GetReadSetExportJob
	fumetti: GetReadSetImportJob

Prefisso del servizio	Azioni
	fumetti: GetReadSetMetadata
	fumetti: GetReference
	fumetti: GetReferenceImportJob
	fumetti: GetReferenceMetadata
	fumetti: GetReferenceStore
	fumetti: GetRun
	fumetti: GetRunGroup
	fumetti: GetRunTask
	fumetti: GetSequenceStore
	fumetti: GetShare
	fumetti: GetVariantImportJob
	fumetti: GetVariantStore
	fumetti: GetWorkflow
	fumetti: GetWorkflowVersion
	fumetti: ListAnnotationImportJobs
	fumetti: ListAnnotationStoreVersions
	fumetti: ListAnnotationStores
	fumetti: ListBatch
	fumetti: ListConfigurations
	fumetti: ListMultipartReadSetUploads
	fumetti: ListReadSetActivationJobs

Prefisso del servizio	Azioni
	fumetti: ListReadSetExportJobs
	fumetti: ListReadSetImportJobs
	fumetti: ListReadSetUploadParts
	fumetti: ListReadSets
	fumetti: ListReferenceImportJobs
	fumetti: ListReferenceStores
	fumetti: ListReferences
	fumetti: ListRunGroups
	fumetti: ListRunTasks
	fumetti: ListRuns
	fumetti: ListRunsInBatch
	fumetti: ListSequenceStores
	fumetti: ListShares
	fumetti: ListVariantImportJobs
	fumetti: ListVariantStores
	fumetti: ListWorkflowVersions
	fumetti: ListWorkflows
	fumetti: StartAnnotationImportJob
	fumetti: StartReadSetActivationJob
	fumetti: StartReadSetExportJob
	fumetti: StartReadSetImportJob

Prefisso del servizio	Azioni
	fumetti: StartReferenceImportJob
	fumetti: StartRun
	fumetti: StartRunBatch
	fumetti: StartVariantImportJob
	fumetti: UpdateAnnotationStore
	fumetti: UpdateAnnotationStoreVersion
	fumetti: UpdateRunGroup
	fumetti: UpdateVariantStore
	fumetti: UpdateWorkflow
	fumetti: UpdateWorkflowVersion
	fumetti: UploadReadSetPart

Prefisso del servizio	Azioni
opsworks	Ops funziona: AssignInstance opsworks: AssignVolume opsworks: AssociateElasticIp opsworks: AttachElasticLoadBalancer opsworks: CloneStack opsworks: CreateApp opsworks: CreateDeployment opsworks: CreateInstance opsworks: CreateLayer opsworks: CreateStack opsworks: CreateUserProfile opsworks: DeleteApp opsworks: DeleteInstance opsworks: DeleteLayer opsworks: DeleteStack opsworks: DeleteUserProfile opsworks: DeregisterEcsCluster opsworks: DeregisterElasticIp opsworks: DeregisterInstance opsworks: DeregisterRdsDbInstance opsworks: DeregisterVolume

Prefisso del servizio	Azioni
	opsworks: DescribeAgentVersions opsworks: DescribeApps opsworks: DescribeCommands opsworks: DescribeDeployments opsworks: DescribeEcsClusters opsworks: DescribeElasticIps opsworks: DescribeElasticLoadBalancers opsworks: DescribeInstances opsworks: DescribeLayers opsworks: DescribeLoadBasedAutoScaling opsworks: DescribeMyUserProfile opsworks: DescribeOperatingSystems opsworks: DescribePermissions opsworks: DescribeRaidArrays opsworks: DescribeRdsDbInstances opsworks: DescribeServiceErrors opsworks: DescribeStackProvisioningParameters opsworks: DescribeStackSummary opsworks: DescribeStacks opsworks: DescribeTimeBasedAutoScaling opsworks: DescribeUserProfiles

Prefisso del servizio	Azioni
	opsworks: DescribeVolumes opsworks: DetachElasticLoadBalancer opsworks: DisassociateElasticIp opsworks: GetHostnameSuggestion opsworks: GrantAccess opsworks: RebootInstance opsworks: RegisterEcsCluster opsworks: RegisterElasticIp opsworks: RegisterInstance opsworks: RegisterRdsDbInstance opsworks: RegisterVolume opsworks: SetLoadBasedAutoScaling opsworks: SetPermission opsworks: SetTimeBasedAutoScaling opsworks: StartInstance opsworks: StartStack opsworks: StopInstance opsworks: StopStack opsworks: UnassignInstance opsworks: UnassignVolume opsworks: UpdateApp

Prefisso del servizio	Azioni
	opsworks: UpdateElasticIp opsworks: UpdateInstance opsworks: UpdateLayer opsworks: UpdateMyUserProfile opsworks: UpdateRdsDbInstance opsworks: UpdateStack opsworks: UpdateUserProfile opsworks: UpdateVolume

Prefisso del servizio	Azioni
opsworks-cm	opsworks-cm: AssociateNode
	opsworks-cm: CreateBackup
	opsworks-cm: CreateServer
	opsworks-cm: DeleteBackup
	opsworks-cm: DeleteServer
	opsworks-cm: DescribeAccountAttributes
	opsworks-cm: DescribeBackups
	opsworks-cm: DescribeEvents
	opsworks-cm: DescribeNodeAssociationStatus
	opsworks-cm: DescribeServers
	opsworks-cm: DisassociateNode
	opsworks-cm: ExportServerEngineAttribute
	opsworks-cm: RestoreServer
	opsworks-cm: StartMaintenance
	opsworks-cm: UpdateServer
	opsworks-cm: UpdateServerEngineAttributes

Prefisso del servizio	Azioni
organizations	organizzazioni: AcceptHandshake organizzazioni: AttachPolicy organizzazioni: CancelHandshake organizzazioni: CloseAccount organizzazioni: CreateAccount organizzazioni: CreateGovCloudAccount organizzazioni: CreateOrganization organizzazioni: CreateOrganizationalUnit organizzazioni: CreatePolicy organizzazioni: DeclineHandshake organizzazioni: DeleteOrganization organizzazioni: DeleteOrganizationalUnit organizzazioni: DeletePolicy organizzazioni: DeleteResourcePolicy organizzazioni: DeregisterDelegatedAdministrator organizzazioni: DescribeAccount organizzazioni: DescribeCreateAccountStatus organizzazioni: DescribeEffectivePolicy organizzazioni: DescribeHandshake organizzazioni: DescribeOrganization organizzazioni: DescribeOrganizationalUnit

Prefisso del servizio	Azioni
	organizzazioni: DescribePolicy organizzazioni: DescribeResourcePolicy organizzazioni: DescribeResponsibilityTransfer organizzazioni: DetachPolicy organizzazioni: DisableAWSServiceAccess organizzazioni: DisablePolicyType organizzazioni: EnableAWSServiceAccess organizzazioni: EnableAllFeatures organizzazioni: EnablePolicyType organizzazioni: InviteAccountToOrganization organizzazioni: LeaveOrganization organizzazioni: ListAWSServiceAccessForOrganization organizzazioni: ListAccounts organizzazioni: ListAccountsForParent organizzazioni: ListAccountsWithInvalidEffectivePolicy organizzazioni: ListChildren organizzazioni: ListCreateAccountStatus organizzazioni: ListDelegatedAdministrators organizzazioni: ListDelegatedServicesForAccount organizzazioni: ListHandshakesForAccount organizzazioni: ListHandshakesForOrganization

Prefisso del servizio	Azioni
	organizzazioni: ListInboundResponsibilityTransfers organizzazioni: ListOrganizationalUnitsForParent organizzazioni: ListOutboundResponsibilityTransfers organizzazioni: ListParents organizzazioni: ListPolicies organizzazioni: ListPoliciesForTarget organizzazioni: ListRoots organizzazioni: ListTargetsForPolicy organizzazioni: MoveAccount organizzazioni: PutResourcePolicy organizzazioni: RegisterDelegatedAdministrator organizzazioni: RemoveAccountFromOrganization organizzazioni: TerminateResponsibilityTransfer organizzazioni: UpdateOrganizationalUnit organizzazioni: UpdatePolicy organizzazioni: UpdateResponsibilityTransfer

Prefisso del servizio	Azioni
outposts	avamposti: CancelCapacityTask
	avamposti: CancelOrder
	avamposti: CreateOrder
	avamposti: CreateOutpost
	avamposti: CreatePrivateConnectivityConfig
	avamposti: CreateQuote
	avamposti: CreateRenewal
	avamposti: CreateSite
	avamposti: DeleteOutpost
	avamposti: DeleteQuote
	avamposti: DeleteSite
	avamposti: GetCapacityTask
	avamposti: GetCatalogItem
	avamposti: GetConnection
	avamposti: GetOrder
	avamposti: GetOutpost
	avamposti: GetOutpostBillingInformation
	avamposti: GetOutpostInstanceTypes
	avamposti: GetOutpostSupportedInstanceTypes
	avamposti: GetPrivateConnectivityConfig
	avamposti: GetQuote

Prefisso del servizio	Azioni
	avamposti: GetRenewalPricing
	avamposti: GetSite
	avamposti: GetSiteAddress
	avamposti: ListAssetInstances
	avamposti: ListAssets
	avamposti: ListBlockingInstancesForCapacityTask
	avamposti: ListCapacityTasks
	avamposti: ListCatalogItems
	avamposti: ListOrderableInstanceTypes
	avamposti: ListOrders
	avamposti: ListOutposts
	avamposti: ListQuotes
	avamposti: ListSites
	avamposti: StartCapacityTask
	avamposti: StartConnection
	avamposti: UpdateOutpost
	avamposti: UpdateQuote
	avamposti: UpdateSite
	avamposti: UpdateSiteAddress
	avamposti: UpdateSiteRackPhysicalProperties

Prefisso del servizio	Azioni
pi	pipì: CreatePerformanceAnalysisReport pipì: DeletePerformanceAnalysisReport pipì: DescribeDimensionKeys pipì: GetDimensionKeyDetails pipì: GetPerformanceAnalysisReport pipì: GetResourceMetadata pipì: GetResourceMetrics pipì: ListAvailableResourceDimensions pipì: ListAvailableResourceMetrics pipì: ListPerformanceAnalysisReports
pipes	tubi: CreatePipe tubi: DeletePipe tubi: DescribePipe tubi: ListPipes tubi: StartPipe tubi: StopPipe tubi: UpdatePipe

Prefisso del servizio	Azioni
polly	pollame: DeleteLexicon polly: DescribeVoices polly: GetLexicon polly: GetSpeechSynthesisTask polly: ListLexicons polly: ListSpeechSynthesisTasks polly: PutLexicon polly: StartSpeechSynthesisTask polly: SynthesizeSpeech

Prefisso del servizio	Azioni
profile	profilo: AddProfileKey
	profilo: BatchGetCalculatedAttributeForProfile
	profilo: BatchGetProfile
	profilo: CreateCalculatedAttributeDefinition
	profilo: CreateDomain
	profilo: CreateEventStream
	profilo: CreateProfile
	profilo: CreateRecommender
	profilo: CreateRecommenderFilter
	profilo: CreateRecommenderSchema
	profilo: CreateSegmentDefinition
	profilo: CreateSegmentEstimate
	profilo: CreateSegmentSnapshot
	profilo: CreateUploadJob
	profilo: DeleteCalculatedAttributeDefinition
	profilo: DeleteDomain
	profilo: DeleteDomainObjectType
	profilo: DeleteEventStream
	profilo: DeleteIntegration
	profilo: DeleteProfile
	profilo: DeleteProfileKey

Prefisso del servizio	Azioni
	profilo: DeleteProfileObject
	profilo: DeleteProfileObjectType
	profilo: DeleteRecommender
	profilo: DeleteRecommenderFilter
	profilo: DeleteRecommenderSchema
	profilo: DeleteSegmentDefinition
	profilo: DeleteWorkflow
	profilo: DetectProfileObjectType
	profilo: GetAutoMergingPreview
	profilo: GetCalculatedAttributeDefinition
	profilo: GetCalculatedAttributeForProfile
	profilo: GetDomain
	profilo: GetDomainObjectType
	profilo: GetEventStream
	profilo: GetIdentityResolutionJob
	profilo: GetIntegration
	profilo: GetMatches
	profilo: GetObjectTypeAttributeStatistics
	profilo: GetProfileObjectType
	profilo: GetProfileObjectTypeTemplate
	profilo: GetProfileRecommendations

Prefisso del servizio	Azioni
	profilo: GetRecommender
	profilo: GetRecommenderFilter
	profilo: GetRecommenderSchema
	profilo: GetSegmentDefinition
	profilo: GetSegmentEstimate
	profilo: GetSegmentMembership
	profilo: GetSegmentSnapshot
	profilo: GetSimilarProfiles
	profilo: GetUploadJob
	profilo: GetUploadJobPath
	profilo: GetWorkflow
	profilo: GetWorkflowSteps
	profilo: ListAccountIntegrations
	profilo: ListCalculatedAttributeDefinitions
	profilo: ListCalculatedAttributesForProfile
	profilo: ListDomainLayouts
	profilo: ListDomainObjectTypes
	profilo: ListDomains
	profilo: ListEventStreams
	profilo: ListIdentityResolutionJobs
	profilo: ListIntegrations

Prefisso del servizio	Azioni
	profilo: ListObjectTypeAttributeValues
	profilo: ListObjectTypeAttributes
	profilo: ListProfileAttributeValues
	profilo: ListProfileObjectTypeTemplates
	profilo: ListProfileObjectTypes
	profilo: ListProfileObjects
	profilo: ListRecommenderFilters
	profilo: ListRecommenderRecipes
	profilo: ListRecommenderSchemas
	profilo: ListRecommenders
	profilo: ListRuleBasedMatches
	profilo: ListSegmentDefinitions
	profilo: ListUploadJobs
	profilo: ListWorkflows
	profilo: MergeProfiles
	profilo: PutDomainObjectType
	profilo: PutIntegration
	profilo: PutProfileObject
	profilo: PutProfileObjectType
	profilo: SearchProfiles
	profilo: StartRecommender

Prefisso del servizio	Azioni
	profilo: StartUploadJob profilo: StopRecommender profilo: StopUploadJob profilo: UpdateCalculatedAttributeDefinition profilo: UpdateDomain profilo: UpdateProfile profilo: UpdateRecommender

Prefisso del servizio	Azioni
qldb	qldb: CancelJournalKinesisStream qldb: CreateLedger qldb: DeleteLedger qldb: DescribeJournalKinesisStream qldb: DescribeJournalS3Export qldb: DescribeLedger qldb: ExportJournalToS3 qldb: GetBlock qldb: GetDigest qldb: GetRevision qldb: ListJournalKinesisStreamsForLedger qldb: ListJournalS3Exports qldb: ListJournalS3ExportsForLedger qldb: ListLedgers qldb: StreamJournalToKinesis qldb: UpdateLedger qldb: UpdateLedgerPermissionsMode

Prefisso del servizio	Azioni
ram	ram: AcceptResourceShareInvitation
	ram: AssociateResourceShare
	ram: AssociateResourceSharePermission
	ram: CreatePermission
	ram: CreatePermissionVersion
	ram: CreateResourceShare
	ram: DeletePermission
	ram: DeletePermissionVersion
	ram: DeleteResourceShare
	ram: DisassociateResourceShare
	ram: DisassociateResourceSharePermission
	ram: EnableSharingWithAwsOrganization
	ram: GetPermission
	ram: GetResourcePolicies
	ram: GetResourceShareAssociations
	ram: GetResourceShareInvitations
	ram: GetResourceShares
	ram: ListPendingInvitationResources
	ram: ListPermissionAssociations
	ram: ListPermissionVersions
	ram: ListPermissions

Prefisso del servizio	Azioni
	ram: ListPrincipals ram: ListReplacePermissionAssociationsWork ram: ListResourceSharePermissions ram: ListResourceTypes ram: ListResources ram: ListSourceAssociations ram: PromotePermissionCreatedFromPolicy ram: PromoteResourceShareCreatedFromPolicy ram: RejectResourceShareInvitation ram: ReplacePermissionAssociations ram: SetDefaultPermissionVersion ram: UpdateResourceShare
rbin	pettine: CreateRule rbin: DeleteRule rbin: GetRule rbin: ListRules rbin: LockRule rbin: UnlockRule rbin: UpdateRule

Prefisso del servizio	Azioni
rds	rds: AddRoleToDBCluster rds: AddRoleToDBInstance rds: AddSourceIdentifierToSubscription rds: ApplyPendingMaintenanceAction rds: AuthorizeDBSecurityGroupIngress rds:BacktrackDBCluster rds: CancelExportTask rds: CopyDBClusterParameterGroup rds: CopyDBClusterSnapshot rds: CopyDBParameterGroup rds:CopyDBSnapshot rds: CopyOptionGroup rds: CreateCustomDBEngineVersion rds: CreateDBClusterParameterGroup rds: CreateDBParameterGroup rds:CreateDBProxy rds: CreateDBProxyEndpoint rds: CreateDBSecurityGroup rds: CreateDBSubnetGroup rds: CreateEventSubscription rds: CreateGlobalCluster

Prefisso del servizio	Azioni
	rds: CreateOptionGroup
	rds: DeleteBlueGreenDeployment
	rds: DeleteDBClusterAutomatedBackup
	rds: DeleteDBClusterParameterGroup
	rds: DeleteDBClusterSnapshot
	rds: DeleteDBInstanceAutomatedBackup
	rds: DeleteDBParameterGroup
	rds:DeleteDBProxy
	rds: DeleteDBProxyEndpoint
	rds: DeleteDBSecurityGroup
	rds:DeleteDBSnapshot
	rds: DeleteDBSubnetGroup
	rds: DeleteEventSubscription
	rds: DeleteGlobalCluster
	rds: DeleteOptionGroup
	rds: DeregisterDBProxyTargets
	rds: DescribeAccountAttributes
	rds: DescribeBlueGreenDeployments
	rds: DescribeCertificates
	rds: DescribeDBClusterAutomatedBackups
	rds: DescribeDBClusterBacktracks

Prefisso del servizio	Azioni
	rds: DescribeDBClusterEndpoints
	rds: DescribeDBClusterParameterGroups
	rds: DescribeDBClusterParameters
	rds: DescribeDBClusterSnapshotAttributes
	rds: DescribeDBClusterSnapshots
	rds:DescribeDBClusters
	rds: DescribeDBEngineVersions
	rds: DescribeDBInstanceAutomatedBackups
	rds:DescribeDBInstances
	rds: DescribeDBLogFiles
	rds: DescribeDBMajorEngineVersions
	rds: DescribeDBParameterGroups
	rds:DescribeDBParameters
	rds:DescribeDBProxies
	rds: DescribeDBProxyEndpoints
	rds: DescribeDBProxyTargetGroups
	rds: DescribeDBProxyTargets
	rds:DescribeDBRecommendations
	rds: DescribeDBSecurityGroups
	rds: DescribeDBSnapshotAttributes
	rds: DescribeDBSnapshotTenantDatabases

Prefisso del servizio	Azioni
	rds:DescribeDBSnapshots
	rds: DescribeDBSubnetGroups
	rds: DescribeEngineDefaultClusterParameters
	rds: DescribeEngineDefaultParameters
	rds: DescribeEventCategories
	rds: DescribeEventSubscriptions
	rds: DescribeEvents
	rds: DescribeExportTasks
	rds: DescribeGlobalClusters
	rds: DescribeIntegrations
	rds: DescribeOptionGroupOptions
	rds: DescribeOptionGroups
	rds: DescribeOrderableDBInstanceOptions
	rds: DescribePendingMaintenanceActions
	rds: DescribeReservedDBInstances
	rds: DescribeReservedDBInstancesOfferings
	rds: DescribeSourceRegions
	rds: DescribeTenantDatabases
	rds: DescribeValidDBInstanceModifications
	rds: DownloadCompleteDBLogFile
	rds: DownloadDBLogFilePortion

Prefisso del servizio	Azioni
	rds:FailoverDBCluster
	rds: FailoverGlobalCluster
	rds: ModifyActivityStream
	rds: ModifyCertificates
	rds: ModifyCurrentDBClusterCapacity
	rds: ModifyDBClusterEndpoint
	rds: ModifyDBClusterParameterGroup
	rds: ModifyDBClusterSnapshotAttribute
	rds: ModifyDBParameterGroup
	rds:ModifyDBProxy
	rds: ModifyDBProxyEndpoint
	rds: ModifyDBProxyTargetGroup
	rds:ModifyDBRecommendation
	rds:ModifyDBSnapshot
	rds: ModifyDBSnapshotAttribute
	rds: ModifyDBSubnetGroup
	rds: ModifyEventSubscription
	rds: ModifyGlobalCluster
	rds: ModifyOptionGroup
	rds: ModifyTenantDatabase
	rds: PurchaseReservedDBInstancesOffering

Prefisso del servizio	Azioni
	rds:RebootDBCluster
	rds: RegisterDBProxyTargets
	rds: RemoveFromGlobalCluster
	rds: RemoveRoleFromDBCluster
	rds: RemoveRoleFromDBInstance
	rds: RemoveSourceIdentifierFromSubscription
	rds: ResetDBClusterParameterGroup
	rds: ResetDBParameterGroup
	rds: RestoreDBClusterFromS3
	rds: RestoreDBClusterFromSnapshot
	rds: RestoreDBClusterToPointInTime
	rds: RestoreDBInstanceFromDBSnapshot
	rds: RestoreDBInstanceFromS3
	rds: RestoreDBInstanceToPointInTime
	rds: RevokeDBSecurityGroupIngress
	rds: StartActivityStream
	rds:StartDBCluster
	rds:StartDBInstance
	rds: StartDBInstanceAutomatedBackupsReplication
	rds: StartExportTask
	rds: StopActivityStream

Prefisso del servizio	Azioni
	rds:StopDBCluster rds:StopDBInstance rds: StopDBInstanceAutomatedBackupsReplication rds: SwitchoverBlueGreenDeployment rds: SwitchoverGlobalCluster rds: SwitchoverReadReplica

Prefisso del servizio	Azioni
redshift	spostamento verso il rosso: AcceptReservedNodeExchange spostamento verso il rosso: AddPartner spostamento verso il rosso: AssociateDataShareConsumer spostamento verso il rosso: AuthorizeClusterSecurityGroupIngress spostamento verso il rosso: AuthorizeDataShare spostamento verso il rosso: AuthorizeEndpointAccess spostamento verso il rosso: AuthorizeSnapshotAccess spostamento verso il rosso: BatchDeleteClusterSnapshots spostamento verso il rosso: BatchModifyClusterSnapshots spostamento verso il rosso: CancelResize spostamento verso il rosso: CopyClusterSnapshot spostamento verso il rosso: CreateAuthenticationProfile spostamento verso il rosso: CreateCluster spostamento verso il rosso: CreateClusterParameterGroup spostamento verso il rosso: CreateClusterSecurityGroup spostamento verso il rosso: CreateClusterSnapshot spostamento verso il rosso: CreateClusterSubnetGroup spostamento verso il rosso: CreateCustomDomainAssociation spostamento verso il rosso: CreateEndpointAccess spostamento verso il rosso: CreateEventSubscription spostamento verso il rosso: CreateHsmClientCertificate

Prefisso del servizio	Azioni
	spostamento verso il rosso: CreateHsmConfiguration spostamento verso il rosso: CreateIntegration spostamento verso il rosso: CreateRedshiftIdcApplication spostamento verso il rosso: CreateScheduledAction spostamento verso il rosso: CreateSnapshotCopyGrant spostamento verso il rosso: CreateSnapshotSchedule spostamento verso il rosso: CreateUsageLimit spostamento verso il rosso: DeauthorizeDataShare spostamento verso il rosso: DeleteAuthenticationProfile spostamento verso il rosso: DeleteCluster spostamento verso il rosso: DeleteClusterParameterGroup spostamento verso il rosso: DeleteClusterSecurityGroup spostamento verso il rosso: DeleteClusterSnapshot spostamento verso il rosso: DeleteClusterSubnetGroup spostamento verso il rosso: DeleteCustomDomainAssociation spostamento verso il rosso: DeleteEndpointAccess spostamento verso il rosso: DeleteEventSubscription spostamento verso il rosso: DeleteHsmClientCertificate spostamento verso il rosso: DeleteHsmConfiguration spostamento verso il rosso: DeletePartner spostamento verso il rosso: DeleteRedshiftIdcApplication

Prefisso del servizio	Azioni
	spostamento verso il rosso: DeleteResourcePolicy spostamento verso il rosso: DeleteScheduledAction spostamento verso il rosso: DeleteSnapshotCopyGrant spostamento verso il rosso: DeleteSnapshotSchedule spostamento verso il rosso: DeleteUsageLimit spostamento verso il rosso: DeregisterNamespace spostamento verso il rosso: DescribeAccountAttributes spostamento verso il rosso: DescribeAuthenticationProfiles spostamento verso il rosso: DescribeClusterDbRevisions spostamento verso il rosso: DescribeClusterParameterGroups spostamento verso il rosso: DescribeClusterParameters spostamento verso il rosso: DescribeClusterSecurityGroups spostamento verso il rosso: DescribeClusterSnapshots spostamento verso il rosso: DescribeClusterSubnetGroups spostamento verso il rosso: DescribeClusterTracks spostamento verso il rosso: DescribeClusterVersions spostamento verso il rosso: DescribeClusters spostamento verso il rosso: DescribeCustomDomainAssociations spostamento verso il rosso: DescribeDataShares spostamento verso il rosso: DescribeDataSharesForConsumer spostamento verso il rosso: DescribeDataSharesForProducer

Prefisso del servizio	Azioni
	spostamento verso il rosso: DescribeDefaultClusterParameters spostamento verso il rosso: DescribeEndpointAccess spostamento verso il rosso: DescribeEndpointAuthorization spostamento verso il rosso: DescribeEventCategories spostamento verso il rosso: DescribeEventSubscriptions spostamento verso il rosso: DescribeEvents spostamento verso il rosso: DescribeHsmClientCertificates spostamento verso il rosso: DescribeHsmConfigurations spostamento verso il rosso: DescribeInboundIntegrations spostamento verso il rosso: DescribeIntegrations spostamento verso il rosso: DescribeLoggingStatus spostamento verso il rosso: DescribeNodeConfigurationOptions spostamento verso il rosso: DescribeOrderableClusterOptions spostamento verso il rosso: DescribePartners spostamento verso il rosso: DescribeRedshiftIdcApplications spostamento verso il rosso: DescribeReservedNodeExchangeStatus spostamento verso il rosso: DescribeReservedNodeOfferings spostamento verso il rosso: DescribeReservedNodes spostamento verso il rosso: DescribeResize spostamento verso il rosso: DescribeScheduledActions

Prefisso del servizio	Azioni
	spostamento verso il rosso: DescribeSnapshotCopyGrants spostamento verso il rosso: DescribeSnapshotSchedules spostamento verso il rosso: DescribeStorage spostamento verso il rosso: DescribeTableRestoreStatus spostamento verso il rosso: DescribeUsageLimits spostamento verso il rosso: DisableLogging spostamento verso il rosso: DisableSnapshotCopy spostamento verso il rosso: DisassociateDataShareConsumer spostamento verso il rosso: EnableLogging spostamento verso il rosso: EnableSnapshotCopy spostamento verso il rosso: FailoverPrimaryCompute spostamento verso il rosso: GetClusterCredentials spostamento verso il rosso: GetClusterCredentialsWithIAM spostamento verso il rosso: GetIdentityCenterAuthToken spostamento verso il rosso: GetReservedNodeExchangeConfigurationOptions spostamento verso il rosso: GetReservedNodeExchangeOfferings spostamento verso il rosso: GetResourcePolicy spostamento verso il rosso: ListRecommendations spostamento verso il rosso: ModifyAquaConfiguration spostamento verso il rosso: ModifyAuthenticationProfile

Prefisso del servizio	Azioni
	spostamento verso il rosso: ModifyCluster spostamento verso il rosso: ModifyClusterDbRevision spostamento verso il rosso: ModifyClusterIamRoles spostamento verso il rosso: ModifyClusterMaintenance spostamento verso il rosso: ModifyClusterParameterGroup spostamento verso il rosso: ModifyClusterSnapshot spostamento verso il rosso: ModifyClusterSnapshotSchedule spostamento verso il rosso: ModifyClusterSubnetGroup spostamento verso il rosso: ModifyCustomDomainAssociation spostamento verso il rosso: ModifyEndpointAccess spostamento verso il rosso: ModifyEventSubscription spostamento verso il rosso: ModifyRedshiftIdcApplication spostamento verso il rosso: ModifyScheduledAction spostamento verso il rosso: ModifySnapshotCopyRetentionPeriod spostamento verso il rosso: ModifySnapshotSchedule spostamento verso il rosso: ModifyUsageLimit spostamento verso il rosso: PauseCluster spostamento verso il rosso: PurchaseReservedNodeOffering spostamento verso il rosso: PutResourcePolicy spostamento verso il rosso: RebootCluster spostamento verso il rosso: RegisterNamespace

Prefisso del servizio	Azioni
	spostamento verso il rosso: RejectDataShare spostamento verso il rosso: ResetClusterParameterGroup spostamento verso il rosso: ResizeCluster spostamento verso il rosso: RestoreFromClusterSnapshot spostamento verso il rosso: RestoreTableFromClusterSnapshot spostamento verso il rosso: ResumeCluster spostamento verso il rosso: RevokeClusterSecurityGroupIngress spostamento verso il rosso: RevokeEndpointAccess spostamento verso il rosso: RevokeSnapshotAccess spostamento verso il rosso: RotateEncryptionKey spostamento verso il rosso: UpdatePartnerStatus

Prefisso del servizio	Azioni
redshift-data	dati redshift: BatchExecuteStatement dati redshift: CancelStatement dati redshift: DescribeStatement dati redshift: DescribeTable dati redshift: ExecuteStatement dati redshift: GetStatementResult dati redshift: ListDatabases dati redshift: ListSchemas dati redshift: ListSessions dati redshift: ListStatements dati redshift: ListTables

Prefisso del servizio	Azioni
refactor-spaces	spazi refactorici: CreateApplication spazi di refattore: CreateEnvironment spazi di refattore: CreateRoute spazi di refattore: CreateService spazi di refattore: DeleteApplication spazi di refattore: DeleteEnvironment spazi di refattore: DeleteResourcePolicy spazi di refattore: DeleteRoute spazi di refattore: DeleteService spazi di refattore: GetApplication spazi di refattore: GetEnvironment spazi di refattore: GetResourcePolicy spazi di refattore: GetRoute spazi di refattore: GetService spazi di refattore: ListApplications spazi di refattore: ListEnvironmentVpcs spazi di refattore: ListEnvironments spazi di refattore: ListRoutes spazi di refattore: ListServices spazi di refattore: PutResourcePolicy spazi di refattore: UpdateRoute

Prefisso del servizio	Azioni
rekognition	riconoscimento: AssociateFaces
	riconoscimento: CompareFaces
	riconoscimento: CopyProjectVersion
	riconoscimento: CreateCollection
	riconoscimento: CreateDataset
	riconoscimento: CreateFaceLivenessSession
	riconoscimento: CreateProject
	riconoscimento: CreateProjectVersion
	riconoscimento: CreateStreamProcessor
	riconoscimento: CreateUser
	riconoscimento: DeleteCollection
	riconoscimento: DeleteDataset
	riconoscimento: DeleteFaces
	riconoscimento: DeleteProject
	riconoscimento: DeleteProjectPolicy
	riconoscimento: DeleteProjectVersion
	riconoscimento: DeleteStreamProcessor
	riconoscimento: DeleteUser
	riconoscimento: DescribeCollection
	riconoscimento: DescribeDataset
	riconoscimento: DescribeProjectVersions

Prefisso del servizio	Azioni
	riconoscimento: DescribeProjects
	riconoscimento: DescribeStreamProcessor
	riconoscimento: DetectCustomLabels
	riconoscimento: DetectFaces
	riconoscimento: DetectLabels
	riconoscimento: DetectModerationLabels
	riconoscimento: DetectProtectiveEquipment
	riconoscimento: DetectText
	riconoscimento: DisassociateFaces
	riconoscimento: DistributeDatasetEntries
	riconoscimento: GetCelebrityInfo
	riconoscimento: GetCelebrityRecognition
	riconoscimento: GetContentModeration
	riconoscimento: GetFaceDetection
	riconoscimento: GetFaceLivenessSessionResults
	riconoscimento: GetFaceSearch
	riconoscimento: GetLabelDetection
	riconoscimento: GetMediaAnalysisJob
	riconoscimento: GetPersonTracking
	riconoscimento: GetSegmentDetection
	riconoscimento: GetTextDetection

Prefisso del servizio	Azioni
	riconoscimento: IndexFaces
	riconoscimento: ListCollections
	riconoscimento: ListDatasetEntries
	riconoscimento: ListDatasetLabels
	riconoscimento: ListFaces
	riconoscimento: ListMediaAnalysisJobs
	riconoscimento: ListProjectPolicies
	riconoscimento: ListStreamProcessors
	riconoscimento: ListUsers
	riconoscimento: PutProjectPolicy
	riconoscimento: RecognizeCelebrities
	riconoscimento: SearchFaces
	riconoscimento: SearchFacesByImage
	riconoscimento: SearchUsers
	riconoscimento: SearchUsersByImage
	riconoscimento: StartCelebrityRecognition
	riconoscimento: StartContentModeration
	riconoscimento: StartFaceDetection
	riconoscimento: StartFaceLivenessSession
	riconoscimento: StartFaceSearch
	riconoscimento: StartLabelDetection

Prefisso del servizio	Azioni
	riconoscimento: StartMediaAnalysisJob
	riconoscimento: StartPersonTracking
	riconoscimento: StartProjectVersion
	riconoscimento: StartSegmentDetection
	riconoscimento: StartStreamProcessor
	riconoscimento: StartTextDetection
	riconoscimento: StopProjectVersion
	riconoscimento: StopStreamProcessor
	riconoscimento: UpdateDatasetEntries
	riconoscimento: UpdateStreamProcessor

Prefisso del servizio	Azioni
resiliencehub	hub di resilienza: AcceptResourceGroupingRecommendations hub di resilienza: AddDraftAppVersionResourceMappings hub di resilienza: BatchUpdateRecommendationStatus hub di resilienza: CreateApp hub di resilienza: CreateAppVersionAppComponent hub di resilienza: CreateAppVersionResource hub di resilienza: CreateRecommendationTemplate hub di resilienza: CreateResiliencyPolicy hub di resilienza: DeleteApp hub di resilienza: DeleteAppAssessment hub di resilienza: DeleteAppInputSource hub di resilienza: DeleteAppVersionAppComponent hub di resilienza: DeleteAppVersionResource hub di resilienza: DeleteRecommendationTemplate hub di resilienza: DeleteResiliencyPolicy hub di resilienza: DescribeApp hub di resilienza: DescribeAppAssessment hub di resilienza: DescribeAppVersion hub di resilienza: DescribeAppVersionAppComponent hub di resilienza: DescribeAppVersionResource hub di resilienza: DescribeAppVersionResourcesResolutionStatus

Prefisso del servizio	Azioni
	hub di resilienza: DescribeAppVersionTemplate hub di resilienza: DescribeDraftAppVersionResourcesImportStatus hub di resilienza: DescribeMetricsExport hub di resilienza: DescribeResiliencyPolicy hub di resilienza: DescribeResourceGroupingRecommendationTask hub di resilienza: ImportResourcesToDraftAppVersion hub di resilienza: ListAlarmRecommendations hub di resilienza: ListAppAssessmentComplianceDrifts hub di resilienza: ListAppAssessmentResourceDrifts hub di resilienza: ListAppAssessments hub di resilienza: ListAppComponentCompliances hub di resilienza: ListAppComponentRecommendations hub di resilienza: ListAppInputSources hub di resilienza: ListAppVersionAppComponents hub di resilienza: ListAppVersionResourceMappings hub di resilienza: ListAppVersionResources hub di resilienza: ListAppVersions hub di resilienza: ListApps hub di resilienza: ListMetrics hub di resilienza: ListRecommendationTemplates hub di resilienza: ListResiliencyPolicies

Prefisso del servizio	Azioni
	hub di resilienza: ListResourceGroupingRecommendations hub di resilienza: ListSopRecommendations hub di resilienza: ListSuggestedResiliencyPolicies hub di resilienza: ListTestRecommendations hub di resilienza: ListUnsupportedAppVersionResources hub di resilienza: PublishAppVersion hub di resilienza: PutDraftAppVersionTemplate hub di resilienza: RejectResourceGroupingRecommendations hub di resilienza: RemoveDraftAppVersionResourceMappings hub di resilienza: ResolveAppVersionResources hub di resilienza: StartAppAssessment hub di resilienza: StartResourceGroupingRecommendationTask hub di resilienza: UpdateApp hub di resilienza: UpdateAppVersion hub di resilienza: UpdateAppVersionAppComponent hub di resilienza: UpdateAppVersionResource hub di resilienza: UpdateResiliencyPolicy

Prefisso del servizio	Azioni
resource-explorer-2	resource-explorer-2: AssociateDefaultView
	esploratore-risorsa-2: BatchGetView
	esploratore-risorsa-2: CreateIndex
	esploratore-risorsa-2: CreateResourceExplorerSetup
	esploratore-risorsa-2: CreateView
	esploratore-risorsa-2: DeleteIndex
	esploratore-risorsa-2: DeleteResourceExplorerSetup
	esploratore-risorsa-2: DeleteView
	esploratore-risorsa-2: DisassociateDefaultView
	esploratore-risorsa-2: GetAccountLevelServiceConfiguration
	esploratore-risorsa-2: GetDefaultView
	esploratore-risorsa-2: GetIndex
	esploratore-risorsa-2: GetManagedView
	esploratore-risorsa-2: GetResourceExplorerSetup
	esploratore-risorsa-2: GetServiceIndex
	esploratore-risorsa-2: GetServiceView
	esploratore-risorsa-2: ListIndexes
	esploratore-risorsa-2: ListIndexesForMembers
	esploratore-risorsa-2: ListManagedViews
	esploratore-risorsa-2: ListServiceIndexes
	esploratore-risorsa-2: ListServiceViews

Prefisso del servizio	Azioni
	esploratore-risorsa-2: ListStreamingAccessForServices
	esploratore-risorsa-2: ListSupportedResourceTypes
	esploratore-risorsa-2: ListViews
	resource-explorer-2:Search
	esploratore-risorsa-2: UpdateIndexType
	esploratore-risorsa-2: UpdateView

Prefisso del servizio	Azioni
resource-groups	gruppi di risorse: CancelTagSyncTask gruppi di risorse: GetAccountSettings gruppi di risorse: GetGroup gruppi di risorse: GetGroupConfiguration gruppi di risorse: GetGroupQuery gruppi di risorse: GetTagSyncTask gruppi di risorse: GroupResources gruppi di risorse: ListGroupResources gruppi di risorse: ListGroupingStatuses gruppi di risorse: ListGroups gruppi di risorse: ListTagSyncTasks gruppi di risorse: PutGroupConfiguration gruppi di risorse: SearchResources gruppi di risorse: StartTagSyncTask gruppi di risorse: UngroupResources gruppi di risorse: UpdateAccountSettings gruppi di risorse: UpdateGroup gruppi di risorse: UpdateGroupQuery

Prefisso del servizio	Azioni
robomaker	robomaker: BatchDeleteWorlds
	robomaker: BatchDescribeSimulationJob
	robomaker: CancelDeploymentJob
	robomaker: CancelSimulationJob
	robomaker: CancelSimulationJobBatch
	robomaker: CancelWorldExportJob
	robomaker: CancelWorldGenerationJob
	robomaker: CreateDeploymentJob
	robomaker: CreateFleet
	robomaker: CreateRobot
	robomaker: CreateRobotApplication
	robomaker: CreateRobotApplicationVersion
	robomaker: CreateSimulationApplication
	robomaker: CreateSimulationApplicationVersion
	robomaker: CreateSimulationJob
	robomaker: CreateWorldExportJob
	robomaker: CreateWorldGenerationJob
	robomaker: CreateWorldTemplate
	robomaker: DeleteFleet
	robomaker: DeleteRobot
	robomaker: DeleteRobotApplication

Prefisso del servizio	Azioni
	robomaker: DeleteSimulationApplication
	robomaker: DeleteWorldTemplate
	robomaker: DeregisterRobot
	robomaker: DescribeDeploymentJob
	robomaker: DescribeFleet
	robomaker: DescribeRobot
	robomaker: DescribeRobotApplication
	robomaker: DescribeSimulationApplication
	robomaker: DescribeSimulationJob
	robomaker: DescribeSimulationJobBatch
	robomaker: DescribeWorld
	robomaker: DescribeWorldExportJob
	robomaker: DescribeWorldGenerationJob
	robomaker: DescribeWorldTemplate
	robomaker: GetWorldTemplateBody
	robomaker: ListDeploymentJobs
	robomaker: ListFleets
	robomaker: ListRobotApplications
	robomaker: ListRobots
	robomaker: ListSimulationApplications
	robomaker: ListSimulationJobBatches

Prefisso del servizio	Azioni
	robomaker: ListSimulationJobs
	robomaker: ListWorldExportJobs
	robomaker: ListWorldGenerationJobs
	robomaker: ListWorldTemplates
	robomaker: ListWorlds
	robomaker: RegisterRobot
	robomaker: RestartSimulationJob
	robomaker: StartSimulationJobBatch
	robomaker: SyncDeploymentJob
	robomaker: UpdateRobotApplication
	robomaker: UpdateSimulationApplication
	robomaker: UpdateWorldTemplate

Prefisso del servizio	Azioni
rolesanywhere	ruoli ovunque: CreateProfile
	ruoli ovunque: CreateTrustAnchor
	ruoli ovunque: DeleteAttributeMapping
	ruoli ovunque: DeleteCrl
	ruoli ovunque: DeleteProfile
	ruoli ovunque: DeleteTrustAnchor
	ruoli ovunque: DisableCrl
	ruoli ovunque: DisableProfile
	ruoli ovunque: DisableTrustAnchor
	ruoli ovunque: EnableCrl
	ruoli ovunque: EnableProfile
	ruoli ovunque: EnableTrustAnchor
	ruoli ovunque: GetCrl
	ruoli ovunque: GetProfile
	ruoli ovunque: GetSubject
	ruoli ovunque: GetTrustAnchor
	ruoli ovunque: ImportCrl
	ruoli ovunque: ListCrls
	ruoli ovunque: ListProfiles
	ruoli ovunque: ListSubjects
	ruoli ovunque: ListTrustAnchors

Prefisso del servizio	Azioni
	ruoli ovunque: PutAttributeMapping ruoli ovunque: PutNotificationSettings ruoli ovunque: ResetNotificationSettings ruoli ovunque: UpdateCrl ruoli ovunque: UpdateProfile ruoli ovunque: UpdateTrustAnchor

Prefisso del servizio	Azioni
route53	percorso 53: ActivateKeySigningKey
	percorso 53: AssociateVPCWithHostedZone
	percorso 53: ChangeCidrCollection
	percorso 53: ChangeResourceRecordSets
	percorso 53: CreateCidrCollection
	percorso 53: CreateHealthCheck
	percorso 53: CreateHostedZone
	percorso 53: CreateKeySigningKey
	percorso 53: CreateQueryLoggingConfig
	percorso 53: CreateReusableDelegationSet
	percorso 53: CreateTrafficPolicy
	percorso 53: CreateTrafficPolicyInstance
	percorso 53: CreateTrafficPolicyVersion
	percorso 53: CreateVPCAssociationAuthorization
	percorso 53: DeactivateKeySigningKey
	percorso 53: DeleteCidrCollection
	percorso 53: DeleteHealthCheck
	percorso 53: DeleteHostedZone
	percorso 53: DeleteKeySigningKey
	percorso 53: DeleteQueryLoggingConfig
	percorso 53: DeleteReusableDelegationSet

Prefisso del servizio	Azioni
	percorso 53: DeleteTrafficPolicy
	percorso 53: DeleteTrafficPolicyInstance
	percorso 53: DeleteVPCAssociationAuthorization
	percorso 53: DisableHostedZoneDNSSEC
	percorso 53: DisassociateVPCFromHostedZone
	percorso 53: EnableHostedZoneDNSSEC
	percorso 53: GetAccountLimit
	percorso 53: GetChange
	percorso 53: GetCheckerIpRanges
	route53:GetDNSSEC
	percorso 53: GetGeoLocation
	percorso 53: GetHealthCheck
	percorso 53: GetHealthCheckCount
	percorso 53: GetHealthCheckLastFailureReason
	percorso 53: GetHealthCheckStatus
	percorso 53: GetHostedZone
	percorso 53: GetHostedZoneCount
	percorso 53: GetHostedZoneLimit
	percorso 53: GetQueryLoggingConfig
	percorso 53: GetReusableDelegationSet
	percorso 53: GetReusableDelegationSetLimit

Prefisso del servizio	Azioni
	percorso 53: GetTrafficPolicy
	percorso 53: GetTrafficPolicyInstance
	percorso 53: GetTrafficPolicyInstanceCount
	percorso 53: ListCidrBlocks
	percorso 53: ListCidrCollections
	percorso 53: ListCidrLocations
	percorso 53: ListGeoLocations
	percorso 53: ListHealthChecks
	percorso 53: ListHostedZones
	percorso 53: ListHostedZonesByName
	percorso 53: ListHostedZonesByVPC
	percorso 53: ListQueryLoggingConfigs
	percorso 53: ListResourceRecordSets
	percorso 53: ListReusableDelegationSets
	percorso 53: ListTrafficPolicies
	percorso 53: ListTrafficPolicyInstances
	percorso 53: ListTrafficPolicyInstancesByHostedZone
	percorso 53: ListTrafficPolicyInstancesByPolicy
	percorso 53: ListTrafficPolicyVersions
	percorso 53: ListVPCAssociationAuthorizations
	route53:TestDNSAnswer

Prefisso del servizio	Azioni
	percorso 53: UpdateHealthCheck
	percorso 53: UpdateHostedZoneComment
	percorso 53: UpdateTrafficPolicyComment
	percorso 53: UpdateTrafficPolicyInstance

Prefisso del servizio	Azioni
route53-recovery-control-config	route53-recovery-control-config: CreateCluster route53-recovery-control-config: CreateControlPanel route53-recovery-control-config: CreateRoutingControl route53-recovery-control-config: CreateSafetyRule route53-recovery-control-config: DeleteCluster route53-recovery-control-config: DeleteControlPanel route53-recovery-control-config: DeleteRoutingControl route53-recovery-control-config: DeleteSafetyRule route53-recovery-control-config: DescribeCluster route53-recovery-control-config: DescribeControlPanel route53-recovery-control-config: DescribeRoutingControl route53-recovery-control-config: DescribeSafetyRule route53-recovery-control-config: GetResourcePolicy route53-recovery-control-config: ListAssociatedRoute53HealthChecks route53-recovery-control-config: ListClusters route53-recovery-control-config: ListControlPanels route53-recovery-control-config: ListRoutingControls route53-recovery-control-config: ListSafetyRules route53-recovery-control-config: UpdateCluster route53-recovery-control-config: UpdateControlPanel

Prefisso del servizio	Azioni
	route53-recovery-control-config: UpdateRoutingControl route53-recovery-control-config: UpdateSafetyRule

Prefisso del servizio	Azioni
route53-recovery-readiness	route53-prontezza per il ripristino: CreateCell
	predisposizione al ripristino del route53: CreateCrossAccount Authorization
	predisposizione al ripristino del route53: CreateReadinessCheck
	predisposizione al ripristino del route53: CreateRecoveryGroup
	predisposizione al ripristino del route53: CreateResourceSet
	predisposizione al ripristino del route53: DeleteCell
	predisposizione al ripristino del route53: DeleteCrossAccount Authorization
	predisposizione al ripristino del route53: DeleteReadinessCheck
	predisposizione al ripristino del route53: DeleteRecoveryGroup
	predisposizione al ripristino del route53: DeleteResourceSet
	predisposizione al ripristino del route53: GetArchitectureRecommendations
	predisposizione al ripristino del route53: GetCell
	predisposizione al ripristino del route53: GetCellReadinessSummary
	predisposizione al ripristino del route53: GetReadinessCheck
	predisposizione al ripristino del route53: GetReadinessCheckResourceStatus
	predisposizione al ripristino del route53: GetReadinessCheckStatus
	predisposizione al ripristino del route53: GetRecoveryGroup

Prefisso del servizio	Azioni
	predisposizione al ripristino del route53: GetRecoveryGroupReadinessSummary
	predisposizione al ripristino del route53: GetResourceSet
	predisposizione al ripristino del route53: ListCells
	predisposizione al ripristino del route53: ListCrossAccountAuthorizations
	predisposizione al ripristino del route53: ListReadinessChecks
	predisposizione al ripristino del route53: ListRecoveryGroups
	predisposizione al ripristino del route53: ListResourceSets
	predisposizione al ripristino del route53: ListRules
	predisposizione al ripristino del route53: UpdateCell
	predisposizione al ripristino del route53: UpdateReadinessCheck
	predisposizione al ripristino del route53: UpdateRecoveryGroup
	predisposizione al ripristino del route53: UpdateResourceSet

Prefisso del servizio	Azioni
route53resolver	resolver route53: AssociateFirewallRuleGroup resolver route53: AssociateResolverEndpointIpAddress resolver route53: AssociateResolverQueryLogConfig resolver route53: AssociateResolverRule resolver route53: CreateFirewallDomainList resolver route53: CreateFirewallRule resolver route53: CreateFirewallRuleGroup resolver route53: CreateResolverEndpoint resolver route53: CreateResolverQueryLogConfig resolver route53: CreateResolverRule resolver route53: DeleteFirewallDomainList resolver route53: DeleteFirewallRule resolver route53: DeleteFirewallRuleGroup resolver route53: DeleteOutpostResolver resolver route53: DeleteResolverEndpoint resolver route53: DeleteResolverQueryLogConfig resolver route53: DeleteResolverRule resolver route53: DisassociateFirewallRuleGroup resolver route53: DisassociateResolverEndpointIpAddress resolver route53: DisassociateResolverQueryLogConfig resolver route53: DisassociateResolverRule

Prefisso del servizio	Azioni
	resolver route53: GetFirewallConfig resolver route53: GetFirewallDomainList resolver route53: GetFirewallRuleGroup resolver route53: GetFirewallRuleGroupAssociation resolver route53: GetFirewallRuleGroupPolicy resolver route53: GetOutpostResolver resolver route53: GetResolverConfig resolver route53: GetResolverDnssecConfig resolver route53: GetResolverEndpoint resolver route53: GetResolverQueryLogConfig resolver route53: GetResolverQueryLogConfigAssociation resolver route53: GetResolverQueryLogConfigPolicy resolver route53: GetResolverRule resolver route53: GetResolverRuleAssociation resolver route53: GetResolverRulePolicy resolver route53: ImportFirewallDomains resolver route53: ListFirewallConfigs resolver route53: ListFirewallDomainLists resolver route53: ListFirewallDomains resolver route53: ListFirewallRuleGroupAssociations resolver route53: ListFirewallRuleGroups

Prefisso del servizio	Azioni
	resolver route53: ListFirewallRuleTypes resolver route53: ListFirewallRules resolver route53: ListOutpostResolvers resolver route53: ListResolverConfigs resolver route53: ListResolverDnssecConfigs resolver route53: ListResolverEndpointIpAddresses resolver route53: ListResolverEndpoints resolver route53: ListResolverQueryLogConfigAssociations resolver route53: ListResolverQueryLogConfigs resolver route53: ListResolverRuleAssociations resolver route53: ListResolverRules resolver route53: PutFirewallRuleGroupPolicy resolver route53: PutResolverQueryLogConfigPolicy resolver route53: UpdateFirewallConfig resolver route53: UpdateFirewallDomains resolver route53: UpdateFirewallRule resolver route53: UpdateFirewallRuleGroupAssociation resolver route53: UpdateOutpostResolver resolver route53: UpdateResolverConfig resolver route53: UpdateResolverDnssecConfig resolver route53: UpdateResolverEndpoint

Prefisso del servizio	Azioni
	resolver route53: UpdateResolverRule
rum	tamburo: BatchCreateRumMetricDefinitions rum: BatchDeleteRumMetricDefinitions rum: BatchGetRumMetricDefinitions rum: CreateAppMonitor rum: DeleteAppMonitor rum: DeleteResourcePolicy rum: DeleteRumMetricsDestination rum: GetAppMonitor rum: GetAppMonitorData rum: GetResourcePolicy rum: ListAppMonitors rum: ListRumMetricsDestinations rum: PutResourcePolicy rum: PutRumMetricsDestination rum: UpdateAppMonitor rum: UpdateRumMetricDefinition

Prefisso del servizio	Azioni
s3	s3: AssociateAccessGrantsIdentityCenter s3: CreateAccessGrant s3: CreateAccessGrantsInstance s3: CreateAccessGrantsLocation s3: CreateAccessPoint s3: CreateAccessPointForObjectLambda s3: CreateBucket s3: CreateBucketMetadataTableConfiguration s3: CreateJob s3: CreateMultiRegionAccessPoint s3: DeleteAccessGrant s3: DeleteAccessGrantsInstance s3: DeleteAccessGrantsInstanceResourcePolicy s3: DeleteAccessGrantsLocation s3: DeleteAccessPoint s3: DeleteAccessPointForObjectLambda s3: DeleteAccessPointPolicy s3: DeleteAccessPointPolicyForObjectLambda s3: DeleteBucket s3: DeleteBucketMetadataTableConfiguration s3: DeleteBucketPolicy

Prefisso del servizio	Azioni
	s3: DeleteBucketWebsite s3: DeleteMultiRegionAccessPoint s3: DeleteStorageLensConfiguration s3: DescribeJob s3: DescribeMultiRegionAccessPointOperation s3: DissociateAccessGrantsIdentityCenter s3: GetAccelerateConfiguration s3: GetAccessGrant s3: GetAccessGrantsInstance s3: GetAccessGrantsInstanceForPrefix s3: GetAccessGrantsInstanceResourcePolicy s3: GetAccessGrantsLocation s3: GetAccessPoint s3: GetAccessPointConfigurationForObjectLambda s3: GetAccessPointForObjectLambda s3: GetAccessPointPolicy s3: GetAccessPointPolicyForObjectLambda s3: GetAccessPointPolicyStatus s3: GetAccessPointPolicyStatusForObjectLambda s3: GetAccountPublicAccessBlock s3: GetAnalyticsConfiguration

Prefisso del servizio	Azioni
	s3: GetBucketAbac s3: GetBucketAcl s3: NUCLEO GetBucket s3: GetBucketLocation s3: GetBucketLogging s3: GetBucketNotification s3: GetBucketObjectLockConfiguration s3: GetBucketOwnershipControls s3: GetBucketPolicy s3: GetBucketPolicyStatus s3: GetBucketPublicAccessBlock s3: GetBucketRequestPayment s3: GetBucketVersioning s3: GetBucketWebsite s3: GetDataAccess s3: GetEncryptionConfiguration s3: GetIntelligentTieringConfiguration s3: GetInventoryConfiguration s3: GetLifecycleConfiguration s3: GetMetricsConfiguration s3: GetMultiRegionAccessPoint

Prefisso del servizio	Azioni
	s3: GetMultiRegionAccessPointPolicy s3: GetMultiRegionAccessPointPolicyStatus s3: GetMultiRegionAccessPointRoutes s3: GetReplicationConfiguration s3: GetStorageLensConfiguration s3: GetStorageLensDashboard s3: ListAccessGrants s3: ListAccessGrantsInstances s3: ListAccessGrantsLocations s3: ListAccessPoints s3: ListAccessPointsForObjectLambda s3: ListAllMyBuckets s3: ListBucketMultipartUploads s3: ListCallerAccessGrants s3: ListJobs s3: ListMultiRegionAccessPoints s3: ListStorageLensConfigurations s3: PutAccelerateConfiguration s3: PutAccessGrantsInstanceResourcePolicy s3: PutAccessPointConfigurationForObjectLambda s3: PutAccessPointPolicy

Prefisso del servizio	Azioni
	s3: PutAccessPointPolicyForObjectLambda s3: PutAccountPublicAccessBlock s3: PutAnalyticsConfiguration s3: PutBucketAbac s3: PutBucketAcl s3: NUCLEO PutBucket s3: PutBucketLogging s3: PutBucketNotification s3: PutBucketObjectLockConfiguration s3: PutBucketOwnershipControls s3: PutBucketPolicy s3: PutBucketPublicAccessBlock s3: PutBucketRequestPayment s3: PutBucketVersioning s3: PutBucketWebsite s3: PutEncryptionConfiguration s3: PutIntelligentTieringConfiguration s3: PutInventoryConfiguration s3: PutLifecycleConfiguration s3: PutMetricsConfiguration s3: PutMultiRegionAccessPointPolicy

Prefisso del servizio	Azioni
	s3: PutReplicationConfiguration s3: PutStorageLensConfiguration s3: SubmitMultiRegionAccessPointRoutes s3: UpdateAccessGrantsLocation s3: UpdateBucketMetadataAnnotationTableConfiguration s3: UpdateBucketMetadataJournalTableConfiguration s3: UpdateJobPriority s3: UpdateJobStatus
s3-outposts	avamposti s3: CreateEndpoint avamposti s3: DeleteEndpoint avamposti s3: ListEndpoints avamposti s3: ListOutpostsWithS3 avamposti s3: ListSharedEndpoints

Prefisso del servizio	Azioni
sagemaker-geospatial	sagemaker geospaziale: DeleteEarthObservationJob sagemaker geospaziale: DeleteVectorEnrichmentJob sagemaker geospaziale: ExportEarthObservationJob sagemaker geospaziale: ExportVectorEnrichmentJob sagemaker geospaziale: GetEarthObservationJob sagemaker geospaziale: GetRasterDataCollection sagemaker geospaziale: GetTile sagemaker geospaziale: GetVectorEnrichmentJob sagemaker geospaziale: ListEarthObservationJobs sagemaker geospaziale: ListRasterDataCollections sagemaker geospaziale: ListVectorEnrichmentJobs sagemaker geospaziale: SearchRasterDataCollection sagemaker geospaziale: StartEarthObservationJob sagemaker geospaziale: StartVectorEnrichmentJob sagemaker geospaziale: StopEarthObservationJob sagemaker geospaziale: StopVectorEnrichmentJob

Prefisso del servizio	Azioni
savingsplans	piani di risparmio: CreateSavingsPlan
	piani di risparmio: DeleteQueuedSavingsPlan
	piani di risparmio: DescribeSavingsPlanRates
	piani di risparmio: DescribeSavingsPlans
	piani di risparmio: DescribeSavingsPlansOfferingRates
	piani di risparmio: DescribeSavingsPlansOfferings
	piani di risparmio: ReturnSavingsPlan

Prefisso del servizio	Azioni
schemas	schemi: CreateDiscoverer schemi: CreateRegistry schemi: CreateSchema schemi: DeleteDiscoverer schemi: DeleteRegistry schemi: DeleteResourcePolicy schemi: DeleteSchema schemi: DeleteSchemaVersion schemi: DescribeCodeBinding schemi: DescribeDiscoverer schemi: DescribeRegistry schemi: DescribeSchema schemi: ExportSchema schemi: GetCodeBindingSource schemi: GetDiscoveredSchema schemi: GetResourcePolicy schemi: ListDiscoverers schemi: ListRegistries schemi: ListSchemaVersions schemi: ListSchemas schemi: PutCodeBinding

Prefisso del servizio	Azioni
	schemi: PutResourcePolicy schemi: SearchSchemas schemi: StartDiscoverer schemi: StopDiscoverer schemi: UpdateDiscoverer schemi: UpdateRegistry schemi: UpdateSchema
sdb	sdb: CreateDomain sdb: DeleteDomain sdb: DomainMetadata sdb: ListDomains

Prefisso del servizio	Azioni
secretsmanager	gestore dei segreti: CancelRotateSecret gestore dei segreti: CreateSecret gestore dei segreti: DeleteResourcePolicy gestore dei segreti: DeleteSecret gestore dei segreti: DescribeSecret gestore dei segreti: GetRandomPassword gestore dei segreti: GetResourcePolicy gestore dei segreti: GetSecretValue gestore dei segreti: ListSecretVersionIds gestore dei segreti: ListSecrets gestore dei segreti: PutResourcePolicy gestore dei segreti: PutSecretValue gestore dei segreti: RemoveRegionsFromReplication gestore dei segreti: ReplicateSecretToRegions gestore dei segreti: RestoreSecret gestore dei segreti: RotateSecret gestore dei segreti: StopReplicationToReplica gestore dei segreti: UpdateSecret gestore dei segreti: ValidateResourcePolicy

Prefisso del servizio	Azioni
securityhub	hub di sicurezza: AcceptAdministratorInvitation
	hub di sicurezza: AcceptInvitation
	hub di sicurezza: BatchDeleteAutomationRules
	hub di sicurezza: BatchDisableStandards
	hub di sicurezza: BatchEnableStandards
	hub di sicurezza: BatchGetAutomationRules
	hub di sicurezza: BatchGetConfigurationPolicyAssociations
	hub di sicurezza: BatchGetEnabledRegionsV2
	hub di sicurezza: BatchGetSecurityControls
	hub di sicurezza: BatchGetStandardsControlAssociations
	hub di sicurezza: BatchImportFindings
	hub di sicurezza: BatchUpdateAutomationRules
	hub di sicurezza: BatchUpdateFindings
	hub di sicurezza: BatchUpdateStandardsControlAssociations
	hub di sicurezza: V2 ConnectorRegistrations
	hub di sicurezza: CreateActionTarget
	hub di sicurezza: V2 CreateAggregator
	hub di sicurezza: CreateAutomationRule
	hub di sicurezza: CreateAutomationRuleV2
	hub di sicurezza: CreateConfigurationPolicy
	hub di sicurezza: CreateConnector

Prefisso del servizio	Azioni
	hub di sicurezza: V2 CreateConnector
	hub di sicurezza: CreateFindingAggregator
	hub di sicurezza: CreateInsight
	hub di sicurezza: CreateMembers
	hub di sicurezza: V2 CreateTicket
	hub di sicurezza: DeclineInvitations
	hub di sicurezza: DeleteActionTarget
	hub di sicurezza: V2 DeleteAggregator
	hub di sicurezza: DeleteAutomationRuleV2
	hub di sicurezza: DeleteConfigurationPolicy
	hub di sicurezza: DeleteConnector
	hub di sicurezza: V2 DeleteConnector
	hub di sicurezza: DeleteFindingAggregator
	hub di sicurezza: DeleteInsight
	hub di sicurezza: DeleteInvitations
	hub di sicurezza: DeleteMembers
	hub di sicurezza: DescribeActionTargets
	hub di sicurezza: DescribeHub
	hub di sicurezza: DescribeOrganizationConfiguration
	hub di sicurezza: DescribeProducts
	hub di sicurezza: DescribeSecurityHubV2

Prefisso del servizio	Azioni
	hub di sicurezza: DescribeStandards
	hub di sicurezza: DisableImportFindingsForProduct
	hub di sicurezza: DisableOrganizationAdminAccount
	hub di sicurezza: DisableSecurityHub
	hub di sicurezza: DisableSecurityHubFeatureV2
	hub di sicurezza: DisableSecurityHubV2
	hub di sicurezza: DisassociateFromAdministratorAccount
	hub di sicurezza: DisassociateFromMasterAccount
	hub di sicurezza: DisassociateMembers
	hub di sicurezza: EnableImportFindingsForProduct
	hub di sicurezza: EnableOrganizationAdminAccount
	hub di sicurezza: EnableSecurityHub
	hub di sicurezza: EnableSecurityHubFeatureV2
	hub di sicurezza: GetAdministratorAccount
	hub di sicurezza: V2 GetAggregator
	hub di sicurezza: GetAutomationRuleV2
	hub di sicurezza: GetConfigurationPolicy
	hub di sicurezza: GetConfigurationPolicyAssociation
	hub di sicurezza: GetConnector
	hub di sicurezza: V2 GetConnector
	hub di sicurezza: GetCoverageStatisticsV2

Prefisso del servizio	Azioni
	hub di sicurezza: GetEnabledStandards
	hub di sicurezza: GetFindingAggregator
	hub di sicurezza: GetFindingHistory
	hub di sicurezza: GetFindings
	hub di sicurezza: GetInsightResults
	hub di sicurezza: GetInsights
	hub di sicurezza: GetInvitationsCount
	hub di sicurezza: GetMasterAccount
	hub di sicurezza: GetMembers
	hub di sicurezza: GetSecurityControlDefinition
	hub di sicurezza: InviteMembers
	hub di sicurezza: V2 ListAggregators
	hub di sicurezza: ListAutomationRules
	hub di sicurezza: ListAutomationRulesV2
	hub di sicurezza: ListConfigurationPolicies
	hub di sicurezza: ListConfigurationPolicyAssociations
	hub di sicurezza: ListConnectors
	hub di sicurezza: V2 ListConnectors
	hub di sicurezza: ListEnabledProductsForImport
	hub di sicurezza: ListFindingAggregators
	hub di sicurezza: ListInvitations

Prefisso del servizio	Azioni
	hub di sicurezza: ListMembers
	hub di sicurezza: ListOrganizationAdminAccounts
	hub di sicurezza: ListSecurityControlDefinitions
	hub di sicurezza: ListStandardsControlAssociations
	hub di sicurezza: StartConfigurationPolicyAssociation
	hub di sicurezza: StartConfigurationPolicyDisassociation
	hub di sicurezza: UpdateActionTarget
	hub di sicurezza: V2 UpdateAggregator
	hub di sicurezza: UpdateAutomationRuleV2
	hub di sicurezza: UpdateConfigurationPolicy
	hub di sicurezza: UpdateConnector
	hub di sicurezza: V2 UpdateConnector
	hub di sicurezza: UpdateFindingAggregator
	hub di sicurezza: UpdateFindings
	hub di sicurezza: UpdateInsight
	hub di sicurezza: UpdateOrganizationConfiguration
	hub di sicurezza: UpdateSecurityControl
	hub di sicurezza: UpdateSecurityHubConfiguration

Prefisso del servizio	Azioni
securitylake	lago di sicurezza: CreateAwsLogSource lago di sicurezza: CreateCustomLogSource lago di sicurezza: CreateDataLakeExceptionSubscription lago di sicurezza: CreateDataLakeOrganizationConfiguration lago di sicurezza: CreateSubscriber lago di sicurezza: CreateSubscriberNotification lago di sicurezza: DeleteAwsLogSource lago di sicurezza: DeleteCustomLogSource lago di sicurezza: DeleteDataLakeExceptionSubscription lago di sicurezza: DeleteDataLakeOrganizationConfiguration lago di sicurezza: DeleteSubscriber lago di sicurezza: DeleteSubscriberNotification lago di sicurezza: DeregisterDataLakeDelegatedAdministrator lago di sicurezza: GetDataLakeExceptionSubscription lago di sicurezza: GetDataLakeOrganizationConfiguration lago di sicurezza: GetDataLakeSources lago di sicurezza: GetSubscriber lago di sicurezza: ListDataLakes lago di sicurezza: ListLogSources lago di sicurezza: ListSubscribers lago di sicurezza: RegisterDataLakeDelegatedAdministrator

Prefisso del servizio	Azioni
	lago di sicurezza: UpdateDataLakeExceptionSubscription lago di sicurezza: UpdateSubscriber lago di sicurezza: UpdateSubscriberNotification
serverlessrepo	repository senza server: CreateApplication repository senza server: CreateApplicationVersion repository senza server: CreateCloudFormationChangeSet repository senza server: CreateCloudFormationTemplate repository senza server: DeleteApplication repository senza server: GetApplication repository senza server: GetApplicationPolicy repository senza server: GetCloudFormationTemplate repository senza server: ListApplicationDependencies repository senza server: ListApplicationVersions repository senza server: ListApplications repository senza server: PutApplicationPolicy repository senza server: UnshareApplication repository senza server: UpdateApplication

Prefisso del servizio	Azioni
servicecatalog	catalogo dei servizi: AcceptPortfolioShare catalogo dei servizi: AssociateBudgetWithResource catalogo dei servizi: AssociatePrincipalWithPortfolio catalogo dei servizi: AssociateProductWithPortfolio catalogo dei servizi: AssociateServiceActionWithProvisioningArtifact catalogo dei servizi: BatchAssociateServiceActionWithProvisioningArtifact catalogo dei servizi: BatchDisassociateServiceActionFromProvisioningArtifact catalogo dei servizi: CopyProduct catalogo dei servizi: CreateAttributeGroup catalogo dei servizi: CreateConstraint catalogo dei servizi: CreatePortfolio catalogo dei servizi: CreatePortfolioShare catalogo dei servizi: CreateProduct catalogo dei servizi: CreateProvisionedProductPlan catalogo dei servizi: CreateProvisioningArtifact catalogo dei servizi: CreateServiceAction catalogo dei servizi: DeleteAttributeGroup catalogo dei servizi: DeleteConstraint catalogo dei servizi: DeletePortfolio catalogo dei servizi: DeletePortfolioShare

Prefisso del servizio	Azioni
	catalogo dei servizi: DeleteProduct catalogo dei servizi: DeleteProvisionedProductPlan catalogo dei servizi: DeleteProvisioningArtifact catalogo dei servizi: DeleteServiceAction catalogo dei servizi: DescribeConstraint catalogo dei servizi: DescribeCopyProductStatus catalogo dei servizi: DescribePortfolio catalogo dei servizi: DescribePortfolioShareStatus catalogo dei servizi: DescribePortfolioShares catalogo dei servizi: DescribeProduct catalogo dei servizi: DescribeProductAsAdmin catalogo dei servizi: DescribeProductView catalogo dei servizi: DescribeProvisionedProduct catalogo dei servizi: DescribeProvisionedProductPlan catalogo dei servizi: DescribeProvisioningArtifact catalogo dei servizi: DescribeProvisioningParameters catalogo dei servizi: DescribeRecord catalogo dei servizi: DescribeServiceAction catalogo dei servizi: DescribeServiceActionExecutionParameters catalogo dei servizi: DisableAWSOrganizationsAccess catalogo dei servizi: DisassociateBudgetFromResource

Prefisso del servizio	Azioni
	catalogo dei servizi: DisassociatePrincipalFromPortfolio catalogo dei servizi: DisassociateProductFromPortfolio catalogo dei servizi: DisassociateServiceActionFromProvisioningArtifact catalogo dei servizi: EnableAWSOrganizationsAccess catalogo dei servizi: ExecuteProvisionedProductPlan catalogo dei servizi: ExecuteProvisionedProductServiceAction catalogo dei servizi: GetAWSOrganizationsAccessStatus catalogo dei servizi: GetProvisionedProductOutputs catalogo dei servizi: ImportAsProvisionedProduct catalogo dei servizi: ListAcceptedPortfolioShares catalogo dei servizi: ListAttributeGroups catalogo dei servizi: ListBudgetsForResource catalogo dei servizi: ListConstraintsForPortfolio catalogo dei servizi: ListLaunchPaths catalogo dei servizi: ListOrganizationPortfolioAccess catalogo dei servizi: ListPortfolioAccess catalogo dei servizi: ListPortfolios catalogo dei servizi: ListPortfoliosForProduct catalogo dei servizi: ListPrincipalsForPortfolio catalogo dei servizi: ListProvisionedProductPlans

Prefisso del servizio	Azioni
	catalogo dei servizi: ListProvisioningArtifacts catalogo dei servizi: ListProvisioningArtifactsForServiceAction catalogo dei servizi: ListRecordHistory catalogo dei servizi: ListServiceActions catalogo dei servizi: ListServiceActionsForProvisioningArtifact catalogo dei servizi: ListStackInstancesForProvisionedProduct catalogo dei servizi: NotifyProvisionProductEngineWorkflowResult catalogo dei servizi: NotifyTerminateProvisionedProductEngineWorkflowResult catalogo dei servizi: NotifyUpdateProvisionedProductEngineWorkflowResult catalogo dei servizi: ProvisionProduct catalogo dei servizi: RejectPortfolioShare catalogo dei servizi: ScanProvisionedProducts catalogo dei servizi: SearchProducts catalogo dei servizi: SearchProductsAsAdmin catalogo dei servizi: SearchProvisionedProducts catalogo dei servizi: TerminateProvisionedProduct catalogo dei servizi: UpdateConstraint catalogo dei servizi: UpdatePortfolio catalogo dei servizi: UpdatePortfolioShare catalogo dei servizi: UpdateProduct

Prefisso del servizio	Azioni
	catalogo dei servizi: UpdateProvisionedProduct catalogo dei servizi: UpdateProvisionedProductProperties catalogo dei servizi: UpdateProvisioningArtifact catalogo dei servizi: UpdateServiceAction

Prefisso del servizio	Azioni
servicediscovery	individuazione dei servizi: CreateHttpNamespace
	individuazione del servizio: CreatePrivateDnsNamespace
	individuazione del servizio: CreatePublicDnsNamespace
	individuazione del servizio: CreateService
	individuazione del servizio: DeleteNamespace
	individuazione del servizio: DeleteService
	individuazione del servizio: DeleteServiceAttributes
	individuazione del servizio: DeregisterInstance
	individuazione del servizio: GetInstance
	individuazione del servizio: GetInstancesHealthStatus
	individuazione del servizio: GetNamespace
	individuazione del servizio: GetOperation
	individuazione del servizio: GetService
	individuazione del servizio: ListInstances
	individuazione del servizio: ListNamespaces
	individuazione del servizio: ListOperations
	individuazione del servizio: ListServices
	individuazione del servizio: RegisterInstance
	individuazione del servizio: UpdateHttpNamespace
	individuazione del servizio: UpdateInstanceCustomHealthStatus
	individuazione del servizio: UpdatePrivateDnsNamespace

Prefisso del servizio	Azioni
	individuazione del servizio: UpdatePublicDnsNamespace individuazione del servizio: UpdateService individuazione del servizio: UpdateServiceAttributes

Prefisso del servizio	Azioni
servicequotas	quote di servizio: AssociateServiceQuotaTemplate quote di servizio: CreateSupportCase quote di servizio: DeleteServiceQuotaIncreaseRequestFromTemplate quote di servizio: DisassociateServiceQuotaTemplate quote di servizio: GetAWSDefaultServiceQuota quote di servizio: GetAssociationForServiceQuotaTemplate quote di servizio: GetAutoManagementConfiguration quote di servizio: GetQuotaUtilizationReport quote di servizio: GetRequestedServiceQuotaChange quote di servizio: GetServiceQuota quote di servizio: GetServiceQuotaIncreaseRequestFromTemplate quote di servizio: ListAWSDefaultServiceQuotas quote di servizio: ListRequestedServiceQuotaChangeHistory quote di servizio: ListRequestedServiceQuotaChangeHistoryByQuota quote di servizio: ListServiceQuotaIncreaseRequestsInTemplate quote di servizio: ListServiceQuotas quote di servizio: ListServices quote di servizio: PutServiceQuotaIncreaseRequestIntoTemplate quote di servizio: RequestServiceQuotaIncrease quote di servizio: StartAutoManagement

Prefisso del servizio	Azioni
	quote di servizio: StartQuotaUtilizationReport quote di servizio: StopAutoManagement quote di servizio: UpdateAutoManagement

Prefisso del servizio	Azioni
ses	usi: BatchGetMetricData
	usa: CloneReceiptRuleSet
	usa: CreateAddonInstance
	usa: CreateAddonSubscription
	usa: CreateAddressList
	usa: CreateAddressListImportJob
	usa: CreateArchive
	usa: CreateConfigurationSet
	usa: CreateConfigurationSetEventDestination
	usa: CreateConfigurationSetTrackingOptions
	usa: CreateContact
	usa: CreateContactList
	usa: CreateCustomVerificationEmailTemplate
	usa: CreateDedicatedIpPool
	usa: CreateDeliverabilityTestReport
	usa: CreateEmailIdentity
	usa: CreateEmailIdentityPolicy
	usa: CreateEmailTemplate
	usa: CreateImportJob
	usa: CreateIngressPoint
	usa: CreateMultiRegionEndpoint

Prefisso del servizio	Azioni
	usa: CreateReceiptFilter
	usa: CreateReceiptRule
	usa: CreateReceiptRuleSet
	usa: CreateRelay
	usa: CreateRuleSet
	usa: CreateTemplate
	usa: CreateTenant
	usa: CreateTenantResourceAssociation
	usa: CreateTrafficPolicy
	usa: DeleteAddonInstance
	usa: DeleteAddonSubscription
	usa: DeleteAddressList
	usa: DeleteArchive
	usa: DeleteConfigurationSet
	usa: DeleteConfigurationSetEventDestination
	usa: DeleteConfigurationSetTrackingOptions
	usa: DeleteContact
	usa: DeleteContactList
	usa: DeleteCustomVerificationEmailTemplate
	usa: DeleteDedicatedIpPool
	usa: DeleteEmailIdentity

Prefisso del servizio	Azioni
	usa: DeleteEmailIdentityPolicy
	usa: DeleteEmailTemplate
	usa: DeleteIdentity
	usa: DeleteIdentityPolicy
	usa: DeleteIngressPoint
	usa: DeleteMultiRegionEndpoint
	usa: DeleteReceiptFilter
	usa: DeleteReceiptRule
	usa: DeleteReceiptRuleSet
	usa: DeleteRelay
	usa: DeleteRuleSet
	usa: DeleteSuppressedDestination
	usa: DeleteTemplate
	usa: DeleteTenant
	usa: DeleteTenantResourceAssociation
	usa: DeleteTrafficPolicy
	usa: DeleteVerifiedEmailAddress
	usa: DeregisterMemberFromAddressList
	usa: DescribeActiveReceiptRuleSet
	usa: DescribeConfigurationSet
	usa: DescribeReceiptRule

Prefisso del servizio	Azioni
	usa: DescribeReceiptRuleSet
	usa: GetAccount
	usa: GetAccountSendingEnabled
	usa: GetAddonInstance
	usa: GetAddonSubscription
	usa: GetAddressList
	usa: GetArchive
	usa: GetArchiveExport
	usa: GetArchiveMessage
	usa: GetArchiveMessageContent
	usa: GetArchiveSearch
	usa: GetArchiveSearchResults
	usa: GetBlacklistReports
	usa: GetConfigurationSet
	usa: GetConfigurationSetEventDestinations
	usa: GetContact
	usa: GetContactList
	usa: GetCustomVerificationEmailTemplate
	usa: GetDedicatedIp
	usa: GetDedicatedIpPool
	usa: GetDedicatedIps

Prefisso del servizio	Azioni
	usa: GetDeliverabilityDashboardOptions
	usa: GetDeliverabilityTestReport
	usa: GetDomainDeliverabilityCampaign
	usa: GetDomainStatisticsReport
	usa: GetEmailAddressInsights
	usa: GetEmailIdentity
	usa: GetEmailIdentityPolicies
	usa: GetEmailTemplate
	usa: GetIdentityDkimAttributes
	usa: GetIdentityMailFromDomainAttributes
	usa: GetIdentityNotificationAttributes
	usa: GetIdentityPolicies
	usa: GetIdentityVerificationAttributes
	usa: GetImportJob
	usa: GetIngressPoint
	usa: GetMemberOfAddressList
	usa: GetMessageInsights
	usa: GetMultiRegionEndpoint
	usa: GetRelay
	usa: GetRuleSet
	usa: GetSendQuota

Prefisso del servizio	Azioni
	usa: GetSendStatistics
	usa: GetSuppressedDestination
	usa: GetTemplate
	usa: GetTenant
	usa: GetTrafficPolicy
	usa: ListAddonInstances
	usa: ListAddonSubscriptions
	usa: ListAddressListImportJobs
	usa: ListAddressLists
	usa: ListArchiveExports
	usa: ListArchiveSearches
	usa: ListArchives
	usa: ListConfigurationSets
	usa: ListContactLists
	usa: ListContacts
	usa: ListCustomVerificationEmailTemplates
	usa: ListDedicatedIpPools
	usa: ListDeliverabilityTestReports
	usa: ListDomainDeliverabilityCampaigns
	usa: ListEmailIdentities
	usa: ListEmailTemplates

Prefisso del servizio	Azioni
	usa: ListExportJobs
	usa: ListIdentities
	usa: ListIdentityPolicies
	usa: ListImportJobs
	usa: ListIngressPoints
	usa: ListMembersOfAddressList
	usa: ListMultiRegionEndpoints
	usa: ListReceiptFilters
	usa: ListReceiptRuleSets
	usa: ListRecommendations
	usa: ListRelays
	usa: ListReputationEntities
	usa: ListResourceTenants
	usa: ListRuleSets
	usa: ListSuppressedDestinations
	usa: ListTemplates
	usa: ListTenantResources
	usa: ListTenants
	usa: ListTrafficPolicies
	usa: ListVerifiedEmailAddresses
	usa: PutAccountDedicatedIpWarmupAttributes

Prefisso del servizio	Azioni
	usa: PutAccountDetails
	usa: PutAccountSendingAttributes
	usa: PutAccountSuppressionAttributes
	usa: PutAccountVdmAttributes
	usa: PutConfigurationSetArchivingOptions
	usa: PutConfigurationSetDeliveryOptions
	usa: PutConfigurationSetReputationOptions
	usa: PutConfigurationSetSendingOptions
	usa: PutConfigurationSetSuppressionOptions
	usa: PutConfigurationSetTrackingOptions
	usa: PutConfigurationSetVdmOptions
	usa: PutDedicatedIpInPool
	usa: PutDedicatedIpPoolScalingAttributes
	usa: PutDedicatedIpWarmupAttributes
	usa: PutDeliverabilityDashboardOption
	usa: PutEmailIdentityConfigurationSetAttributes
	usa: PutEmailIdentityDkimAttributes
	usa: PutEmailIdentityDkimSigningAttributes
	usa: PutEmailIdentityFeedbackAttributes
	usa: PutEmailIdentityMailFromAttributes
	usa: PutIdentityPolicy

Prefisso del servizio	Azioni
	usa: PutSuppressedDestination
	usa: PutTenantSuppressionAttributes
	usa: RegisterMemberToAddressList
	usa: ReorderReceiptRuleSet
	usa: SendBounce
	usa: SendCustomVerificationEmail
	usa: SetActiveReceiptRuleSet
	usa: SetIdentityDkimEnabled
	usa: SetIdentityFeedbackForwardingEnabled
	usa: SetIdentityHeadersInNotificationsEnabled
	usa: SetIdentityMailFromDomain
	usa: SetIdentityNotificationTopic
	usa: SetReceiptRulePosition
	usa: StartArchiveExport
	usa: StartArchiveSearch
	usa: StopArchiveExport
	usa: StopArchiveSearch
	usa: TestRenderEmailTemplate
	usa: TestRenderTemplate
	usa: UpdateAccountSendingEnabled
	usa: UpdateArchive

Prefisso del servizio	Azioni
	usa: UpdateConfigurationSetEventDestination
	usa: UpdateConfigurationSetReputationMetricsEnabled
	usa: UpdateConfigurationSetSendingEnabled
	usa: UpdateConfigurationSetTrackingOptions
	usa: UpdateContact
	usa: UpdateContactList
	usa: UpdateCustomVerificationEmailTemplate
	usa: UpdateEmailIdentityPolicy
	usa: UpdateEmailTemplate
	usa: UpdateIngressPoint
	usa: UpdateReceiptRule
	usa: UpdateRelay
	usa: UpdateRuleSet
	usa: UpdateTemplate
	usa: UpdateTrafficPolicy
	usa: VerifyDomainDkim
	usa: VerifyDomainIdentity
	usa: VerifyEmailAddress
	usa: VerifyEmailIdentity

Prefisso del servizio	Azioni
shield	scudo: AssociateDRTLogBucket scudo: AssociateHealthCheck scudo: AssociateProactiveEngagementDetails scudo: CreateProtection scudo: CreateProtectionGroup scudo: CreateSubscription scudo: DeleteProtection scudo: DeleteProtectionGroup scudo: DeleteSubscription scudo: DescribeAttack scudo: DescribeAttackStatistics shield:DescribeDRTAccess scudo: DescribeEmergencyContactSettings scudo: DescribeProtection scudo: DescribeProtectionGroup scudo: DescribeSubscription scudo: DisableApplicationLayerAutomaticResponse scudo: DisableProactiveEngagement scudo: DisassociateDRTLogBucket shield:DisassociateDRTRole scudo: DisassociateHealthCheck

Prefisso del servizio	Azioni
	scudo: EnableApplicationLayerAutomaticResponse scudo: EnableProactiveEngagement scudo: GetSubscriptionState scudo: ListAttacks scudo: ListProtectionGroups scudo: ListProtections scudo: ListResourcesInProtectionGroup scudo: UpdateApplicationLayerAutomaticResponse scudo: UpdateEmergencyContactSettings scudo: UpdateProtectionGroup scudo: UpdateSubscription

Prefisso del servizio	Azioni
signer	firmatario: AddProfilePermission
	firmatario: CancelSigningProfile
	firmatario: DescribeSigningJob
	firmatario: GetSigningPlatform
	firmatario: GetSigningProfile
	firmatario: ListProfilePermissions
	firmatario: ListSigningJobs
	firmatario: ListSigningPlatforms
	firmatario: ListSigningProfiles
	firmatario: PutSigningProfile
	firmatario: RemoveProfilePermission
	firmatario: RevokeSignature
	firmatario: RevokeSigningProfile
	firmatario: SignPayload
	firmatario: StartSigningJob

Prefisso del servizio	Azioni
simspaceweaver	simspaceweaver: CreateSnapshot simspaceweaver: DeleteApp simspaceweaver: DeleteSimulation simspaceweaver: DescribeApp simspaceweaver: DescribeSimulation simspaceweaver: ListApps simspaceweaver: ListSimulations simspaceweaver: StartApp simspaceweaver: StartClock simspaceweaver: StartSimulation simspaceweaver: StopApp simspaceweaver: StopClock simspaceweaver: StopSimulation

Prefisso del servizio	Azioni
sms	sms: CreateApp
	sms: CreateReplicationJob
	sms: DeleteApp
	sms: DeleteAppLaunchConfiguration
	sms: DeleteAppReplicationConfiguration
	sms: DeleteAppValidationConfiguration
	sms: DeleteReplicationJob
	sms: DeleteServerCatalog
	sms: DisassociateConnector
	sms: GenerateChangeSet
	sms: GenerateTemplate
	sms: GetApp
	sms: GetAppLaunchConfiguration
	sms: GetAppReplicationConfiguration
	sms: GetAppValidationConfiguration
	sms: GetAppValidationOutput
	sms: GetConnectors
	sms: GetReplicationJobs
	sms: GetReplicationRuns
	sms: GetServers
	sms: ImportAppCatalog

Prefisso del servizio	Azioni
	sms: ImportServerCatalog
	sms: LaunchApp
	sms: ListApps
	sms: NotifyAppValidationOutput
	sms: PutAppLaunchConfiguration
	sms: PutAppReplicationConfiguration
	sms: PutAppValidationConfiguration
	sms: StartAppReplication
	sms: StartOnDemandAppReplication
	sms: StartOnDemandReplicationRun
	sms: StopAppReplication
	sms: TerminateApp
	sms: UpdateApp
	sms: UpdateReplicationJob

Prefisso del servizio	Azioni
sms-voice	sms vocale: AssociateProtectConfiguration messaggio vocale via sms: CreateConfigurationSet messaggio vocale via sms: CreateConfigurationSetEventDestination messaggio vocale via sms: CreateEventDestination messaggio vocale via sms: CreateOptOutList messaggio vocale via sms: CreatePool messaggio vocale via sms: CreateProtectConfiguration messaggio vocale via sms: CreateRegistration messaggio vocale via sms: CreateRegistrationAssociation messaggio vocale via sms: CreateRegistrationAttachment messaggio vocale via sms: CreateRegistrationVersion messaggio vocale via sms: CreateVerifiedDestinationNumber messaggio vocale via sms: DeleteAccountDefaultProtectConfiguration messaggio vocale via sms: DeleteConfigurationSet messaggio vocale via sms: DeleteConfigurationSetEventDestination messaggio vocale via sms: DeleteDefaultMessageType messaggio vocale via sms: DeleteDefaultSenderId messaggio vocale via sms: DeleteEventDestination messaggio vocale via sms: DeleteKeyword

Prefisso del servizio	Azioni
	messaggio vocale via sms: DeleteMediaMessageSpendLimitOverride messaggio vocale via sms: DeleteOptOutList messaggio vocale via sms: DeleteOptedOutNumber messaggio vocale via sms: DeletePool messaggio vocale via sms: DeleteProtectConfiguration messaggio vocale via sms: DeleteProtectConfigurationRuleSetNumberOverride messaggio vocale via sms: DeleteRegistration messaggio vocale via sms: DeleteRegistrationAttachment messaggio vocale via sms: DeleteResourcePolicy messaggio vocale via sms: DeleteTextMessageSpendLimitOverride messaggio vocale via sms: DeleteVerifiedDestinationNumber messaggio vocale via sms: DeleteVoiceMessageSpendLimitOverride messaggio vocale via sms: DescribeAccountAttributes messaggio vocale via sms: DescribeAccountLimits messaggio vocale via sms: DescribeConfigurationSets messaggio vocale via sms: DescribeKeywords messaggio vocale via sms: DescribeOptOutLists messaggio vocale via sms: DescribeOptedOutNumbers messaggio vocale via sms: DescribePhoneNumbers

Prefisso del servizio	Azioni
	messaggio vocale via sms: DescribePools messaggio vocale via sms: DescribeProtectConfigurations messaggio vocale via sms: DescribeRegistrationAttachments messaggio vocale via sms: DescribeRegistrationFieldDefinitions messaggio vocale via sms: DescribeRegistrationFieldValues messaggio vocale via sms: DescribeRegistrationSectionDefinitions messaggio vocale via sms: DescribeRegistrationTypeDefinitions messaggio vocale via sms: DescribeRegistrationVersions messaggio vocale via sms: DescribeRegistrations messaggio vocale via sms: DescribeSenderIds messaggio vocale via sms: DescribeSpendLimits messaggio vocale via sms: DescribeVerifiedDestinationNumbers messaggio vocale via sms: DisassociateOriginationIdentity messaggio vocale via sms: DisassociateProtectConfiguration messaggio vocale via sms: DiscardRegistrationVersion messaggio vocale via sms: GetConfigurationSetEventDestinations messaggio vocale via sms: GetProtectConfigurationCountryRuleSet messaggio vocale via sms: GetResourcePolicy messaggio vocale via sms: ListConfigurationSets messaggio vocale via sms: ListPoolOriginationIdentities

Prefisso del servizio	Azioni
	messaggio vocale via sms: ListProtectConfigurationRuleSetNumberOverrides messaggio vocale via sms: ListRegistrationAssociations messaggio vocale via sms: PutKeyword messaggio vocale via sms: PutOptedOutNumber messaggio vocale via sms: PutProtectConfigurationRuleSetNumberOverride messaggio vocale via sms: PutResourcePolicy messaggio vocale via sms: ReleasePhoneNumber messaggio vocale via sms: ReleaseSenderId messaggio vocale via sms: RequestPhoneNumber messaggio vocale via sms: RequestSenderId messaggio vocale via sms: SendDestinationNumberVerificationCode messaggio vocale via sms: SetAccountDefaultProtectConfiguration messaggio vocale via sms: SetDefaultMessageFeedbackEnabled messaggio vocale via sms: SetDefaultMessageType messaggio vocale via sms: SetDefaultSenderId messaggio vocale via sms: SetMediaMessageSpendLimitOverride messaggio vocale via sms: SetTextMessageSpendLimitOverride messaggio vocale via sms: SetVoiceMessageSpendLimitOverride messaggio vocale via sms: SubmitRegistrationVersion

Prefisso del servizio	Azioni
	messaggio vocale via sms: UpdateConfigurationSetEventDestination messaggio vocale via sms: UpdateEventDestination messaggio vocale via sms: UpdatePhoneNumber messaggio vocale via sms: UpdatePool messaggio vocale via sms: UpdateProtectConfiguration messaggio vocale via sms: UpdateProtectConfigurationCountryRuleSet messaggio vocale via sms: UpdateSenderId

Prefisso del servizio	Azioni
snowball	palla di neve: CancelCluster
	palla di neve: CancelJob
	palla di neve: CreateAddress
	palla di neve: CreateCluster
	palla di neve: CreateJob
	palla di neve: CreateLongTermPricing
	palla di neve: CreateReturnShippingLabel
	palla di neve: DescribeAddress
	palla di neve: DescribeAddresses
	palla di neve: DescribeCluster
	palla di neve: DescribeJob
	palla di neve: DescribeReturnShippingLabel
	palla di neve: GetJobManifest
	palla di neve: GetJobUnlockCode
	palla di neve: GetSnowballUsage
	palla di neve: GetSoftwareUpdates
	palla di neve: ListClusterJobs
	palla di neve: ListClusters
	palla di neve: ListCompatibleImages
	palla di neve: ListJobs
	palla di neve: ListLongTermPricing

Prefisso del servizio	Azioni
	palla di neve: ListPickupLocations palla di neve: ListServiceVersions palla di neve: UpdateCluster palla di neve: UpdateJob palla di neve: UpdateJobShipmentState palla di neve: UpdateLongTermPricing
sqs	seghe: AddPermission seggioloni: CancelMessageMoveTask seggioloni: CreateQueue seggioloni: DeleteQueue seggioloni: PurgeQueue seggioloni: RemovePermission seggioloni: SetQueueAttributes

Prefisso del servizio	Azioni
ssm	ssm: AssociateOpsItemRelatedItem
	ssm: CancelCommand
	ssm: CancelMaintenanceWindowExecution
	ssm: CreateActivation
	ssm: CreateAssociation
	ssm: CreateAssociationBatch
	ssm: CreateDocument
	ssm: CreateMaintenanceWindow
	ssm: CreateOpsItem
	ssm: CreateOpsMetadata
	ssm: CreatePatchBaseline
	ssm: CreateResourceDataSync
	ssm: DeleteActivation
	ssm: DeleteAssociation
	ssm: DeleteCloudConnector
	ssm: DeleteDocument
	ssm: DeleteInventory
	ssm: DeleteMaintenanceWindow
	ssm: DeleteOpsItem
	ssm: DeleteOpsMetadata
	ssm: DeleteParameter

Prefisso del servizio	Azioni
	ssm: DeleteParameters
	ssm: DeletePatchBaseline
	ssm: DeleteResourceDataSync
	ssm: DeleteResourcePolicy
	ssm: DeregisterManagedInstance
	ssm: DeregisterPatchBaselineForPatchGroup
	ssm: DeregisterTargetFromMaintenanceWindow
	ssm: DeregisterTaskFromMaintenanceWindow
	ssm: DescribeActivations
	ssm: DescribeAssociation
	ssm: DescribeAssociationExecutionTargets
	ssm: DescribeAssociationExecutions
	ssm: DescribeAutomationExecutions
	ssm: DescribeAutomationStepExecutions
	ssm: DescribeAvailablePatches
	ssm: DescribeDocument
	ssm: DescribeDocumentParameters
	ssm: DescribeDocumentPermission
	ssm: DescribeEffectiveInstanceAssociations
	ssm: DescribeEffectivePatchesForPatchBaseline
	ssm: DescribeInstanceAssociationsStatus

Prefisso del servizio	Azioni
	ssm: DescribeInstanceInformation
	ssm: DescribeInstancePatchStates
	ssm: DescribeInstancePatchStatesForPatchGroup
	ssm: DescribeInstancePatches
	ssm: DescribeInstanceProperties
	ssm: DescribeInventoryDeletions
	ssm: DescribeMaintenanceWindowExecutionTaskInvocations
	ssm: DescribeMaintenanceWindowExecutionTasks
	ssm: DescribeMaintenanceWindowExecutions
	ssm: DescribeMaintenanceWindowSchedule
	ssm: DescribeMaintenanceWindowTargets
	ssm: DescribeMaintenanceWindowTasks
	ssm: DescribeMaintenanceWindows
	ssm: DescribeMaintenanceWindowsForTarget
	ssm: DescribeOpsItems
	ssm: DescribeParameters
	ssm: DescribePatchBaselines
	ssm: DescribePatchGroupState
	ssm: DescribePatchGroups
	ssm: DescribePatchProperties
	ssm: DescribeSessions

Prefisso del servizio	Azioni
	ssm: DisassociateOpsItemRelatedItem
	ssm: GetAccessToken
	ssm: GetAutomationExecution
	ssm: GetCalendarState
	ssm: GetCloudConnector
	ssm: GetCommandInvocation
	ssm: GetConnectionStatus
	ssm: GetDefaultPatchBaseline
	ssm: GetDeployablePatchSnapshotForInstance
	ssm: GetDocument
	ssm: GetExecutionPreview
	ssm: GetInventory
	ssm: GetInventorySchema
	ssm: GetMaintenanceWindow
	ssm: GetMaintenanceWindowExecution
	ssm: GetMaintenanceWindowExecutionTask
	ssm: GetMaintenanceWindowExecutionTaskInvocation
	ssm: GetMaintenanceWindowTask
	ssm: GetOpsItem
	ssm: GetOpsMetadata
	ssm: GetOpsSummary

Prefisso del servizio	Azioni
	ssm: GetParameter
	ssm: GetParameterHistory
	ssm: GetParameters
	ssm: GetParametersByPath
	ssm: GetPatchBaseline
	ssm: GetPatchBaselineForPatchGroup
	ssm: GetResourcePolicies
	ssm: GetServiceSetting
	ssm: LabelParameterVersion
	ssm: ListAssociationVersions
	ssm: ListAssociations
	ssm: ListCloudConnectors
	ssm: ListCommandInvocations
	ssm: ListCommands
	ssm: ListComplianceItems
	ssm: ListComplianceSummaries
	ssm: ListDocumentMetadataHistory
	ssm: ListDocumentVersions
	ssm: ListDocuments
	ssm: ListInstanceAssociations
	ssm: ListInventoryEntries

Prefisso del servizio	Azioni
	ssm: ListNodes
	ssm: ListNodesSummary
	ssm: ListOpsItemEvents
	ssm: ListOpsItemRelatedItems
	ssm: ListOpsMetadata
	ssm: ListResourceComplianceSummaries
	ssm: ListResourceDataSync
	ssm: ModifyDocumentPermission
	ssm: PutComplianceItems
	ssm: PutInventory
	ssm: PutParameter
	ssm: PutResourcePolicy
	ssm: RegisterDefaultPatchBaseline
	ssm: RegisterManagedInstance
	ssm: RegisterPatchBaselineForPatchGroup
	ssm: RegisterTargetWithMaintenanceWindow
	ssm: RegisterTaskWithMaintenanceWindow
	ssm: ResetServiceSetting
	ssm: ResumeSession
	ssm: SendAutomationSignal
	ssm: SendCommand

Prefisso del servizio	Azioni
	ssm: StartAssociationsOnce
	ssm: StartAutomationExecution
	ssm: StartChangeRequestExecution
	ssm: StartSession
	ssm: StopAutomationExecution
	ssm: TerminateSession
	ssm: UnlabelParameterVersion
	ssm: UpdateAssociation
	ssm: UpdateAssociationStatus
	ssm: UpdateCloudConnector
	ssm: UpdateDocument
	ssm: UpdateDocumentDefaultVersion
	ssm: UpdateDocumentMetadata
	ssm: UpdateInstanceInformation
	ssm: UpdateMaintenanceWindow
	ssm: UpdateMaintenanceWindowTarget
	ssm: UpdateMaintenanceWindowTask
	ssm: UpdateManagedInstanceRole
	ssm: UpdateOpsItem
	ssm: UpdateOpsMetadata
	ssm: UpdatePatchBaseline

Prefisso del servizio	Azioni
	ssm: UpdateResourceDataSync ssm: UpdateServiceSetting ssm: ValidateCloudConnector

Prefisso del servizio	Azioni
ssm-incidents	incidenti ssm: BatchGetIncidentFindings
	incidenti ssm: CreateReplicationSet
	incidenti ssm: CreateResponsePlan
	incidenti ssm: CreateTimelineEvent
	incidenti ssm: DeleteIncidentRecord
	incidenti ssm: DeleteReplicationSet
	incidenti ssm: DeleteResourcePolicy
	incidenti ssm: DeleteResponsePlan
	incidenti ssm: DeleteTimelineEvent
	incidenti ssm: GetIncidentRecord
	incidenti ssm: GetReplicationSet
	incidenti ssm: GetResourcePolicies
	incidenti ssm: GetResponsePlan
	incidenti ssm: GetTimelineEvent
	incidenti ssm: ListIncidentFindings
	incidenti ssm: ListIncidentRecords
	incidenti ssm: ListRelatedItems
	incidenti ssm: ListReplicationSets
	incidenti ssm: ListResponsePlans
	incidenti ssm: ListTimelineEvents
	incidenti ssm: PutResourcePolicy

Prefisso del servizio	Azioni
	incidenti ssm: StartIncident incidenti ssm: UpdateDeletionProtection incidenti ssm: UpdateIncidentRecord incidenti ssm: UpdateRelatedItems incidenti ssm: UpdateReplicationSet incidenti ssm: UpdateResponsePlan incidenti ssm: UpdateTimelineEvent

Prefisso del servizio	Azioni
ssm-sap	ssm-sap: BackupDatabase
	ssm-sap: DeleteResourcePermission
	ssm-sap: DeregisterApplication
	ssm-sap: GetApplication
	ssm-sap: GetComponent
	ssm-sap: GetConfigurationCheckOperation
	ssm-sap: GetDatabase
	ssm-sap: GetOperation
	ssm-sap: GetResourcePermission
	ssm-sap: ListApplications
	ssm-sap: ListComponents
	ssm-sap: ListConfigurationCheckDefinitions
	ssm-sap: ListConfigurationCheckOperations
	ssm-sap: ListDatabases
	ssm-sap: ListOperationEvents
	ssm-sap: ListOperations
	ssm-sap: ListSubCheckResults
	ssm-sap: ListSubCheckRuleResults
	ssm-sap: PutResourcePermission
	ssm-sap: RegisterApplication
	ssm-sap: RestoreDatabase

Prefisso del servizio	Azioni
	ssm-sap: StartApplication ssm-sap: StartApplicationRefresh ssm-sap: StartConfigurationChecks ssm-sap: StopApplication ssm-sap: UpdateApplicationSettings ssm-sap: UpdateHANABackupSettings

Prefisso del servizio	Azioni
states	afferma: CreateActivity stati: CreateStateMachine stati: CreateStateMachineAlias stati: DeleteActivity stati: DeleteStateMachine stati: DeleteStateMachineAlias stati: DeleteStateMachineVersion stati: DescribeActivity stati: DescribeExecution stati: DescribeMapRun stati: DescribeStateMachine stati: DescribeStateMachineAlias stati: DescribeStateMachineForExecution stati: GetExecutionHistory stati: ListActivities stati: ListExecutions stati: ListMapRuns stati: ListStateMachineAliases stati: ListStateMachineVersions stati: ListStateMachines stati: SendTaskFailure

Prefisso del servizio	Azioni
	stati: SendTaskHeartbeat
	stati: SendTaskSuccess
	stati: StartExecution
	stati: StopExecution
	stati: UpdateMapRun
	stati: UpdateStateMachine
	stati: UpdateStateMachineAlias
	stati: ValidateStateMachineDefinition
sts	set: AssumeRole
	set: AssumeRoleWithSAML
	set: AssumeRoleWithWebIdentity
	set: DecodeAuthorizationMessage
	set: GetAccessKeyInfo
	set: GetCallerIdentity
	set: GetFederationToken
	set: GetSessionToken
	set: GetWebIdentityToken

Prefisso del servizio	Azioni
swf	swf: DeleteActivityType file swf: DeleteWorkflowType file swf: DeprecateActivityType file swf: DeprecateDomain file swf: DeprecateWorkflowType file swf: DescribeActivityType file swf: DescribeDomain file swf: DescribeWorkflowType file swf: ListActivityTypes file swf: ListDomains file swf: ListWorkflowTypes file swf: RegisterActivityType file swf: RegisterDomain file swf: RegisterWorkflowType file swf: UndeprecateActivityType file swf: UndeprecateDomain file swf: UndeprecateWorkflowType

Prefisso del servizio	Azioni
synthetics	sintetici: AssociateResource
	sintetici: CreateCanary
	sintetici: CreateGroup
	sintetici: DeleteCanary
	sintetici: DeleteGroup
	sintetici: DescribeCanaries
	sintetici: DescribeCanariesLastRun
	sintetici: DescribeRuntimeVersions
	sintetici: DisassociateResource
	sintetici: GetCanary
	sintetici: GetCanaryRuns
	sintetici: GetGroup
	sintetici: ListAssociatedGroups
	sintetici: ListGroupResources
	sintetici: ListGroups
	sintetici: StartCanary
	sintetici: StartCanaryDryRun
	sintetici: StopCanary
	sintetici: UpdateCanary

Prefisso del servizio	Azioni
tag	etichetta: DescribeReportCreation etichetta: GetComplianceSummary etichetta: GetResources etichetta: StartReportCreation

Prefisso del servizio	Azioni
textract	estratto: AnalyzeDocument
	estratto: AnalyzeExpense
	textract:AnalyzeID
	estratto: CreateAdapter
	estratto: CreateAdapterVersion
	estratto: DeleteAdapter
	estratto: DeleteAdapterVersion
	estratto: DetectDocumentText
	estratto: GetAdapter
	estratto: GetAdapterVersion
	estratto: GetDocumentAnalysis
	estratto: GetDocumentTextDetection
	estratto: GetExpenseAnalysis
	estratto: GetLendingAnalysis
	estratto: GetLendingAnalysisSummary
	estratto: ListAdapterVersions
	estratto: ListAdapters
	estratto: StartDocumentAnalysis
	estratto: StartDocumentTextDetection
	estratto: StartExpenseAnalysis
	estratto: StartLendingAnalysis

Prefisso del servizio	Azioni
	estratto: UpdateAdapter

Prefisso del servizio	Azioni
timestream	flusso temporale: CancelQuery flusso temporale: CreateDatabase flusso temporale: CreateScheduledQuery flusso temporale: CreateTable flusso temporale: DeleteDatabase flusso temporale: DeleteScheduledQuery flusso temporale: DeleteTable flusso temporale: DescribeAccountSettings flusso temporale: DescribeDatabase flusso temporale: DescribeScheduledQuery flusso temporale: DescribeTable flusso temporale: ExecuteScheduledQuery flusso temporale: ListBatchLoadTasks flusso temporale: ListDatabases flusso temporale: ListScheduledQueries flusso temporale: ListTables flusso temporale: PrepareQuery flusso temporale: UpdateAccountSettings flusso temporale: UpdateDatabase flusso temporale: UpdateScheduledQuery flusso temporale: UpdateTable

Prefisso del servizio	Azioni
tnb	tb: CancelSolNetworkOperation tnb: CreateSolFunctionPackage tnb: CreateSolNetworkInstance tnb: CreateSolNetworkPackage tnb: DeleteSolFunctionPackage tnb: DeleteSolNetworkInstance tnb: DeleteSolNetworkPackage tnb: GetSolFunctionInstance tnb: GetSolFunctionPackage tnb: GetSolFunctionPackageContent tnb: GetSolFunctionPackageDescriptor tnb: GetSolNetworkInstance tnb: GetSolNetworkOperation tnb: GetSolNetworkPackage tnb: GetSolNetworkPackageContent tnb: GetSolNetworkPackageDescriptor tnb: InstantiateSolNetworkInstance tnb: ListSolFunctionInstances tnb: ListSolFunctionPackages tnb: ListSolNetworkInstances tnb: ListSolNetworkOperations

Prefisso del servizio	Azioni
	tnb: ListSolNetworkPackages tnb: PutSolFunctionPackageContent tnb: PutSolNetworkPackageContent tnb: TerminateSolNetworkInstance tnb: UpdateSolFunctionPackage tnb: UpdateSolNetworkInstance tnb: UpdateSolNetworkPackage tnb: ValidateSolFunctionPackageContent tnb: ValidateSolNetworkPackageContent

Prefisso del servizio	Azioni
transcribe	trascrivere: CreateCallAnalyticsCategory trascrivere: CreateLanguageModel trascrivere: CreateMedicalVocabulary trascrivere: CreateVocabulary trascrivere: CreateVocabularyFilter trascrivere: DeleteCallAnalyticsCategory trascrivere: DeleteCallAnalyticsJob trascrivere: DeleteLanguageModel trascrivere: DeleteMedicalScribeJob trascrivere: DeleteMedicalTranscriptionJob trascrivere: DeleteMedicalVocabulary trascrivere: DeleteTranscriptionJob trascrivere: DeleteVocabulary trascrivere: DeleteVocabularyFilter trascrivere: DescribeLanguageModel trascrivere: GetCallAnalyticsCategory trascrivere: GetCallAnalyticsJob trascrivere: GetMedicalScribeJob trascrivere: GetMedicalTranscriptionJob trascrivere: GetMedicalVocabulary trascrivere: GetTranscriptionJob

Prefisso del servizio	Azioni
	trascrivere: GetVocabulary trascrivere: GetVocabularyFilter trascrivere: ListCallAnalyticsCategories trascrivere: ListCallAnalyticsJobs trascrivere: ListLanguageModels trascrivere: ListMedicalScribeJobs trascrivere: ListMedicalTranscriptionJobs trascrivere: ListMedicalVocabularies trascrivere: ListTranscriptionJobs trascrivere: ListVocabularies trascrivere: ListVocabularyFilters trascrivere: StartCallAnalyticsJob trascrivere: StartCallAnalyticsStreamTranscription trascrivere: StartCallAnalyticsStreamTranscriptionWebSocket trascrivere: StartMedicalScribeJob trascrivere: StartMedicalStreamTranscription trascrivere: StartMedicalStreamTranscriptionWebSocket trascrivere: StartMedicalTranscriptionJob trascrivere: StartStreamTranscription trascrivere: StartStreamTranscriptionWebSocket trascrivere: StartTranscriptionJob

Prefisso del servizio	Azioni
	trascrivere: UpdateCallAnalyticsCategory trascrivere: UpdateMedicalVocabulary trascrivere: UpdateVocabulary trascrivere: UpdateVocabularyFilter

Prefisso del servizio	Azioni
transfer	trasferimento: CreateAccess
	trasferimento: CreateAgreement
	trasferimento: CreateConnector
	trasferimento: CreateProfile
	trasferimento: CreateServer
	trasferimento: CreateUser
	trasferimento: CreateWebApp
	trasferimento: CreateWorkflow
	trasferimento: DeleteAccess
	trasferimento: DeleteAgreement
	trasferimento: DeleteCertificate
	trasferimento: DeleteConnector
	trasferimento: DeleteHostKey
	trasferimento: DeleteProfile
	trasferimento: DeleteServer
	trasferimento: DeleteSshPublicKey
	trasferimento: DeleteUser
	trasferimento: DeleteWebApp
	trasferimento: DeleteWebAppCustomization
	trasferimento: DeleteWorkflow
	trasferimento: DescribeAccess

Prefisso del servizio	Azioni
	trasferimento: DescribeAgreement
	trasferimento: DescribeCertificate
	trasferimento: DescribeConnector
	trasferimento: DescribeExecution
	trasferimento: DescribeHostKey
	trasferimento: DescribeProfile
	trasferimento: DescribeSecurityPolicy
	trasferimento: DescribeServer
	trasferimento: DescribeUser
	trasferimento: DescribeWebApp
	trasferimento: DescribeWebAppCustomization
	trasferimento: DescribeWorkflow
	trasferimento: ImportCertificate
	trasferimento: ImportHostKey
	trasferimento: ImportSshPublicKey
	trasferimento: ListAccesses
	trasferimento: ListCertificates
	trasferimento: ListConnectors
	trasferimento: ListExecutions
	trasferimento: ListFileTransferResults
	trasferimento: ListHostKeys

Prefisso del servizio	Azioni
	trasferimento: ListProfiles
	trasferimento: ListSecurityPolicies
	trasferimento: ListServers
	trasferimento: ListUsers
	trasferimento: ListWebApps
	trasferimento: ListWorkflows
	trasferimento: SendWorkflowStepState
	trasferimento: StartDirectoryListing
	trasferimento: StartFileTransfer
	trasferimento: StartRemoteDelete
	trasferimento: StartRemoteMove
	trasferimento: StartServer
	trasferimento: StopServer
	trasferimento: TestConnection
	trasferimento: TestIdentityProvider
	trasferimento: UpdateAccess
	trasferimento: UpdateAgreement
	trasferimento: UpdateCertificate
	trasferimento: UpdateConnector
	trasferimento: UpdateHostKey
	trasferimento: UpdateProfile

Prefisso del servizio	Azioni
	trasferimento: UpdateServer trasferimento: UpdateUser trasferimento: UpdateWebApp trasferimento: UpdateWebAppCustomization
translate	tradurre: CreateParallelData tradurre: DeleteParallelData tradurre: DeleteTerminology tradurre: DescribeTextTranslationJob tradurre: GetParallelData tradurre: GetTerminology tradurre: ImportTerminology tradurre: ListLanguages tradurre: ListParallelData tradurre: ListTerminologies tradurre: ListTextTranslationJobs tradurre: StartTextTranslationJob tradurre: StopTextTranslationJob tradurre: TranslateDocument tradurre: TranslateText tradurre: UpdateParallelData

Prefisso del servizio	Azioni
voiceid	voiceid: AssociateFraudster
	identificatore vocale: CreateDomain
	identificatore vocale: CreateWatchlist
	identificatore vocale: DeleteDomain
	identificatore vocale: DeleteFraudster
	identificatore vocale: DeleteSpeaker
	identificatore vocale: DeleteWatchlist
	identificatore vocale: DescribeDomain
	identificatore vocale: DescribeFraudster
	identificatore vocale: DescribeFraudsterRegistrationJob
	identificatore vocale: DescribeSpeaker
	identificatore vocale: DescribeSpeakerEnrollmentJob
	identificatore vocale: DescribeWatchlist
	identificatore vocale: DisassociateFraudster
	identificatore vocale: EvaluateSession
	identificatore vocale: ListDomains
	identificatore vocale: ListFraudsterRegistrationJobs
	identificatore vocale: ListFraudsters
	identificatore vocale: ListSpeakerEnrollmentJobs
	identificatore vocale: ListSpeakers
	identificatore vocale: ListWatchlists

Prefisso del servizio	Azioni
	identificatore vocale: OptOutSpeaker identificatore vocale: StartFraudsterRegistrationJob identificatore vocale: StartSpeakerEnrollmentJob identificatore vocale: UpdateDomain identificatore vocale: UpdateWatchlist

Prefisso del servizio	Azioni
vpc-lattice	reticolo vpc: CreateAccessLogSubscription reticolo vpc: CreateListener reticolo vpc: CreateResourceConfiguration reticolo vpc: CreateResourceGateway reticolo vpc: CreateRule reticolo vpc: CreateService reticolo vpc: CreateServiceNetwork reticolo vpc: CreateServiceNetworkResourceAssociation reticolo vpc: CreateServiceNetworkServiceAssociation reticolo vpc: CreateServiceNetworkVpcAssociation reticolo vpc: CreateTargetGroup reticolo vpc: DeleteAccessLogSubscription reticolo vpc: DeleteAuthPolicy reticolo vpc: DeleteDomainVerification reticolo vpc: DeleteListener reticolo vpc: DeleteResourceConfiguration reticolo vpc: DeleteResourceEndpointAssociation reticolo vpc: DeleteResourceGateway reticolo vpc: DeleteResourcePolicy reticolo vpc: DeleteRule reticolo vpc: DeleteService

Prefisso del servizio	Azioni
	reticolo vpc: DeleteServiceNetwork reticolo vpc: DeleteServiceNetworkResourceAssociation reticolo vpc: DeleteServiceNetworkServiceAssociation reticolo vpc: DeleteServiceNetworkVpcAssociation reticolo vpc: DeleteTargetGroup reticolo vpc: DeregisterTargets reticolo vpc: GetAccessLogSubscription reticolo vpc: GetAuthPolicy reticolo vpc: GetDomainVerification reticolo vpc: GetListener reticolo vpc: GetResourceConfiguration reticolo vpc: GetResourceGateway reticolo vpc: GetResourcePolicy reticolo vpc: GetRule reticolo vpc: GetService reticolo vpc: GetServiceNetwork reticolo vpc: GetServiceNetworkResourceAssociation reticolo vpc: GetServiceNetworkServiceAssociation reticolo vpc: GetServiceNetworkVpcAssociation reticolo vpc: GetTargetGroup reticolo vpc: ListAccessLogSubscriptions

Prefisso del servizio	Azioni
	reticolo vpc: ListDomainVerifications reticolo vpc: ListListeners reticolo vpc: ListResourceConfigurations reticolo vpc: ListResourceEndpointAssociations reticolo vpc: ListResourceGateways reticolo vpc: ListRules reticolo vpc: ListServiceNetworkResourceAssociations reticolo vpc: ListServiceNetworkServiceAssociations reticolo vpc: ListServiceNetworkVpcAssociations reticolo vpc: ListServiceNetworkVpcEndpointAssociations reticolo vpc: ListServiceNetworks reticolo vpc: ListServices reticolo vpc: ListTargetGroups reticolo vpc: ListTargets reticolo vpc: PutAuthPolicy reticolo vpc: PutResourcePolicy reticolo vpc: RegisterTargets reticolo vpc: StartDomainVerification reticolo vpc: UpdateAccessLogSubscription reticolo vpc: UpdateListener reticolo vpc: UpdateResourceConfiguration

Prefisso del servizio	Azioni
	reticolo vpc: UpdateResourceGateway reticolo vpc: UpdateRule reticolo vpc: UpdateService reticolo vpc: UpdateServiceNetwork reticolo vpc: UpdateServiceNetworkVpcAssociation reticolo vpc: UpdateTargetGroup

Prefisso del servizio	Azioni
wafv2	wafv2: ACL AssociateWeb
	wafv2: CheckCapacity
	wafv2:CreateAPIKey
	wafv2:CreateIPSet
	wafv2: CreateRegexPatternSet
	wafv2: CreateRuleGroup
	wafv2: ACL CreateWeb
	wafv2:DeleteAPIKey
	wafv2: DeleteFirewallManagerRuleGroups
	wafv2:DeleteIPSet
	wafv2: DeleteLoggingConfiguration
	wafv2: DeletePermissionPolicy
	wafv2: DeleteRegexPatternSet
	wafv2: DeleteRuleGroup
	wafv2: ACL DeleteWeb
	wafv2: DescribeAllManagedProducts
	wafv2: DescribeManagedProductsByVendor
	wafv2: DescribeManagedRuleGroup
	wafv2: ACL DisassociateWeb
	wafv2: GenerateMobileSdkReleaseUrl
	wafv2: GetDecryptedAPIKey

Prefisso del servizio	Azioni
	wafv2:GetIPSet
	wafv2: GetLoggingConfiguration
	wafv2: GetManagedRuleSet
	wafv2: GetMobileSdkRelease
	wafv2: GetRateBasedStatementManagedKeys
	wafv2: GetRegexPatternSet
	wafv2: GetRevenueStatistics
	wafv2: GetRevenueStatisticsSummary
	wafv2: GetRevenueStatisticsTimeSeries
	wafv2: GetRuleGroup
	wafv2: GetSampledRequests
	wafv2: GetWebACLForResource
	wafv2:ListAPIKeys
	wafv2: ListAvailableManagedRuleGroupVersions
	wafv2: ListAvailableManagedRuleGroups
	wafv2:ListIPSets
	wafv2: ListLoggingConfigurations
	wafv2: ListManagedRuleSets
	wafv2: ListMobileSdkReleases
	wafv2: ListRegexPatternSets
	wafv2: ListResourcesForWebACL

Prefisso del servizio	Azioni
	wafv2: ListRuleGroups wafv2: ListSettlementRecords wafv2: ListWebACLs wafv2: PutLoggingConfiguration wafv2: PutManagedRuleSetVersions wafv2:UpdateIPSet wafv2: UpdateManagedRuleSetVersionExpiryDate wafv2: UpdateRegexPatternSet wafv2: UpdateRuleGroup wafv2: ACL UpdateWeb

Prefisso del servizio	Azioni
wellarchitected	ben architettato: AssociateLenses ben architettato: AssociateProfiles ben architettato: CreateLensShare ben architettato: CreateLensVersion ben architettato: CreateMilestone ben architettato: CreateProfile ben architettato: CreateProfileShare ben architettato: CreateReviewTemplate ben architettato: CreateWorkload ben architettato: CreateWorkloadShare ben architettato: DeleteLens ben architettato: DeleteLensShare ben architettato: DeleteProfile ben architettato: DeleteProfileShare ben architettato: DeleteReviewTemplate ben architettato: DeleteTemplateShare ben architettato: DeleteWorkload ben architettato: DeleteWorkloadShare ben architettato: DisassociateLenses ben architettato: DisassociateProfiles ben architettato: ExportLens

Prefisso del servizio	Azioni
	ben architettato: GetAnswer ben architettato: GetConsolidatedReport ben architettato: GetGlobalSettings ben architettato: GetLens ben architettato: GetLensReview ben architettato: GetLensReviewReport ben architettato: GetLensVersionDifference ben architettato: GetMilestone ben architettato: GetProfile ben architettato: GetProfileTemplate ben architettato: GetReviewTemplate ben architettato: GetReviewTemplateAnswer ben architettato: GetReviewTemplateLensReview ben architettato: GetWorkload ben architettato: ImportLens ben architettato: ListAnswers ben architettato: ListCheckDetails ben architettato: ListCheckSummaries ben architettato: ListLensReviewImprovements ben architettato: ListLensReviews ben architettato: ListLensShares

Prefisso del servizio	Azioni
	ben architettato: ListLenses ben architettato: ListMilestones ben architettato: ListNotifications ben architettato: ListProfileNotifications ben architettato: ListProfileShares ben architettato: ListProfiles ben architettato: ListReviewTemplateAnswers ben architettato: ListReviewTemplates ben architettato: ListShareInvitations ben architettato: ListTemplateShares ben architettato: ListWorkloadShares ben architettato: ListWorkloads ben architettato: UpdateAnswer ben architettato: UpdateGlobalSettings ben architettato: UpdateIntegration ben architettato: UpdateLensReview ben architettato: UpdateProfile ben architettato: UpdateReviewTemplate ben architettato: UpdateReviewTemplateLensReview ben architettato: UpdateShareInvitation ben architettato: UpdateWorkload

Prefisso del servizio	Azioni
	ben architettato: UpdateWorkloadShare ben architettato: UpgradeLensReview ben architettato: UpgradeProfileVersion ben architettato: UpgradeReviewTemplateLensReview

Prefisso del servizio	Azioni
wisdom	saggezza: CreateAssistant
	saggezza: CreateAssistantAssociation
	saggezza: CreateContent
	saggezza: CreateKnowledgeBase
	saggezza: CreateQuickResponse
	saggezza: CreateSession
	saggezza: DeleteAssistant
	saggezza: DeleteAssistantAssociation
	saggezza: DeleteContent
	saggezza: DeleteImportJob
	saggezza: DeleteKnowledgeBase
	saggezza: DeleteQuickResponse
	saggezza: GetAssistant
	saggezza: GetAssistantAssociation
	saggezza: GetContent
	saggezza: GetContentAssociation
	saggezza: GetContentSummary
	saggezza: GetImportJob
	saggezza: GetKnowledgeBase
	saggezza: GetRecommendations
	saggezza: GetSession

Prefisso del servizio	Azioni
	saggezza: ListAssistantAssociations
	saggezza: ListAssistants
	saggezza: ListContentAssociations
	saggezza: ListContents
	saggezza: ListImportJobs
	saggezza: ListKnowledgeBases
	saggezza: ListQuickResponses
	saggezza: NotifyRecommendationsReceived
	saggezza: QueryAssistant
	saggezza: RemoveKnowledgeBaseTemplateUri
	saggezza: SearchContent
	saggezza: SearchQuickResponses
	saggezza: SearchSessions
	saggezza: StartContentUpload
	saggezza: StartImportJob
	saggezza: UpdateContent
	saggezza: UpdateKnowledgeBaseTemplateUri
	saggezza: UpdateQuickResponse
	saggezza: UpdateSession

Prefisso del servizio	Azioni
worklink	collegamento di lavoro: AssociateDomain collegamento di lavoro: AssociateWebsiteAuthorizationProvider collegamento di lavoro: AssociateWebsiteCertificateAuthority collegamento di lavoro: CreateFleet collegamento di lavoro: DeleteFleet collegamento di lavoro: DescribeAuditStreamConfiguration collegamento di lavoro: DescribeCompanyNetworkConfiguration collegamento di lavoro: DescribeDevice collegamento di lavoro: DescribeDevicePolicyConfiguration collegamento di lavoro: DescribeDomain collegamento di lavoro: DescribeFleetMetadata collegamento di lavoro: DescribeIdentityProviderConfiguration collegamento di lavoro: DescribeWebsiteCertificateAuthority collegamento di lavoro: DisassociateDomain collegamento di lavoro: DisassociateWebsiteAuthorizationProvider collegamento di lavoro: DisassociateWebsiteCertificateAuthority collegamento di lavoro: ListDevices collegamento di lavoro: ListDomains collegamento di lavoro: ListFleets collegamento di lavoro: ListWebsiteAuthorizationProviders collegamento di lavoro: ListWebsiteCertificateAuthorities

Prefisso del servizio	Azioni
	collegamento di lavoro: RestoreDomainAccess collegamento di lavoro: RevokeDomainAccess collegamento di lavoro: SignOutUser collegamento di lavoro: UpdateAuditStreamConfiguration collegamento di lavoro: UpdateCompanyNetworkConfiguration collegamento di lavoro: UpdateDevicePolicyConfiguration collegamento di lavoro: UpdateDomainMetadata collegamento di lavoro: UpdateFleetMetadata collegamento di lavoro: UpdateIdentityProviderConfiguration

Prefisso del servizio	Azioni
workspace	spazi di lavoro: AcceptAccountLinkInvitation
	spazi di lavoro: AssociateConnectionAlias
	spazi di lavoro: AssociateIpGroups
	spazi di lavoro: AssociateWorkspaceApplication
	spazi di lavoro: CopyWorkspaceImage
	spazi di lavoro: CreateAccountLinkInvitation
	spazi di lavoro: CreateConnectClientAddIn
	spazi di lavoro: CreateConnectionAlias
	spazi di lavoro: CreateIpGroup
	spazi di lavoro: CreateStandbyWorkspaces
	spazi di lavoro: CreateUpdatedWorkspaceImage
	spazi di lavoro: CreateWorkspaceBundle
	spazi di lavoro: CreateWorkspaceImage
	spazi di lavoro: CreateWorkspaces
	spazi di lavoro: CreateWorkspacesPool
	spazi di lavoro: DeleteAccountLinkInvitation
	spazi di lavoro: DeleteClientBranding
	spazi di lavoro: DeleteConnectClientAddIn
	spazi di lavoro: DeleteConnectionAlias
	spazi di lavoro: DeleteIpGroup
	spazi di lavoro: DeleteWorkspaceBundle

Prefisso del servizio	Azioni
	spazi di lavoro: DeleteWorkspaceImage
	spazi di lavoro: DeployWorkspaceApplications
	spazi di lavoro: DeregisterWorkspaceDirectory
	spazi di lavoro: DescribeAccount
	spazi di lavoro: DescribeAccountModifications
	spazi di lavoro: DescribeApplicationAssociations
	spazi di lavoro: DescribeApplications
	spazi di lavoro: DescribeBundleAssociations
	spazi di lavoro: DescribeClientBranding
	spazi di lavoro: DescribeClientProperties
	spazi di lavoro: DescribeConnectClientAddIns
	spazi di lavoro: DescribeConnectionAliasPermissions
	spazi di lavoro: DescribeConnectionAliases
	spazi di lavoro: DescribeCustomWorkspaceImageImport
	spazi di lavoro: DescribeImageAssociations
	spazi di lavoro: DescribeIpGroups
	spazi di lavoro: DescribeWorkspaceAssociations
	spazi di lavoro: DescribeWorkspaceBundles
	spazi di lavoro: DescribeWorkspaceDirectories
	spazi di lavoro: DescribeWorkspaceImagePermissions
	spazi di lavoro: DescribeWorkspaceSnapshots

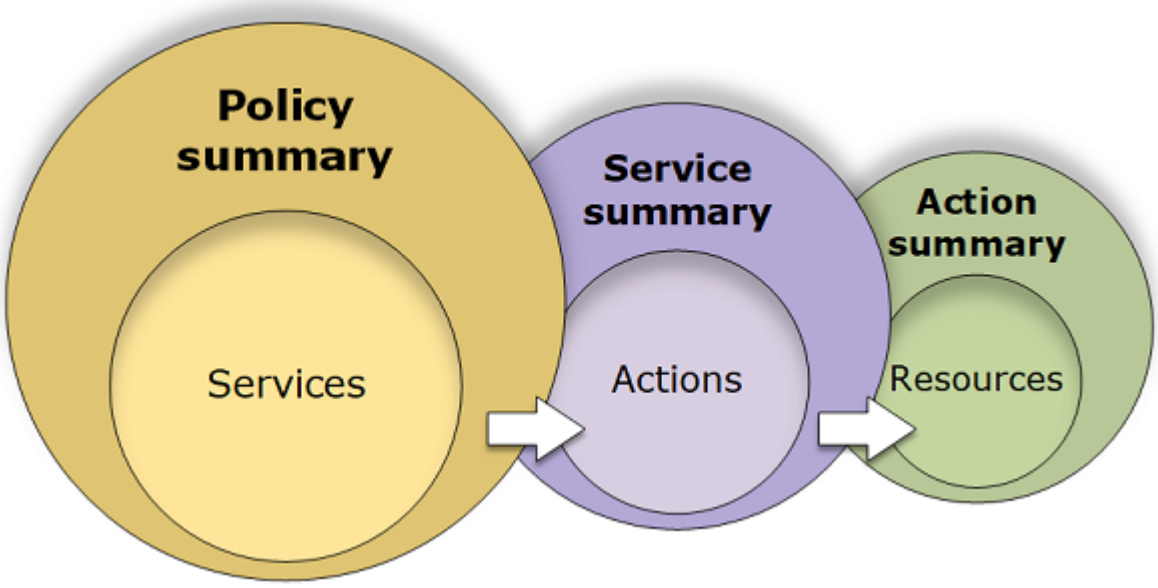
Prefisso del servizio	Azioni
	spazi di lavoro: DescribeWorkspaces
	spazi di lavoro: DescribeWorkspacesConnectionStatus
	spazi di lavoro: DescribeWorkspacesPoolSessions
	spazi di lavoro: DescribeWorkspacesPools
	spazi di lavoro: DisassociateConnectionAlias
	spazi di lavoro: DisassociateIpGroups
	spazi di lavoro: DisassociateWorkspaceApplication
	spazi di lavoro: GetAccountLink
	spazi di lavoro: ImportClientBranding
	spazi di lavoro: ImportWorkspaceImage
	spazi di lavoro: ListAccountLinks
	spazi di lavoro: ListAvailableManagementCidrRanges
	spazi di lavoro: MigrateWorkspace
	spazi di lavoro: ModifyAccount
	spazi di lavoro: ModifyCertificateBasedAuthProperties
	spazi di lavoro: ModifyClientProperties
	spazi di lavoro: ModifyEndpointEncryptionMode
	spazi di lavoro: ModifySamlProperties
	spazi di lavoro: ModifySelfservicePermissions
	spazi di lavoro: ModifyStreamingProperties
	spazi di lavoro: ModifyWorkspaceAccessProperties

Prefisso del servizio	Azioni
	spazi di lavoro: ModifyWorkspaceCreationProperties
	spazi di lavoro: ModifyWorkspaceProperties
	spazi di lavoro: ModifyWorkspaceState
	spazi di lavoro: RebootWorkspaces
	spazi di lavoro: RebuildWorkspaces
	spazi di lavoro: RegisterWorkspaceDirectory
	spazi di lavoro: RejectAccountLinkInvitation
	spazi di lavoro: RestoreWorkspace
	spazi di lavoro: StartWorkspaces
	spazi di lavoro: StartWorkspacesPool
	spazi di lavoro: StopWorkspaces
	spazi di lavoro: StopWorkspacesPool
	spazi di lavoro: TerminateWorkspaces
	spazi di lavoro: TerminateWorkspacesPool
	spazi di lavoro: TerminateWorkspacesPoolSession
	spazi di lavoro: UpdateConnectClientAddIn
	spazi di lavoro: UpdateConnectionAliasPermission
	spazi di lavoro: UpdateWorkspaceBundle
	spazi di lavoro: UpdateWorkspaceImagePermission
	spazi di lavoro: UpdateWorkspacesPool

Prefisso del servizio	Azioni
xray	radiografia: CreateGroup radiografia: CreateSamplingRule radiografia: DeleteGroup radiografia: DeleteResourcePolicy radiografia: DeleteSamplingRule radiografia: GetEncryptionConfig radiografia: GetGroup radiografia: GetGroups radiografia: GetInsight radiografia: GetInsightEvents radiografia: GetInsightImpactGraph radiografia: GetInsightSummaries radiografia: GetSamplingRules radiografia: ListResourcePolicies radiografia: PutEncryptionConfig radiografia: PutResourcePolicy radiografia: UpdateGroup radiografia: UpdateSamplingRule

Riepiloghi delle policy

La console IAM include tabelle di riepilogo di policy che descrivono il livello di accesso, le risorse e le condizioni concesse o rifiutate per ciascun servizio in una policy. Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione. La tabella di riepilogo della policy include un elenco di servizi. Scegli un servizio per visualizzare il riepilogo del servizio. Questa tabella include un elenco delle operazioni e le autorizzazioni associate per il servizio scelto. È possibile scegliere un'operazione dalla tabella per visualizzare il riepilogo dell'operazione. Questa tabella include un elenco di risorse e condizioni per l'operazione scelta.



Puoi visualizzare i riepiloghi delle policy nella pagina Users (Utenti) o Roles (Ruoli) per tutte le policy (gestite e inline) collegate a tale utente. e visualizzare i riepiloghi nella pagina Policies (Policy) per tutte le policy gestite. Le politiche gestite includono le politiche AWS gestite, le politiche relative alle funzioni lavorative AWS gestite e le politiche gestite dal cliente. Puoi visualizzare riepiloghi per queste policy nella pagina Policies (Policy), indipendentemente dal fatto che siano collegate a un utente o a un'altra identità IAM.

Puoi utilizzare le informazioni nei riepiloghi delle policy per comprendere le autorizzazioni che vengono concesse o negate dalla policy. I riepiloghi delle policy facilitano la risoluzione dei problemi e consentono di correggere policy che non forniscono le autorizzazioni previste.

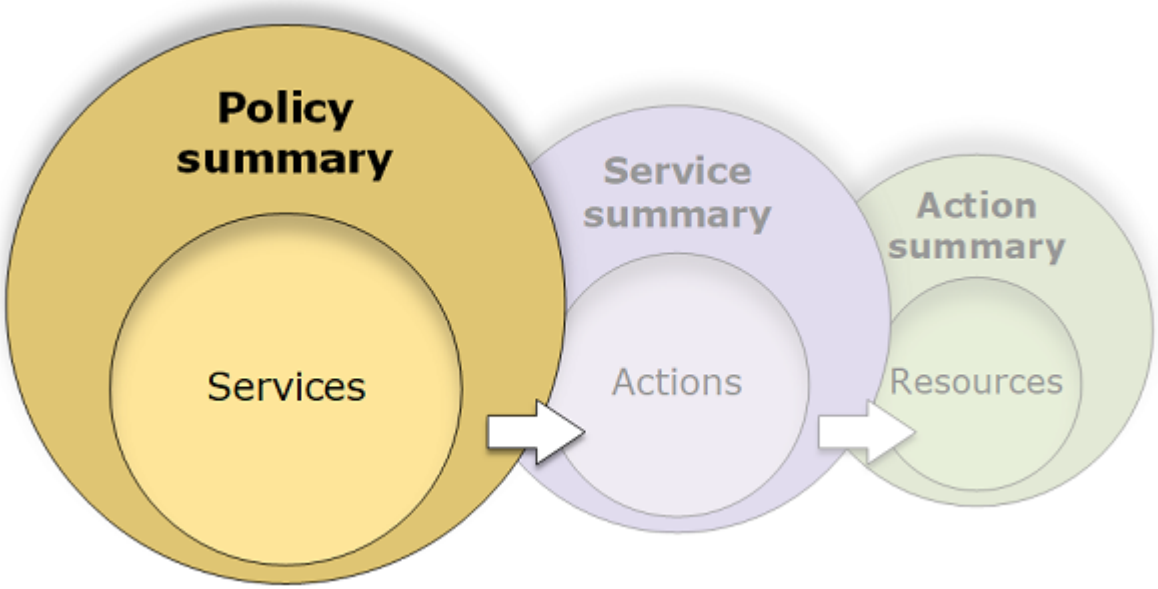
Per istruzioni dettagliate sulla visualizzazione dei riepiloghi, vedereVisualizzazione dei riepiloghi delle policy, e. Visualizzazione dei riepiloghi dei servizi Visualizzazione dei riepiloghi delle azioni

Argomenti

- Riepilogo della policy (elenco di servizi)
- Livelli di accesso nei riepiloghi delle policy
- Riepilogo del servizio (elenco di operazioni)
- Riepilogo delle operazioni (elenco di risorse)
- Esempi di riepiloghi di policy

Riepilogo della policy (elenco di servizi)

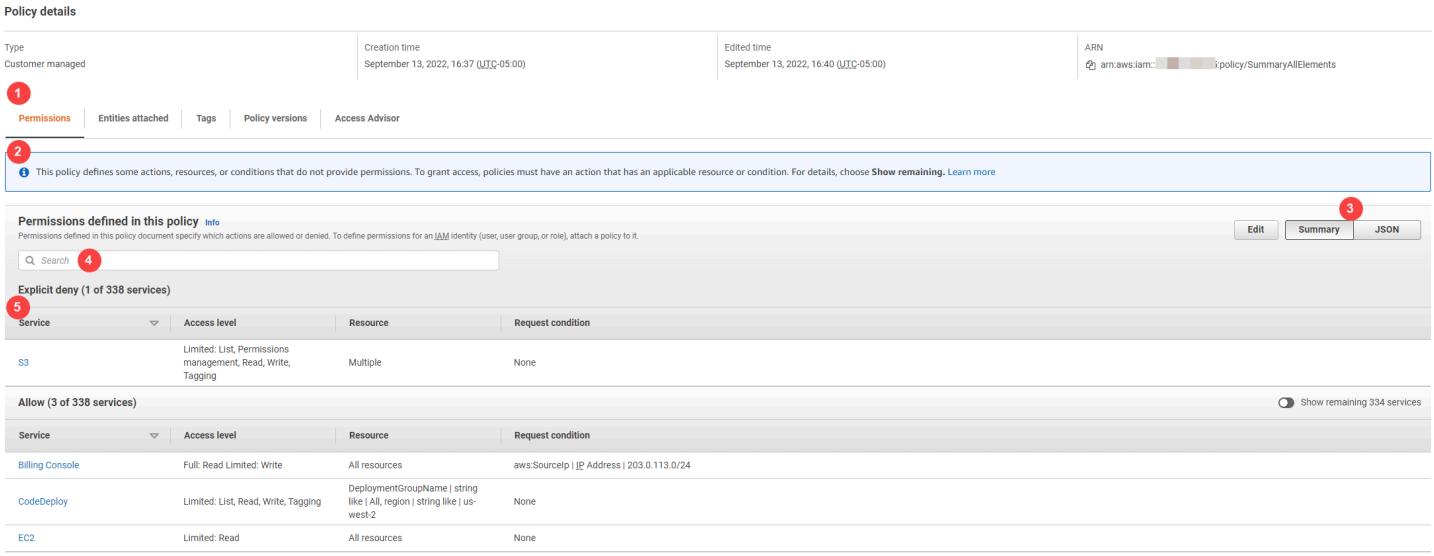
Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione. La tabella riepilogo della policy include un elenco di servizi e riepiloghi delle autorizzazioni definite dalla policy scelta.



La tabella di riepilogo della policy è raggruppata in una o più sezioni: Uncategorized services (Servizi non categorizzati), Explicit deny (Rifiuto esplicito) e Allow (Permetti). Se la policy include un servizio che IAM non riconosce, il servizio viene incluso nella sezione Servizi non categorizzati della tabella. Se IAM riconosce il servizio, viene incluso nelle sezioni Rifiuto esplicito o Permetti della tabella, a seconda dell'effetto della policy (`Deny` o `Allow`).

Comprendere gli elementi del riepilogo di una policy

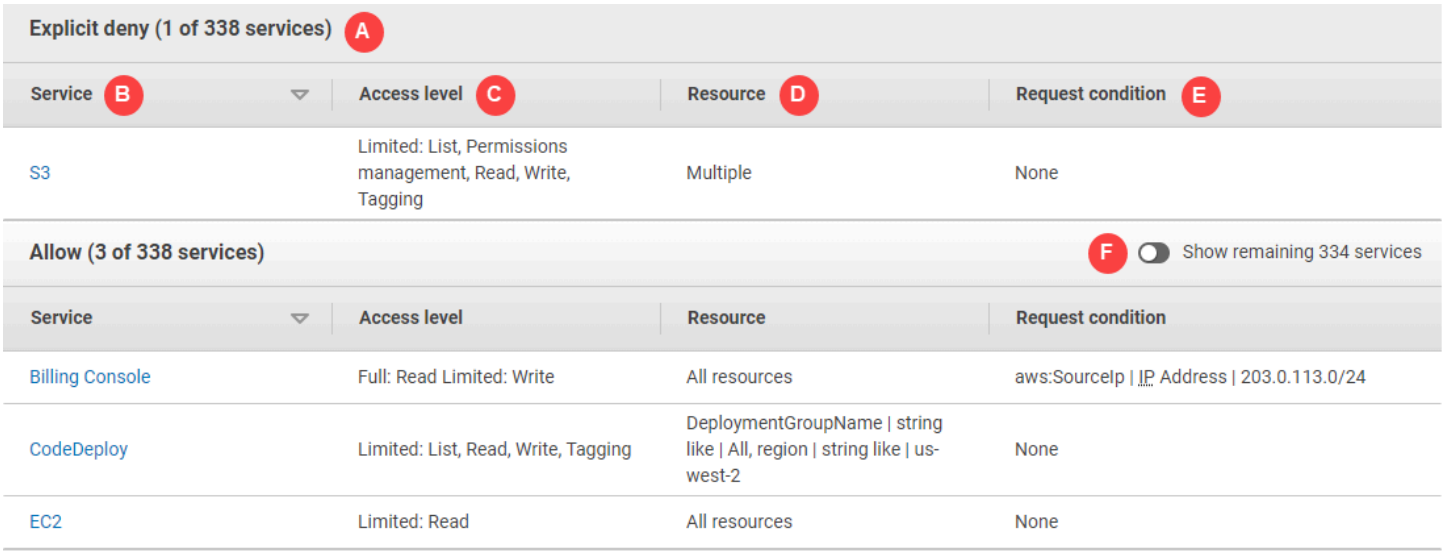
Nell'esempio seguente di pagina dei dettagli di una policy, la SummaryAllElementspolicy è una policy gestita (policy gestita dal cliente) allegata direttamente all'utente. Questa policy è espansa per visualizzare il riepilogo della policy.



Nell'immagine precedente, il riepilogo della policy è visibile all'interno della pagina Policy:

1. La scheda Autorizzazioni include le autorizzazioni definite nella policy.
2. Se la policy non concede le autorizzazioni a tutte le operazioni, risorse e condizioni definite per il servizio nella policy, viene visualizzato un banner di avviso o errore nella parte superiore della pagina. Il riepilogo della policy include i dettagli sul problema. Per ulteriori informazioni su come i riepiloghi della policy aiutano a capire e risolvere i problemi delle autorizzazioni che la policy concede, consultare the section called “La policy non concede le autorizzazioni previste”.
3. Utilizza i pulsanti Riepilogo e JSON per passare tra il riepilogo della policy e il documento della policy JSON.
4. Utilizza la casella Cerca per ridurre l'elenco dei servizi e trovare un servizio specifico.
5. La visualizzazione estesa mostra ulteriori dettagli della SummaryAllElementspolitica.

La seguente immagine della tabella di riepilogo della politica mostra la SummaryAllElementspolitica estesa nella pagina dei dettagli della politica.



Explicit deny (1 of 338 services) A

Service B	Access level C	Resource D	Request condition E
S3	Limited: List, Permissions management, Read, Write, Tagging	Multiple	None

Allow (3 of 338 services) F Show remaining 334 services

Service	Access level	Resource	Request condition
Billing Console	Full: Read Limited: Write	All resources	aws:SourceIp | IP Address | 203.0.113.0/24
CodeDeploy	Limited: List, Read, Write, Tagging	DeploymentGroupName | string like | All, region | string like | us-west-2	None
EC2	Limited: Read	All resources	None

Nell'immagine precedente, il riepilogo della policy è visibile all'interno della pagina Policy:

A. Per i servizi riconosciuti, IAM li organizza in base al fatto che la policy permetta o rifiuti esplicitamente l'utilizzo del servizio. In questo esempio, la policy include una `Deny` dichiarazione per il servizio Amazon S3 e `Allow` dichiarazioni per i servizi di fatturazione e Amazon EC2. CodeDeploy

B. Servizio: questa colonna riporta i servizi definiti all'interno della policy e fornisce i dettagli per ciascun servizio. Ogni nome di servizio nella tabella di riepilogo della policy è un collegamento alla tabella di riepilogo del servizio illustrata in Riepilogo del servizio (elenco di operazioni). In questo esempio, vengono definite le autorizzazioni per i servizi Amazon S3, Billing CodeDeploy e Amazon EC2.

C. Livello di accesso: questa colonna indica se le operazioni di ciascun livello di accesso (`List`, `Read`, `Write`, `Permission Management` e `Tagging`) dispongono dell'autorizzazione `Full` o `Limited` definita nella policy. Per ulteriori informazioni ed esempi del riepilogo del livello di accesso, consultare Livelli di accesso nei riepiloghi delle policy.

- Accesso completo: questa voce indica che il servizio ha accesso a tutte le operazioni entro tutti e quattro i livelli di accesso disponibili per il servizio.
- Se la voce non include Full access (Accesso completo), il servizio ha accesso ad alcune ma non tutte le operazioni per il servizio. L'accesso viene quindi definito dalle seguenti descrizioni per ciascuna delle classificazioni a livello di accesso (`List`, `Read`, `Write`, `Permission Management` e `Tagging`):

Full (Completo): la policy consente l'accesso a tutte le operazioni all'interno di ciascuna classificazione del livello di accesso elencata. In questo esempio, la policy consente l'accesso a tutte le operazioni `Read` di fatturazione.

Limited (Limitato): la policy consente l'accesso a una o più operazioni all'interno di ciascuna classificazione del livello di accesso elencata, ma non a tutte. In questo esempio, la policy consente l'accesso ad alcune operazioni `Write` di fatturazione.

D. Risorsa: questa colonna mostra le risorse che la policy specifica per ogni servizio.

- Multiple: la policy include più di una ma non tutte le risorse all'interno del servizio. In questo esempio, l'accesso viene rifiutato esplicitamente a più di una risorsa Amazon S3.
- Tutte le risorse: la policy è definita per tutte le risorse all'interno del servizio. In questo esempio, la policy permette di eseguire le operazioni elencate per tutte le risorse di fatturazione.
- Testo della risorsa: la policy include una risorsa all'interno del servizio. In questo esempio, le azioni elencate sono consentite solo sulla risorsa. `DeploymentGroupName` CodeDeploy A seconda delle informazioni che il servizio fornisce a IAM, è possibile che venga visualizzato un ARN o il tipo di risorsa definita.

> **Note**
>
> Questa colonna può includere una risorsa da un altro servizio. Se l'istruzione di policy che include la risorsa non include entrambe le operazioni e risorse dallo stesso servizio, la policy include le risorse non corrispondenti. IAM non visualizza avvisi per le risorse non corrispondenti al momento della creazione di una policy o della visualizzazione di una policy nel riepilogo della policy. Se questa colonna include una risorsa non corrispondente, è necessario verificare se ci sono errori nella policy. Per verificare meglio le policy, eseguire sempre un test tramite il simulatore di policy.

E. Condizioni richiesta: questa colonna indica se i servizi o le operazioni associati alla risorsa sono soggetti a condizioni.

- Nessuna: la policy non include condizioni per il servizio. In questo esempio non vengono applicate condizioni alle operazioni rifiutate nel servizio Amazon S3.
- Testo della condizione: la policy include una condizione per il servizio. In questo esempio, le operazioni di Fatturazione elencate sono consentite solo se l'indirizzo IP di origine corrisponde a `203.0.113.0/24`.

- Multiple: la policy include più di una condizione per il servizio. Per visualizzare tutte le condizioni multiple per la policy, seleziona JSON per visualizzare il documento della policy.

F. Mostra servizi rimanenti: attiva/disattiva questo pulsante per espandere la tabella e includere i servizi che non sono definiti dalla policy. Questi servizi vengono rifiutati implicitamente (o rifiutati per impostazione predefinita) all'interno della policy. Tuttavia, un'istruzione in un'altra policy potrebbe permettere o rifiutare esplicitamente l'utilizzo del servizio. Il riepilogo della policy fornisce un elenco delle autorizzazioni di una singola policy. Per ulteriori informazioni su come il AWS servizio decide se consentire o rifiutare una determinata richiesta, consultaLogica di valutazione delle policy.

Quando una policy o un elemento all'interno della policy non concede autorizzazioni, IAM vengono forniti ulteriori avvisi e informazioni nel riepilogo della policy. La seguente tabella di riepilogo delle politiche mostra la versione estesa dei servizi Mostra i servizi rimanenti nella pagina dei dettagli della SummaryAllElementspolitica con i possibili avvisi.

Explicit deny (1 of 338 services)

Service	Access level	Resource a	Request condition b
S3	Limited: List, Permissions management, Read, Write, Tagging	c Multiple| ⚠ One or more actions do not have an applicable resource.	None

Allow (3 of 338 services) Show remaining 334 services

Service	Access level	Resource	Request condition
Billing Console	Full: Read Limited: Write	All resources	aws:SourceIp | IP Address | 203.0.113.0/24
CodeCommit	None	d ⚠ No resources are defined.	None
CodeDeploy	Limited: List, Read, Write, Tagging	e DeploymentGroupName | string like | All, region | string like | us-west-2| ⚠ One or more actions do not have an applicable resource.	None
EC2	Limited: Read	All resources	None
S3	None	None| ⚠ One or more actions do not have an applicable resource.	f None | ⚠ One or more conditions do not have an applicable action.

Nell'immagine precedente, è possibile visualizzare tutti i servizi che includono operazioni, risorse o condizioni definite senza autorizzazioni.

a. Avvisi risorse: per i servizi che non forniscono autorizzazioni per tutte le operazioni o risorse incluse, viene visualizzato uno dei seguenti avvisi nella colonna Risorsa della tabella:

- 

  No resources are defined (Nessuna risorsa definita) : indica che il servizio ha definito le operazioni, ma nessuna risorsa supportata è inclusa nella policy.

- 

  One or more actions do not have an applicable resource (Una o più operazioni non hanno una risorsa applicabile) : indica che il servizio ha definito le operazioni, ma che alcune di esse non hanno una risorsa supportata.

- 

  One or more resources do not have an applicable action (Una o più risorse non hanno un'operazione applicabile) : indica che il servizio ha definito le risorse, ma che alcune di esse non hanno un'operazione di supporto.

  Se un servizio include sia operazioni senza una risorsa applicabile sia risorse con una risorsa applicabile, viene visualizzato solo l'avviso Una o più risorse non hanno un'operazione applicabile. Questo perché quando si visualizza il riepilogo del servizio per il servizio, le risorse che non si applicano a nessuna operazione non vengono visualizzate. Per l'operazione `ListAllMyBuckets`, questa policy include l'ultimo avvertimento perché l'operazione non supporta le autorizzazioni a livello di risorsa e non supporta la chiave di condizione `s3:x-amz-acl`. Se si risolve il problema della risorsa o della condizione, il problema rimanente appare in un avviso dettagliato.

b. Avvisi di condizione richiesta: per i servizi che non forniscono autorizzazioni per tutte le condizioni incluse, viene visualizzato uno dei seguenti avvisi nella colonna Condizione richiesta della tabella:

   - 

     One or more actions do not have an applicable condition (Una o più operazioni non hanno una condizione applicabile) : indica che il servizio ha definito le operazioni, ma che alcune di esse non hanno una condizione supportata.

   - 

     One or more conditions do not have an applicable action (Una o più condizioni non hanno un'operazione applicabile) : indica che il servizio ha definito le condizioni, ma che alcune di esse non hanno un'operazione di supporto.

c. Multiple |

   One or more actions do not have an applicable resource (Multiple | Una o più operazioni non hanno una risorsa applicabile). : l'istruzione `Deny` per Amazon S3 include più di una risorsa.

Include anche più di un'operazione e alcune operazioni supportano le risorse, altre no. Per visualizzare questa policy, consulta the section called "SummaryAllElements Documento sulla politica JSON". In questo caso, la policy include tutte le operazioni Amazon S3 e vengono rifiutate solo le operazioni che possono essere eseguite per un oggetto bucket o un bucket.

d. ⚠

Nessuna risorsa definita: il servizio ha definito le operazioni, ma nella policy non sono incluse risorse supportate e pertanto il servizio non fornisce autorizzazioni. In questo caso, la politica include CodeCommit azioni ma non CodeCommit risorse.

e. DeploymentGroupName | string like | All, region | string like | us-west-2

⚠

| Una o più azioni non hanno una risorsa applicabile. : il servizio dispone di una operazione definita e di almeno un'altra operazione senza una risorsa di supporto.

f. Nessuna |

⚠

Una o più condizioni non hanno un'operazione applicabile. : il servizio dispone almeno di una chiave di condizione che non ha un'operazione di supporto.

SummaryAllElements Documento sulla politica JSON

La SummaryAllElementspolicy non è pensata per essere utilizzata per definire le autorizzazioni nel tuo account. Al contrario, è inclusa per dimostrare gli errori e gli avvisi che possono verificarsi durante la visualizzazione di una policy di riepilogo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "billing:Get*",
                "payments:List*",
                "payments:Update*",
                "account:Get*",
                "account:List*",
```

```
                "cur:GetUsage*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "203.0.113.0/24"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::customer",
                "arn:aws:s3:::customer/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetConsoleScreenshots"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "codedploy:*",
                "codecommit:*"
            ],
            "Resource": [
                "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*",
                "arn:aws:codebuild:us-east-1:123456789012:project/my-demo-
project"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetObject",
                "s3:DeletObject",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*",
                "arn:aws:autoscling:us-east-2:123456789012:autoscalgrp"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": [
                        "public-read"
                    ],
                    "s3:prefix": [
                        "custom",
                        "other"
                    ]
                }
            }
        }
    ]
}
```

Visualizzazione dei riepiloghi delle policy

È possibile visualizzare il riepilogo della policy per qualsiasi policy collegata a un utente o a un ruolo IAM. Per le policy gestite, è possibile visualizzare i riepiloghi della policy nella pagina Policy. Se la policy non include un riepilogo, consultare Riepilogo della policy mancante per scoprire perché.

Visualizzazione dei riepiloghi delle policy dalla pagina Policy

È possibile visualizzare il riepilogo della policy per le policy gestite nella pagina Policies (Policy).

Per visualizzare il riepilogo della policy dalla pagina Policies (Policy)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.

3. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
4. Nella pagina Dettagli della policy per la policy, visualizza la scheda Autorizzazioni per consultare il riepilogo della policy.

Visualizzazione di un riepilogo per una policy collegata a un utente

È possibile visualizzare il riepilogo della policy per qualsiasi policy collegata a un utente IAM.

Per visualizzare il riepilogo per una policy collegata a un utente

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Users (Utenti) dal riquadro di navigazione.
3. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per l'utente, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate all'utente direttamente o da un gruppo.
5. Nella tabella di policy per l'utente, espandere la riga della policy che si desidera visualizzare.

Visualizzazione di un riepilogo per una policy collegata a un ruolo

È possibile visualizzare il riepilogo della policy per qualsiasi policy collegata a un ruolo.

Per visualizzare il riepilogo per una policy collegata a un ruolo

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Ruoli.
3. Nell'elenco di ruoli, selezionare il nome del ruolo la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per il ruolo, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate al ruolo.
5. Nella tabella di policy per il ruolo, espandere la riga della policy che si desidera visualizzare.

Modifica delle policy per correggere gli avvisi

Durante la visualizzazione di un riepilogo della policy, è possibile che venga rilevato un errore o che la policy non fornisca le autorizzazioni previste. Non è possibile modificare direttamente un riepilogo della policy. Tuttavia, è possibile modificare una policy gestita dal cliente utilizzando l'editor visivo della policy, che rileva molti degli stessi errori e avvisi segnalati dal riepilogo della policy. È quindi possibile visualizzare le modifiche nel riepilogo della policy per verificare se sono stati risolti tutti i problemi. Per ulteriori informazioni su come modificare una policy inline, consultare the section called "Modificare le policy IAM". Non è possibile modificare le politiche AWS gestite.

È possibile modificare una policy per il riepilogo della policy con l'opzione Visivo

Per modificare una policy per il riepilogo della policy tramite l'opzione Visivo

1. Aprire il riepilogo della policy come spiegato nelle procedure precedenti.
2. Scegli Modifica.

   Se nella pagina Users (Utenti) si sceglie di modificare una policy gestita dal cliente collegata a tale utente, si viene reindirizzati alla pagina Policies (Policy). È possibile modificare le policy gestite dai clienti solo nella pagina Policies (Policy).
3. Seleziona l'opzione Visivo per visualizzare la rappresentazione visiva modificabile della policy. IAM potrebbe modificare la struttura della policy per ottimizzarla per l'editor visivo e facilitare così la ricerca e la risoluzione di eventuali problemi. Gli avvisi e i messaggi di errore nella pagina possono agevolare la risoluzione di eventuali problemi della policy. Per ulteriori informazioni sul modo in cui IAM modifica la struttura delle policy, consulta Modifica della struttura delle policy.
4. Modifica la policy e seleziona Successivo per visualizzare le modifiche riflesse nel riepilogo della policy. Se sono presenti ancora problemi, selezionare Previous (Precedente) per tornare alla schermata di modifica.
5. Per salvare le modifiche, scegliere Salva modifiche.

È possibile modificare una policy per il riepilogo della policy con l'opzione JSON

Per modificare una policy per il riepilogo della policy tramite l'opzione JSON

1. Aprire il riepilogo della policy come spiegato nelle procedure precedenti.
2. Utilizza i pulsanti Riepilogo e JSON per confrontare il riepilogo della policy e il documento della policy JSON. È possibile utilizzare queste informazioni per determinare quali righe del documento di policy modificare.

3. Scegli Modifica e quindi seleziona l'opzione JSON per modificare il documento della policy JSON.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'opzione dell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

Se nella pagina Users (Utenti) si sceglie di modificare una policy gestita dal cliente collegata a tale utente, si viene reindirizzati alla pagina Policies (Policy). È possibile modificare le policy gestite dai clienti solo nella pagina Policies (Policy).

4. Modifica la policy. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo). Se sono presenti ancora problemi, selezionare Previous (Precedente) per tornare alla schermata di modifica.
5. Per salvare le modifiche, scegliere Salva modifiche.

Livelli di accesso nei riepiloghi delle policy

AWS riepilogo del livello di accesso

I riepiloghi delle policy includono un riepilogo del livello di accesso che descrive le autorizzazioni operative definite per ciascun servizio menzionato nella policy. Per informazioni sui riepiloghi delle policy, consultare Riepiloghi delle policy. I riepiloghi dei livelli di accesso indicano se per le operazioni di ciascun livello di accesso (`List`, `Read`, `Tagging`, `Write`, and `Permissions management`) sono definite autorizzazioni `Full` o `Limited` nella policy. Per visualizzare la classificazione del livello di accesso assegnata a ciascuna azione in un servizio, vedere Azioni, risorse e chiavi di condizione per AWS i servizi.

L'esempio seguente descrive l'accesso fornito da una policy per i servizi in questione. Per esempi di documenti completi della policy JSON e i relativi riepiloghi, consultare Esempi di riepiloghi di policy.

Servizio	Livello di accesso	Questa policy fornisce quanto segue
IAM	Accesso completo ad	Accedi a tutte le operazioni all'interno del servizio IAM.
CloudWatch	Completo: elenco	Accesso a tutte CloudWatch le azioni nel livello di `List` accesso, ma nessun accesso alle azioni con classificazione a livello di `Permissions management` accesso `ReadWrite`, o.
Data Pipeline	Limitato: elenco, lettura	Accesso ad almeno una ma non a tutte AWS Data Pipeline le azioni nel livello di `Read` accesso `List` e, ma non alle `Permissions management` azioni `Write` o.
EC2	Completo: elenco, lettura Limitato: scrittura	Accesso a tutte le operazioni `List` e `Read` di Amazon EC2 e accesso ad almeno una, ma non a tutte le operazioni `Write` di Amazon EC2, ma nessun accesso alle operazioni con la classificazione a livello di accesso `Permissions management` .
S3	Limitato: lettura, scrittura, gestione autorizzazioni	Accesso ad almeno una, ma non a tutte le operazioni Amazon S3, `Read`, `Write` e `Permissions management` .
codedploy	(vuoto)	Accesso sconosciuto, perché IAM non riconosce questo servizio.
Gateway API	Nessuno	Nessun accesso è definito nella policy.
CodeBuild	 Non viene definita nessuna operazione.	Nessun accesso, perché non sono definite operazioni per il servizio. Per ulteriori informazioni su questo problema e sulla sua risoluzione, consultare the section called "La policy non concede le autorizzazioni previste".

Come spiegato in precedenza, Accesso completo indica che la policy concede l'accesso a tutte le operazioni all'interno del servizio. Le policy che forniscono accesso ad alcune ma non a tutte le operazioni all'interno di un servizio sono ulteriormente raggruppate in base alla classificazione del livello di accesso. Tale differenza viene indicata da uno dei seguenti raggruppamenti a livello di accesso:

- Full (Completo): la policy consente l'accesso a tutte le operazioni all'interno della classificazione specificata per il livello di accesso.
- Limited (Limitato): la policy consente l'accesso a una o più operazioni all'interno della classificazione specificata per il livello di accesso, ma non a tutte.
- None (Nessuno): la policy non fornisce alcun accesso.
- (vuoto): IAM non riconosce questo servizio. Se il nome del servizio include un errore ortografico, la policy non concederà l'accesso al servizio. Se il nome è corretto, il servizio potrebbe non supportare i riepiloghi della policy o potrebbe essere in anteprima. In questo caso, la policy potrebbe fornire l'accesso, ma questo non può essere visualizzato nel riepilogo della policy. Per richiedere il riepilogo della policy per un servizio disponibile a livello generale, consultare Il servizio non supporta i riepiloghi delle policy IAM.

I riepiloghi dei livelli di accesso che includono un accesso limitato (parziale) alle azioni sono raggruppati utilizzando le classificazioni dei livelli di AWS accesso`List`, `ReadTagging`, `Write` o. `Permissions management`

AWS livelli di accesso

AWS definisce le seguenti classificazioni dei livelli di accesso per le azioni in un servizio:

- List (Elenco): autorizzazione a elencare le risorse all'interno del servizio per determinare l'esistenza di un oggetto. Le operazioni con questo livello di accesso possono elencare gli oggetti, ma consentono di visualizzare i contenuti di una risorsa. Ad esempio, l'operazione Amazon S3 `ListBucket` ha un livello di accesso di tipo Elenco.
- Read (Lettura): autorizzazione a leggere ma non a modificare i contenuti e gli attributi delle risorse del servizio. Ad esempio, le operazioni di Amazon S3 `GetObject` e `GetBucketLocation` hanno un livello di accesso Lettura.
- Tagging: autorizzazione per eseguire operazioni che modificano solo lo stato di tag delle risorse. Ad esempio, le operazioni IAM `TagRole` e `UntagRole` dispongono del livello di accesso Tagging, in quanto consentono solo l'aggiunta e la rimozione di tag da un ruolo. Tuttavia, l'operazione

`CreateRole` consente il tagging di una risorsa del ruolo al momento della creazione di tale ruolo. Poiché l'operazione non aggiunge solo un tag, dispone del livello di accesso `Write`.

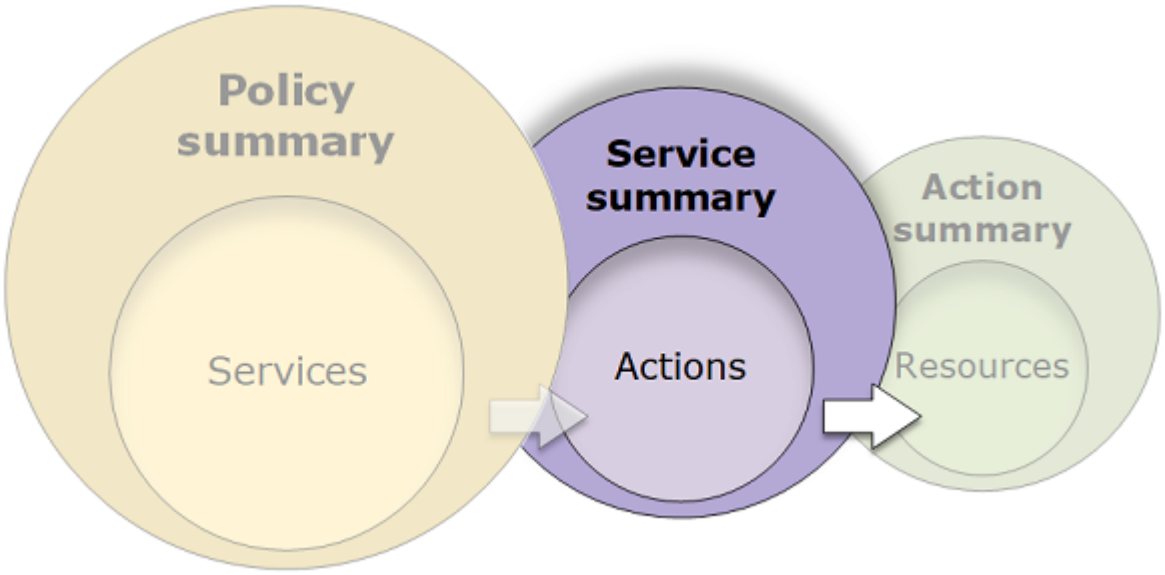
- Write (Scrittura): autorizzazione a creare, eliminare o modificare le risorse del servizio. Ad esempio, le operazioni Amazon S3 `CreateBucket`, `DeleteBucket` e `PutObject` hanno il livello di accesso Scrittura. Le operazioni `Write` potrebbero anche consentire la modifica di un tag della risorsa. Tuttavia, un'operazione che consente solo modifiche ai tag dispone del livello di accesso `Tagging`.
- Gestione delle autorizzazioni: la gestione delle autorizzazioni si riferisce alle azioni che controllano l'accesso all'interno Servizi AWS, incluse le autorizzazioni di identità IAM e non IAM, ma esclude i controlli di accesso a livello di rete come i gruppi di sicurezza. Ad esempio, la maggior parte degli IAM e delle azioni, così come AWS Organizations le azioni di Amazon S3, hanno il livello di accesso per la gestione delle autorizzazioni. `PutBucketPolicy` `DeleteBucketPolicy`

> **Suggerimento**
>
> Per migliorare la sicurezza delle tue Account AWS, limita o monitora regolarmente le policy che includono la classificazione dei livelli di accesso per la gestione delle autorizzazioni.

Per visualizzare la classificazione dei livelli di accesso per tutte le azioni di un servizio, vedi Azioni, risorse e chiavi di condizione per AWS i servizi.

Riepilogo del servizio (elenco di operazioni)

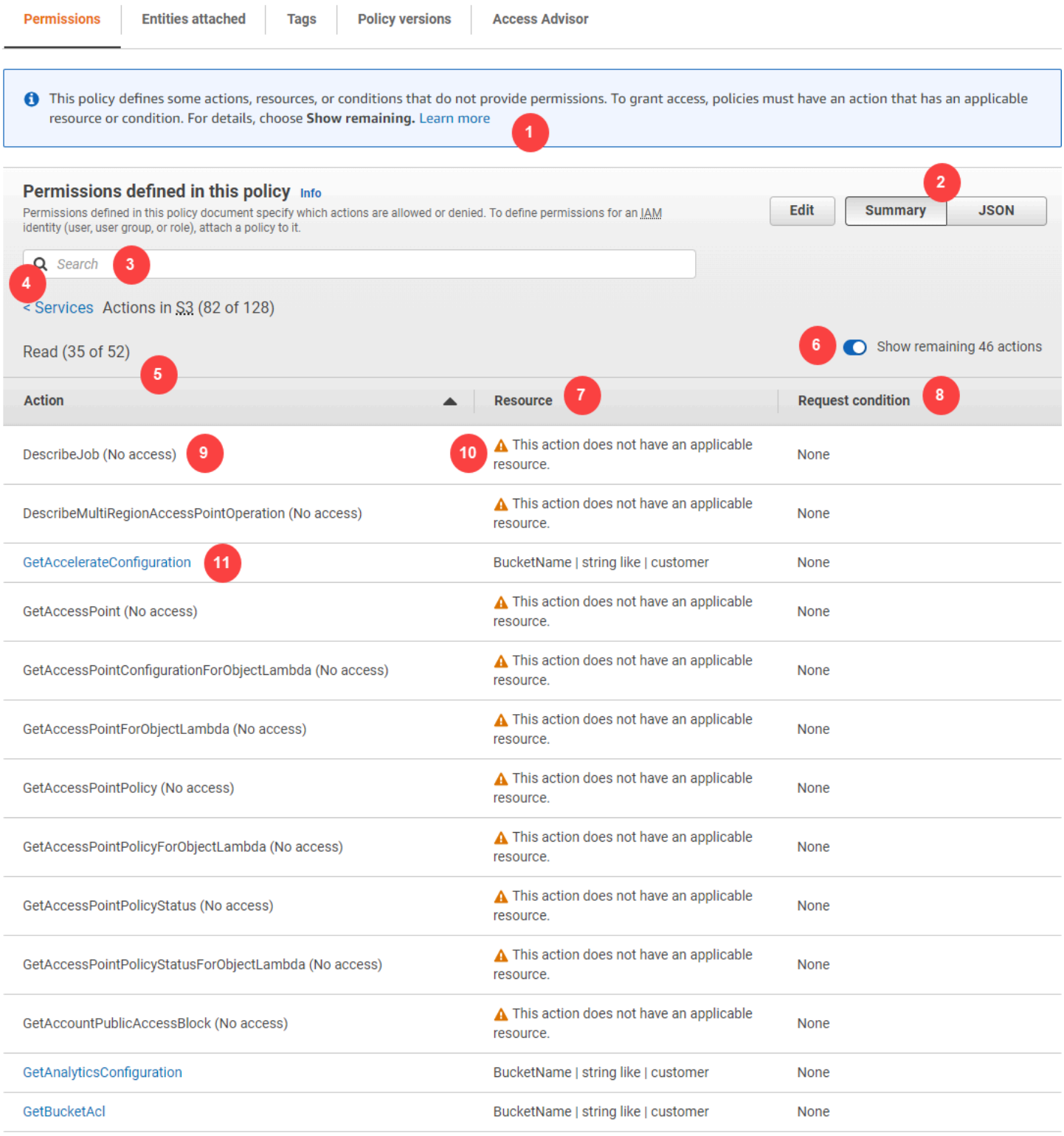
Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione. La tabella riepilogo del servizio include un elenco di operazioni e riepiloghi delle autorizzazioni definite dalla policy per il servizio selezionato.



È possibile visualizzare un riepilogo di servizio per ogni servizio elencato nel riepilogo della policy che concede autorizzazioni. La tabella è raggruppata in Uncategorized actions (Operazioni non categorizzate), Uncategorized resource types (Tipi di risorse non categorizzate) e sezioni di livello di accesso. Se la policy include un'operazione che IAM non riconosce, l'operazione sarà inclusa nella sezione Operazioni non categorizzate della tabella. Se IAM riconosce l'operazione, essa è inclusa in una delle sezioni della tabella dei livelli di accesso (Elenco, Lettura, Scrittura e Gestione autorizzazioni). Per visualizzare la classificazione del livello di accesso assegnata a ciascuna azione in un servizio, vedi Azioni, risorse e chiavi di condizione per AWS i servizi.

Informazioni sugli elementi del riepilogo di un servizio

L'esempio seguente è il riepilogo del servizio per le operazioni Amazon S3 consentite dal riepilogo della policy. Le operazioni per questo servizio sono raggruppate per livello di accesso. Ad esempio, sono definite 35 operazioni di lettura rispetto alle 52 operazioni di lettura totali disponibili per il servizio.



La pagina del riepilogo del servizio per una policy gestita include le seguenti informazioni:

1. Se la policy non concede le autorizzazioni a tutte le operazioni, risorse e condizioni definite per il servizio nella policy, quindi un banner di avviso viene visualizzato nella parte superiore della

pagina. Il riepilogo del servizio include i dettagli sul problema. Per ulteriori informazioni su come i riepiloghi della policy aiutano a capire e risolvere i problemi delle autorizzazioni che la policy concede, consultare the section called "La policy non concede le autorizzazioni previste".

2. Seleziona JSON per visualizzare ulteriori dettagli sulla policy. È possibile eseguire questa operazione per visualizzare tutte le condizioni applicate alle operazioni. (Se si sta visualizzando il riepilogo del servizio per una policy inline collegata direttamente a un utente, è necessario chiudere la finestra del dialogo del riepilogo del servizio e tornare al riepilogo della policy per accedere al documento della policy JSON.)
3. Per visualizzare il riepilogo per una risorsa operazione, digita le parole chiave nella casella Cerca per ridurre l'elenco delle operazioni disponibili.
4. Accanto alla freccia Servizi viene visualizzato il nome del servizio (in questo caso S3). Il riepilogo del servizio per questo servizio include l'elenco delle operazioni consentite o negate definite nella policy. Se il servizio viene visualizzato in (Negazione esplicita) nella scheda Autorizzazioni, le operazioni elencate nella tabella di riepilogo del servizio vengono negate esplicitamente. Se il servizio viene visualizzato in Consenti nella scheda Autorizzazioni, le operazioni elencate nella tabella di riepilogo del servizio vengono consentite.
5. Operazione: questa colonna elenca le operazioni definite nella policy e fornisce le risorse e le condizioni per ciascuna operazione. Se la policy concede o nega le autorizzazioni all'operazione, il nome dell'operazione si collega alla tabella riepilogo dell'operazione. La tabella raggruppa queste operazioni in almeno una o fino a cinque sezioni, a seconda del livello di accesso che la policy consente o nega. Le sezioni sono Elenco, Lettura, Scrittura, Gestione autorizzazionie Tagging. Il conteggio indica il numero di operazioni riconosciute che forniscono le autorizzazioni per ogni livello di accesso. Il totale è il numero di operazioni note per il servizio. In questo esempio, 35 operazioni forniscono le autorizzazioni su un totale di 52 operazioni di lettura Amazon S3 note. Per visualizzare la classificazione del livello di accesso assegnata a ciascuna azione in un servizio, vedere Azioni, risorse e chiavi di condizione per AWS i servizi.
6. Mostra operazioni rimanenti: attiva/disattiva questo pulsante per espandere o nascondere la tabella e includere le operazioni che sono note ma non forniscono le autorizzazioni per questo servizio. L'attivazione o la disattivazione del pulsante visualizza inoltre avvisi per gli elementi che non forniscono le autorizzazioni.
7. Risorsa: questa colonna mostra le risorse che la policy definisce per il servizio. IAM non verifica se la risorsa si applica a ciascuna operazione. In questo esempio, le operazioni nel servizio Amazon S3 sono consentite solo nella risorsa del bucket Amazon S3 `developer_bucket`. A seconda delle informazioni che il servizio fornisce a IAM, è possibile che venga visualizzato un ARN come

`arn:aws:s3:::developer_bucket/*`, oppure il tipo di risorsa definita, come `BucketName = developer_bucket`.

> **Note**
>
> Questa colonna può includere una risorsa da un altro servizio. Se l'istruzione di policy che include la risorsa non include entrambe le operazioni e risorse dallo stesso servizio, la policy include le risorse non corrispondenti. IAM non avvisa riguardo alle risorse non corrispondenti al momento della creazione di una policy oppure quando si visualizza una policy nel riepilogo del servizio. IAM inoltre non indica se l'operazione è valida per le risorse, solo se il servizio corrisponde. Se questa colonna include una risorsa non corrispondente, è necessario verificare se ci sono errori nella policy. Per verificare meglio le policy, eseguire sempre un test tramite il simulatore di policy.

8. Condizioni richiesta: questa colonna mostra se le operazioni che sono associate alla risorsa sono soggette a condizioni. Per ulteriori informazioni su queste condizioni, seleziona JSON per visualizzare il documento della policy JSON.
9. Nessun accesso: questa policy include un'operazione che non fornisce autorizzazioni.
10. Avviso risorsa: per operazioni con risorse che non forniscono autorizzazioni complete, viene visualizzato uno dei seguenti avvisi:
    - Questa operazione non supporta le autorizzazioni a livello di risorsa. Richiede un carattere jolly (*) per la risorsa. : indica che la policy include le autorizzazioni a livello di risorsa, ma deve includere `"Resource": ["*"]` per fornire le autorizzazioni per questa operazione.
    - This action does not have an applicable resource (Questa operazione non ha una risorsa applicabile) : indica che l'operazione è inclusa nella policy senza una risorsa supportata.
    - This action does not have an applicable resource and condition (Questa operazione non ha una risorsa e una condizione applicabili) : indica che l'operazione è inclusa nella policy senza una risorsa supportata e senza una condizione supportata. In questo caso, sussiste anche una condizione inclusa nella policy per questo servizio, ma non ci sono condizioni che si applicano a questa operazione.
11. Le operazioni che forniscono le autorizzazioni includono un collegamento al riepilogo dell'operazione.

Visualizzazione dei riepiloghi dei servizi

È possibile visualizzare un riepilogo di servizio per ogni servizio elencato nel riepilogo della policy che concede autorizzazioni.

Visualizzazione dei riepiloghi dei servizi dalla pagina Policy

È possibile visualizzare il riepilogo dei servizi per le policy gestite nella pagina Policy.

Per visualizzare il riepilogo del servizio per una policy gestita

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
4. Nella pagina Dettagli della policy per la policy, visualizza la scheda Autorizzazioni per consultare il riepilogo della policy.
5. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.

Visualizzazione di un riepilogo del servizio per una policy collegata a un utente

Puoi visualizzare i riepiloghi dei servizi per qualsiasi policy collegata a un utente IAM.

Per visualizzare il riepilogo del servizio per una policy collegata a un utente

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per l'utente, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate all'utente direttamente o da un gruppo.
5. Nella tabella di policy per l'utente, scegli il nome della policy che si desidera visualizzare.

   Se nella pagina Utenti si sceglie di visualizzare il riepilogo dei servizi per una policy collegata a tale utente, si viene reindirizzati alla pagina Policy. È possibile visualizzare i riepiloghi dei servizi solo nella pagina Policy.

6. Scegli Riepilogo. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.

> **Note**
>
> Se la policy selezionata è una policy inline collegata direttamente all'utente, appare la tabella di riepilogo del servizio. Se la policy è una policy inline collegata da un gruppo, si passa al documento della policy JSON per quel gruppo. Se la policy è una policy gestita, si viene reindirizzati al riepilogo del servizio per quella policy nella pagina Policies (Policy).

Visualizzazione di un riepilogo del servizio per una policy collegata a un ruolo

È possibile visualizzare il riepilogo della policy per qualsiasi policy collegata a un ruolo.

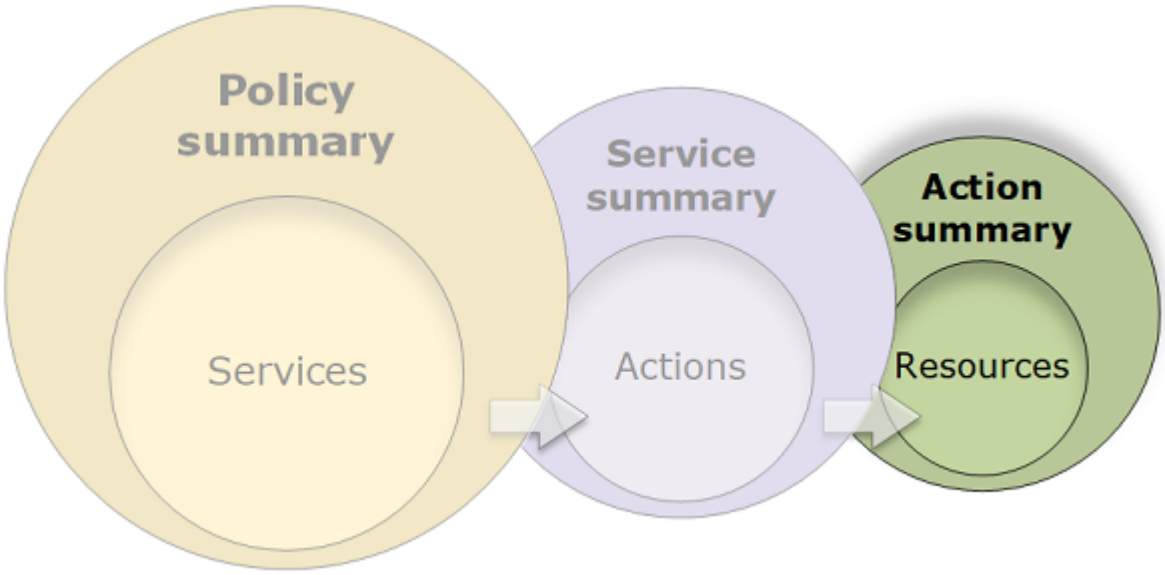
Per visualizzare il riepilogo del servizio per una policy collegata a un ruolo

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Roles (Ruoli) dal riquadro di navigazione.
3. Nell'elenco di ruoli, selezionare il nome del ruolo la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per il ruolo, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate al ruolo.
5. Nella tabella di policy per il ruolo, scegli il nome della policy che si desidera visualizzare.

   Se nella pagina Ruoli si sceglie di visualizzare il riepilogo dei servizi per una policy collegata a tale utente, si viene reindirizzati alla pagina Policy. È possibile visualizzare i riepiloghi dei servizi solo nella pagina Policy.
6. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.

Riepilogo delle operazioni (elenco di risorse)

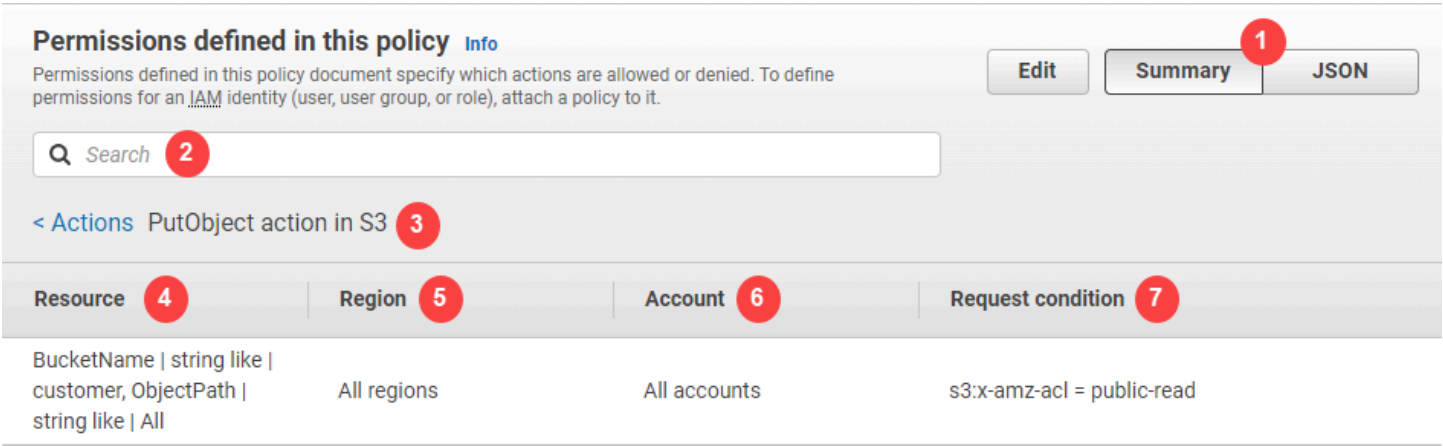
Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione. La tabella del riepilogo dell'operazione include un elenco di risorse e le condizioni associate che si applicano a un'operazione prescelta.



Per visualizzare un riepilogo dell'operazione per ciascuna operazione che consente le autorizzazioni, selezionare il collegamento nel riepilogo dei servizi. La tabella di riepilogo dell'operazione include dettagli sulla risorsa, compresi la Region (Regione) e l'Account. È possibile anche visualizzare le condizioni applicabili a ogni risorsa. Questo illustra le condizioni che si applicano ad alcune risorse ma non altre.

Informazioni sugli elementi del riepilogo di un'operazione

L'esempio seguente è il riepilogo dell'operazione (di scrittura) `PutObject` dal riepilogo del servizio Amazon S3 (consulta Riepilogo del servizio (elenco di operazioni)). Per questa operazione, la policy definisce più condizioni su una singola risorsa.



La pagina di riepilogo dell'operazione include le informazioni riportate di seguito:

1. Seleziona JSON per visualizzare ulteriori dettagli sulla policy, ad esempio le condizioni multiple applicate alle operazioni. Se stai visualizzando il riepilogo del servizio per una policy in linea collegata direttamente a un utente, la procedura potrebbe differire. Per accedere al documento di policy JSON in questo caso, è necessario chiudere la finestra del dialogo di riepilogo delle operazioni e tornare al riepilogo della policy.)
2. Per visualizzare il riepilogo per una risorsa specifica, digita le parole chiave nella casella Cerca per ridurre l'elenco delle risorse disponibili.
3. Accanto alla freccia indietro delle azioni viene visualizzato il nome del servizio e dell'azione nel formato `action name action in service` (in questo caso PutObject l'azione in S3). Il riepilogo dell'operazione per questo servizio include l'elenco delle risorse definite nella policy.
4. Risorsa: questa colonna elenca le risorse che la policy definisce per il servizio scelto. In questo esempio, l'PutObjectazione è consentita su tutti i percorsi degli oggetti, ma solo sulla risorsa bucket `developer_bucket` Amazon S3. A seconda delle informazioni che il servizio fornisce a IAM, è possibile che venga visualizzato un ARN come `arn:aws:s3:::developer_bucket/*`, oppure il tipo di risorsa definita, come `BucketName = developer_bucket, ObjectPath = All`.
5. Regione: questa colonna mostra la regione in cui la risorsa viene definita. Le risorse possono essere definite per tutte le regioni o per una singola regione. Non possono esistere in più di una regione specifica.
   - Tutte le regioni: le operazioni associate alla risorsa si applicano a tutte le regioni. In questo esempio, l'operazione appartiene a un servizio globale, Amazon S3. Le operazioni che appartengono a servizi globali si applicano a tutte le regioni.
   - Testo regione: le operazioni associate alla risorsa si applicano a una regione. Ad esempio, una policy può specificare la regione `us-east-2` per una risorsa.
6. Account: questa colonna indica se i servizi o le operazioni associati alla risorsa si applicano a un determinato account. Le risorse possono esistere in tutti gli account o in un singolo account. Non possono esistere in più di un determinato account.
   - Tutti gli account: le operazioni associate alla risorsa si applicano a tutti gli account. In questo esempio, l'operazione appartiene a un servizio globale, Amazon S3. Le operazioni che appartengono a servizi globali si applicano a tutti gli account.
   - Questo account: le operazioni associate alla risorsa si applicano solo all'account corrente.
   - Numero di account: le operazioni associate alla risorsa si applicano a un account (uno al quale non è stato effettuato l'accesso). Ad esempio, se una policy specifica l'account`123456789012` per una risorsa, quindi il numero di account viene visualizzato nel riepilogo della policy.

7. Condizione richiesta: questa colonna mostra se le operazioni associate alla risorsa sono soggette a condizioni. Questo esempio include la condizione `s3:x-amz-acl = public-read`. Per ulteriori informazioni su queste condizioni, seleziona JSON per visualizzare il documento della policy JSON.

Visualizzazione dei riepiloghi delle azioni

È possibile visualizzare un riepilogo delle azioni per ogni azione riportata nel riepilogo della policy che concede le autorizzazioni.

Visualizzazione dei riepiloghi delle azioni dalla pagina Policy

È possibile visualizzare il riepilogo delle azioni per le policy gestite.

Per visualizzare il riepilogo delle operazioni per una policy gestita

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
4. Nella pagina Dettagli della policy per la policy, visualizza la scheda Autorizzazioni per consultare il riepilogo della policy.
5. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.
6. Nell'elenco delle operazioni del riepilogo del servizio, selezionare il nome dell'operazione che si desidera visualizzare.

Visualizzazione dei riepiloghi delle azioni per una policy collegata a un utente

È possibile visualizzare il riepilogo delle azioni per qualsiasi policy collegata a un utente.

Per visualizzare il riepilogo dell'operazione per una policy collegata a un utente

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Users (Utenti) dal riquadro di navigazione.
3. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare.

4. Nella pagina Summary (Riepilogo) per l'utente, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate all'utente direttamente o da un gruppo.
5. Nella tabella di policy per l'utente, scegli il nome della policy che si desidera visualizzare.

   Se nella pagina Utenti si sceglie di visualizzare il riepilogo dei servizi per una policy collegata a tale utente, si viene reindirizzati alla pagina Policy. È possibile visualizzare i riepiloghi dei servizi solo nella pagina Policy.
6. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.

   > **Note**
   >
   > Se la policy selezionata è una policy inline collegata direttamente all'utente, appare la tabella di riepilogo del servizio. Se la policy è una policy inline collegata da un gruppo, si passa al documento della policy JSON per quel gruppo. Se la policy è una policy gestita, si viene reindirizzati al riepilogo del servizio per quella policy nella pagina Policies (Policy).
7. Nell'elenco delle operazioni del riepilogo del servizio, selezionare il nome dell'operazione che si desidera visualizzare.

Visualizzazione dei riepiloghi delle azioni per una policy collegata a un ruolo

È possibile visualizzare il riepilogo delle azioni per qualsiasi policy collegata a un ruolo.

Per visualizzare il riepilogo dell'operazione per una policy collegata a un ruolo

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Ruoli.
3. Nell'elenco di ruoli, selezionare il nome del ruolo la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per il ruolo, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate al ruolo.
5. Nella tabella di policy per il ruolo, scegli il nome della policy che si desidera visualizzare.

Se nella pagina Ruoli si sceglie di visualizzare il riepilogo dei servizi per una policy collegata a tale utente, si viene reindirizzati alla pagina Policy. È possibile visualizzare i riepiloghi dei servizi solo nella pagina Policy.

6. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.
7. Nell'elenco delle operazioni del riepilogo del servizio, selezionare il nome dell'operazione che si desidera visualizzare.

Esempi di riepiloghi di policy

Gli esempi seguenti includono le policy JSON con i relativi riepiloghi di policy, i riepiloghi dei servizi e i riepiloghi delle operazioni, per aiutarti a comprendere le autorizzazioni concesse tramite una policy.

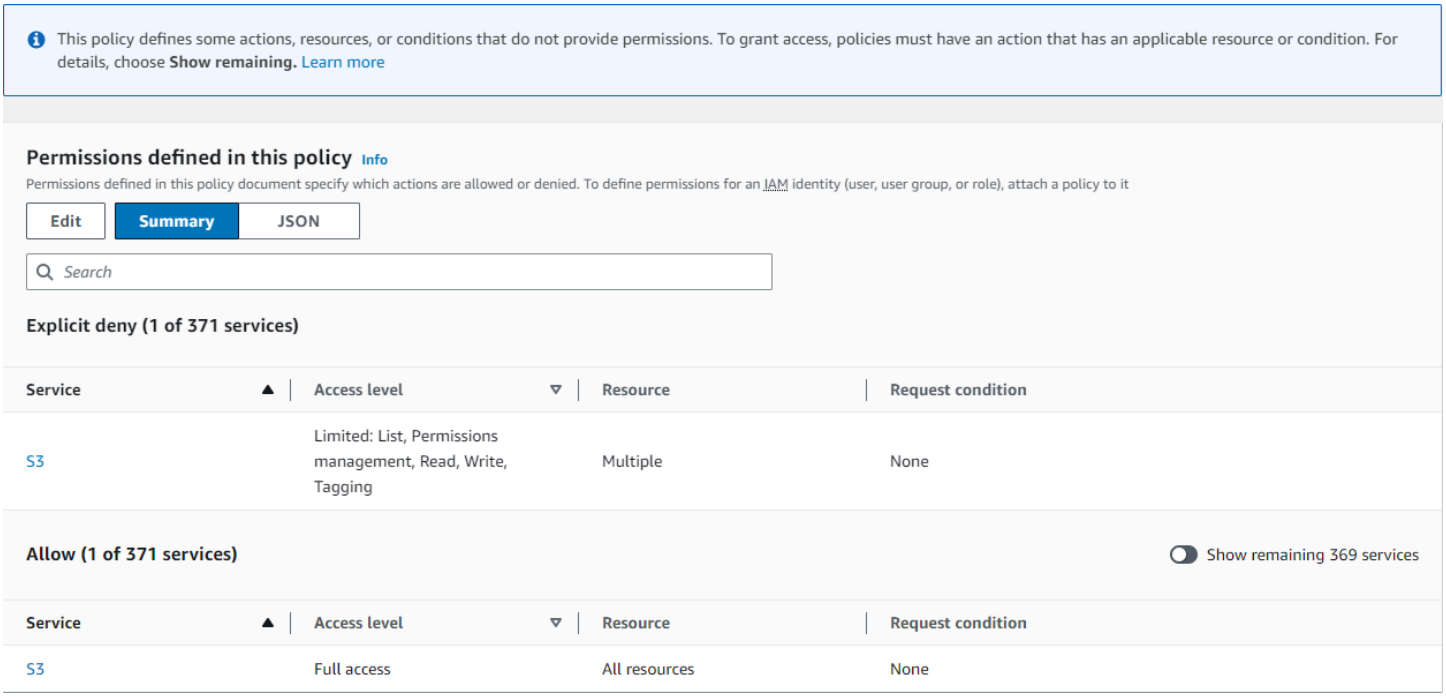
Politica 1: DenyCustomerBucket

Questa policy illustra un permesso e un rifiuto per lo stesso servizio.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
            "Action": ["s3:*"],
            "Resource": ["*"]
        },
        {
            "Sid": "DenyCustomerBucket",
            "Action": ["s3:*"],
            "Effect": "Deny",
            "Resource": ["arn:aws:s3:::customer", "arn:aws:s3:::customer/*" ]
        }
    ]
}
```

DenyCustomerBucketRiepilogo della politica:



DenyCustomerBucket Riepilogo del servizio S3 (Explicit deny):

< Services Actions in S3 (82 of 130) Show remaining 48 actions

Read (35 of 53)

Action	Resource	Request condition
GetAccelerateConfiguration	BucketName| string like |customer	None
GetAnalyticsConfiguration	BucketName| string like |customer	None
GetBucketAcl	BucketName| string like |customer	None
GetBucketCORS	BucketName| string like |customer	None
GetBucketLocation	BucketName| string like |customer	None
GetBucketLogging	BucketName| string like |customer	None
GetBucketNotification	BucketName| string like |customer	None
GetBucketObjectLockConfiguration	BucketName| string like |customer	None
GetBucketOwnershipControls	BucketName| string like |customer	None
GetBucketPolicy	BucketName| string like |customer	None
GetBucketPolicyStatus	BucketName| string like |customer	None
GetBucketPublicAccessBlock	BucketName| string like |customer	None
GetBucketRequestPayment	BucketName| string like |customer	None
GetBucketTagging	BucketName| string like |customer	None
GetBucketVersioning	BucketName| string like |customer	None
GetBucketWebsite	BucketName| string like |customer	None

GetObject (Leggi) Riepilogo delle azioni:

< Actions GetObject action in S3

Resource	Region	Account	Request condition
BucketName| string like |customer, ObjectPath| string like |All	-	All accounts	None

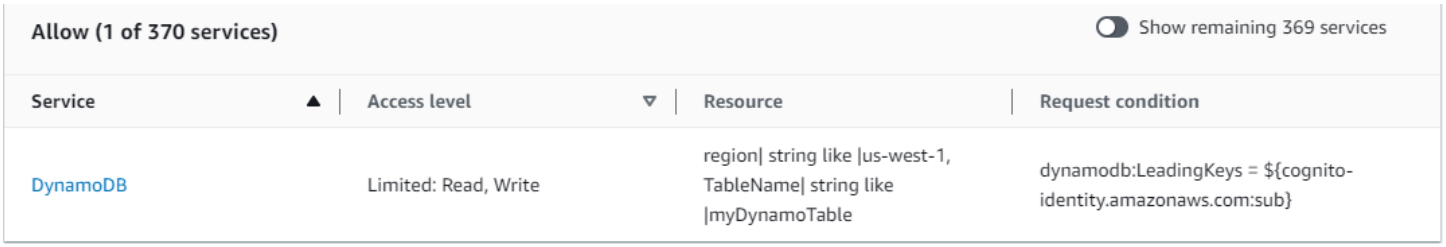
Politica 2: DynamoDbRowCognitoID

Questa policy consente l'accesso a livello di riga ad Amazon DynamoDB in base all'ID Amazon Cognito dell'utente.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:UpdateItem"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-west-1:123456789012:table/myDynamoTable"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:LeadingKeys": [
                        "${cognito-identity.amazonaws.com:sub}"
                    ]
                }
            }
        }
    ]
}
```

DynamoDbRowCognitoIDRiepilogo della politica:



DynamoDbRowCognitoID Riepilogo del servizio DynamoDB (Allow):

< Services Actions in DynamoDB (4 of 65) Show remaining 61 actions

Read (1 of 26)

Action	Resource	Request condition
GetItem	region| string like |us-west-1, TableName| string like |myDynamoTable	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}

Write (3 of 33)

Action	Resource	Request condition
DeleteItem	region| string like |us-west-1, TableName| string like |myDynamoTable	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}
PutItem	region| string like |us-west-1, TableName| string like |myDynamoTable	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}
UpdateItem	region| string like |us-west-1, TableName| string like |myDynamoTable	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}

GetItem (Elenco) Riepilogo delle azioni:

< Actions GetItem action in DynamoDB

Resource	Region	Account	Request condition
region| string like |us-west-1, TableName| string like |myDynamoTable	us-west-1	123456789012	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}

Politica 3: MultipleResourceCondition

Questa policy include più risorse e condizioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": ["arn:aws:s3:::Apple_bucket/*"],
            "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": ["arn:aws:s3:::Orange_bucket/*"],
            "Condition": {"StringEquals": {
                "s3:x-amz-acl": ["custom"],
                "s3:x-amz-grant-full-control": ["1234"]
            }}
        }
    ]
}
```

MultipleResourceConditionRiepilogo della politica:

Allow (1 of 370 services) — Show remaining 369 services

Service	Access level	Resource	Request condition
S3	Limited: Permissions management, Write	Multiple	Multiple

MultipleResourceCondition Riepilogo del servizio S3 (Consenti):

< Services Actions in S3 (2 of 130) — Show remaining 128 actions

Write (1 of 47)

Action	Resource	Request condition
PutObject	Multiple	Multiple

Permission Management (1 of 15)

Action	Resource	Request condition
PutObjectAcl	Multiple	Multiple

PutObject (Scrivi) Riepilogo delle azioni:

< Actions PutObject action in S3

Resource	Region	Account	Request condition
Multiple	-	All accounts	Multiple

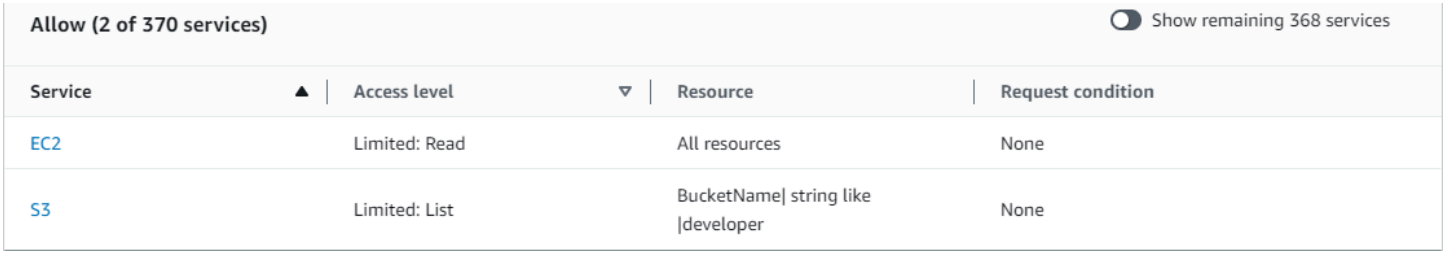
Policy 4: EC2_troubleshoot

La policy seguente permette agli utenti di ottenere uno screenshot di un'istanza Amazon EC2 in esecuzione, che può essere utile per la risoluzione dei problemi di EC2. Questa policy permette inoltre di visualizzare le informazioni sugli elementi nel bucket degli sviluppatori di Amazon S3.

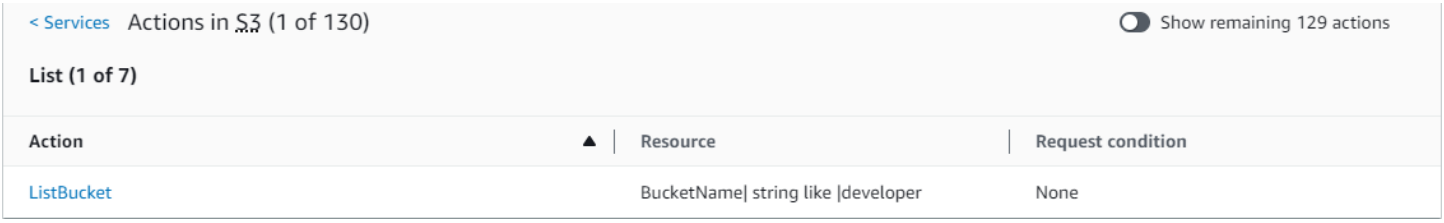
JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetConsoleScreenshot"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::developer"
            ]
        }
    ]
}
```

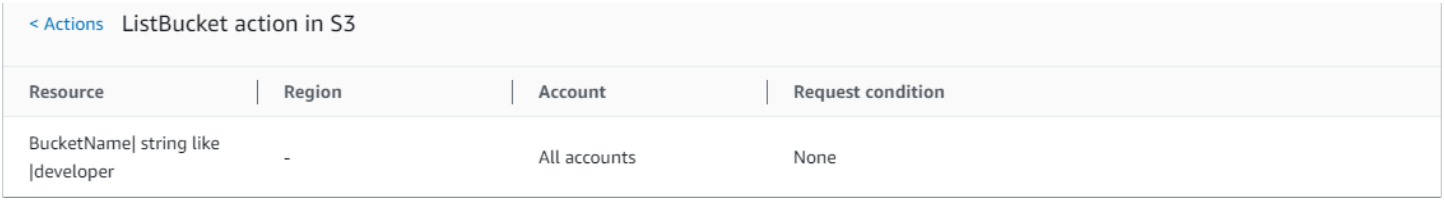
Riepilogo della policy EC2_Troubleshoot:



Riepilogo del servizio EC2_Troubleshoot S3 (permesso):



ListBucket (Elenco) Riepilogo delle azioni:



Politica 5: _ _ CodeBuild CodeCommit CodeDeploy

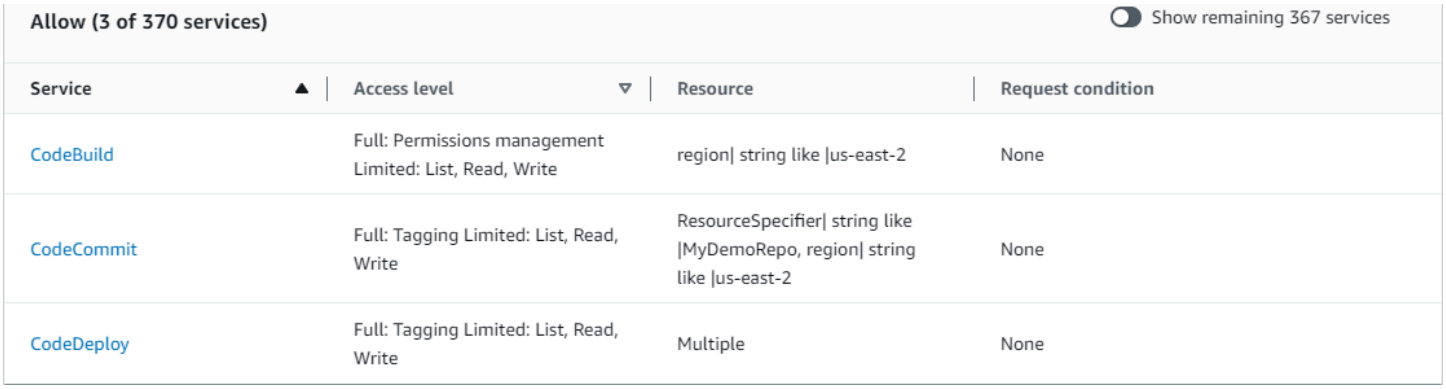
Questa politica fornisce l'accesso a CodeDeploy risorse specifiche CodeBuild e. CodeCommit Poiché queste risorse sono specifiche di ogni servizio, vengono visualizzate solo con il servizio corrispondente. Se includi una risorsa che non corrisponde ad alcun servizio nell'elemento `Action`, la risorsa viene visualizzata in tutti i riepiloghi delle operazioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1487980617000",
            "Effect": "Allow",
            "Action": [
                "codebuild:*",
                "codecommit:*",
                "codedeploy:*"
            ],
            "Resource": [
                "arn:aws:codebuild:us-east-2:123456789012:project/my-demo-
project",
                "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo",
                "arn:aws:codedeploy:us-
east-2:123456789012:application:WordPress_App",
                "arn:aws:codedeploy:us-east-2:123456789012:instance/AssetTag*"
```

```
            ]
        }
    ]
}
```

CodeBuild_ CodeCommit _ Riepilogo CodeDeploy della politica:



Allow (3 of 370 services) — Show remaining 367 services

Service	Access level	Resource	Request condition
CodeBuild	Full: Permissions management Limited: List, Read, Write	region| string like |us-east-2	None
CodeCommit	Full: Tagging Limited: List, Read, Write	ResourceSpecifier| string like |MyDemoRepo, region| string like |us-east-2	None
CodeDeploy	Full: Tagging Limited: List, Read, Write	Multiple	None

CodeBuild_ CodeCommit _ CodeDeploy CodeBuild (Consenti) Riepilogo del servizio:

< Services Actions in CodeBuild (24 of 53) Show remaining 29 actions

Read (4 of 9)

Action	Resource	Request condition
BatchGetBuildBatches	region| string like |us-east-2	None
BatchGetBuilds	region| string like |us-east-2	None
BatchGetProjects	region| string like |us-east-2	None
GetResourcePolicy	region| string like |us-east-2	None

Write (16 of 28)

Action	Resource	Request condition
BatchDeleteBuilds	region| string like |us-east-2	None
CreateProject	region| string like |us-east-2	None
CreateWebhook	region| string like |us-east-2	None
DeleteBuildBatch	region| string like |us-east-2	None
DeleteProject	region| string like |us-east-2	None
DeleteWebhook	region| string like |us-east-2	None
InvalidateProjectCache	region| string like |us-east-2	None
RetryBuild	region| string like |us-east-2	None
RetryBuildBatch	region| string like |us-east-2	None
StartBuild	region| string like |us-east-2	None
StartBuildBatch	region| string like |us-east-2	None
StopBuild	region| string like |us-east-2	None
StopBuildBatch	region| string like |us-east-2	None
UpdateProject	region| string like |us-east-2	None
UpdateProjectVisibility	region| string like |us-east-2	None
UpdateWebhook	region| string like |us-east-2	None

List (2 of 14)

CodeBuild_ CodeCommit _ CodeDeploy StartBuild (Scrivi) Riepilogo delle azioni:

< Actions StartBuild action in CodeBuild

Resource	Region	Account	Request condition
region| string like |us-east-2	us-east-2	123456789012	None

Autorizzazioni necessarie per accedere alle risorse IAM

Le risorse sono oggetti all'interno di un servizio. Le risorse IAM includono gruppi, utenti, ruoli e policy. Se hai effettuato l'accesso con Utente root dell'account AWS le credenziali, non hai restrizioni sull'amministrazione delle credenziali IAM o delle risorse IAM. Tuttavia, agli utenti IAM devono essere esplicitamente concesse le autorizzazioni appropriate per amministrare credenziali o risorse IAM. A tale scopo, è possibile collegare all'utente una policy basata su identità.

> **Note**
>
> In tutta la AWS documentazione, quando facciamo riferimento a una policy IAM senza menzionare nessuna delle categorie specifiche, intendiamo una policy basata sull'identità e gestita dal cliente. Per ulteriori informazioni sulle categorie di policy, consultare the section called "Policy e autorizzazioni".

Autorizzazioni per amministrare le identità IAM

Le autorizzazioni necessarie per amministrare gruppi, utenti, ruoli e credenziali IAM in genere corrispondono alle operazioni API per l'attività. Ad esempio, per creare utenti IAM, è necessario disporre dell'autorizzazione `iam:CreateUser` che corrisponde al comando API: `CreateUser`. Per consentire a un utente IAM di creare altri utenti IAM, è possibile collegare una policy AM come la seguente all'utente specificato:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:CreateUser",
    "Resource": "*"
  }
}
```

In una policy, il valore dell'elemento `Resource` dipende dall'operazione e da quali risorse possono essere interessate dall'operazione. Nell'esempio precedente, la policy consente a un utente di

creare qualsiasi utente (* è un carattere jolly che corrisponde a tutte le stringhe). Per contro, una policy che consente agli utenti di modificare solo le proprie chiavi di accesso (operazioni API `CreateAccessKey` e `UpdateAccessKey`) in genere include un elemento `Resource`. In questo caso l'ARN include una variabile (`${aws:username}`) che viene risolta nel nome dell'utente corrente, come nell'esempio seguente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListUsersForConsole",
            "Effect": "Allow",
            "Action": "iam:ListUsers",
            "Resource": "arn:aws:iam::*:*"
        },
        {
            "Sid": "ViewAndUpdateAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:UpdateAccessKey",
                "iam:CreateAccessKey",
                "iam:ListAccessKeys"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

Nell'esempio precedente, `${aws:username}` è una variabile che viene risulta nel nome utente dell'utente corrente. Per ulteriori informazioni sulle variabili di policy, consultare Elementi delle policy IAM: variabili e tag.

L'utilizzo di un carattere jolly (*) nel nome dell'operazione spesso facilita la concessione delle autorizzazioni per tutte le operazioni correlate a un'attività specifica. Ad esempio, per consentire agli utenti di eseguire qualsiasi operazione IAM, puoi utilizzare `iam:*` per l'operazione. Per consentire agli utenti di eseguire qualsiasi operazione correlata solo alle chiavi di accesso, puoi utilizzare `iam:*AccessKey*` nell'elemento `Action` di un'istruzione della policy. In questo modo l'utente

ottiene l'autorizzazione per eseguire le operazioni `CreateAccessKey`, `DeleteAccessKey`, `GetAccessKeyLastUsed`, `ListAccessKeys` e `UpdateAccessKey`. (Se in futuro verrà aggiunta a IAM un'azione con "AccessKey" nel nome, l'utilizzo dell'`Action`elemento darà anche all'utente il permesso `iam:*AccessKey*` per quella nuova azione.) L'esempio seguente mostra una policy che consente agli utenti di eseguire tutte le azioni relative alle proprie chiavi di accesso (sostituirle *`account-id`* con il proprio Account AWS ID):

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "iam:*AccessKey*",
        "Resource": "arn:aws:iam::111122223333:user/${aws:username}"
    }
}
```

Alcune attività, ad esempio l'eliminazione di un gruppo, coinvolgono più operazioni: devi prima rimuovere gli utenti dal gruppo, quindi scollegare o eliminare le policy del gruppo e quindi effettivamente eliminare il gruppo. Se desideri che un utente sia in grado di eliminare un gruppo, devi concedere all'utente le autorizzazioni necessarie per eseguire tutte le operazioni correlate.

Autorizzazioni per lavorare in Console di gestione AWS

I precedenti esempi mostrano le policy che consentono a un utente di eseguire le operazioni con l'AWS CLIo gli SDK AWS.

Quando gli utenti utilizzano la console, questa emette richieste a IAM per elencare i gruppi, gli utenti, i ruoli e le policy e per ottenere le policy associate a un gruppo, un utente o un ruolo. La console invia anche richieste per ottenere Account AWS informazioni e informazioni sul preside. Il principale è l'utente che effettua le richieste nella console.

In generale, per eseguire un'operazione, è solo necessario che l'operazione corrispondente sia inclusa in una policy. Per creare un utente, è necessaria l'autorizzazione per chiamare l'operazione `CreateUser`. Spesso, quando utilizzi la console per eseguire un'operazione, devi disporre delle autorizzazioni per mostrare, elencare, ottenere o altrimenti visualizzare le risorse nella console. Ciò è necessario per poter navigare nella console allo scopo di eseguire l'operazione specificata.

Ad esempio, se l'utente Jorge desidera utilizzare la console per modificare le proprie chiavi di accesso, passa alla console IAM e sceglie Utenti. Questa operazione determina l'esecuzione di una richiesta `ListUsers` da parte della console. Se Jorge non dispone dell'autorizzazione per l'operazione `iam:ListUsers`, alla console viene negato l'accesso quando cerca di elencare gli utenti. Di conseguenza, Jorge non può accedere al proprio nome e alle proprie chiavi di accesso, anche se dispone delle autorizzazioni per le operazioni `CreateAccessKey` e `UpdateAccessKey`.

Se desideri concedere agli utenti le autorizzazioni per amministrare gruppi, utenti, ruoli, politiche e credenziali con Console di gestione AWS, devi includere le autorizzazioni per le azioni eseguite dalla console. Per alcuni esempi di policy che è possibile utilizzare per concedere a un utente tali autorizzazioni, consultare Esempi di policy per amministrare le risorse IAM.

Concedi le autorizzazioni a tutti AWS account

È possibile concedere direttamente agli utenti IAM dell'account l'accesso alle risorse. Se gli utenti di un altro account devono accedere alle risorse, è possibile creare un ruolo IAM ovvero un'entità che include le autorizzazioni, ma che non è associato a un utente specifico. Gli utenti di altri account possono utilizzare il ruolo e accedere alle risorse in base alle autorizzazioni assegnate al ruolo. Per ulteriori informazioni, consulta Accesso per un utente IAM in un altro Account AWS che possiedi.

> **Note**
>
> Alcuni servizi supportano le policy basate sulle risorse, come descritto in Identity-based politiche e politiche basate sulle risorse (ad esempio Amazon S3, Amazon SNS e Amazon SQS). Per tali servizi, un'alternativa all'utilizzo dei ruoli consiste nel collegare una policy alla risorsa (bucket, argomento o coda) che si desidera condividere. La politica basata sulle risorse può specificare l' AWS account che dispone delle autorizzazioni per accedere alla risorsa.

Autorizzazioni per consentire a un servizio di accedere a un altro servizio

Molti AWS servizi accedono ad altri servizi. AWS Ad esempio, diversi servizi AWS , tra cui Amazon EMR, Elastic Load Balancing e Amazon EC2 Auto Scaling, gestiscono le istanze Amazon EC2. Altri AWS servizi utilizzano bucket Amazon S3, argomenti Amazon SNS, code Amazon SQS e così via.

Lo scenario per gestire le autorizzazioni in questi casi varia a seconda del servizio. Di seguito sono elencati alcuni esempi di come gestire le autorizzazioni per differenti servizi:

- In Amazon EC2 Auto Scaling, gli utenti devono disporre dell'autorizzazione per utilizzare Auto Scaling, ma non hanno bisogno dell'autorizzazione esplicita per gestire le istanze Amazon EC2.
- In AWS Data Pipeline, un ruolo IAM determina cosa può fare una pipeline; gli utenti devono avere l'autorizzazione per assumere il ruolo. Per maggiori dettagli, consulta Concessione di autorizzazioni per le pipeline con IAM nella Guida per gli sviluppatori di AWS Data Pipeline .

Per dettagli su come configurare correttamente le autorizzazioni in modo che un AWS servizio sia in grado di svolgere le attività previste, consulta la documentazione del servizio che stai chiamando. Per informazioni su come creare un ruolo per un servizio, consultare Crea un ruolo per delegare le autorizzazioni a un AWS service.

Configurazione di un servizio con un ruolo IAM in modo che funzioni per tuo conto

Quando desideri configurare un AWS servizio in modo che funzioni per tuo conto, in genere fornisci l'ARN per un ruolo IAM che definisce ciò che il servizio è autorizzato a fare. AWS verifica che tu disponga delle autorizzazioni per passare un ruolo a un servizio. Per ulteriori informazioni, consulta Concedi a un utente le autorizzazioni per passare un ruolo a un AWS service.

Operazioni necessarie

Le operazioni sono le azioni che puoi effettuare su una risorsa, come la visualizzazione, la creazione, la modifica e l'eliminazione di tale risorsa. Le azioni sono definite da ciascun AWS servizio.

Per consentire a qualcuno di eseguire un'operazione, è necessario includere le operazioni necessarie in una policy da applicare all'identità chiamante o alla risorsa interessata. In generale, per fornire le autorizzazioni necessarie per eseguire un'operazione, devi includere tale operazione nella policy. Ad esempio, per creare un utente, devi aggiungere l' CreateUser azione alla tua policy.

In alcuni casi, potrebbe essere necessario includere nella policy ulteriori azioni correlate per eseguire una determinata operazione. Ad esempio, per fornire a qualcuno l'autorizzazione per creare una directory in Servizio di directory AWS utilizzando l'operazione `ds:CreateDirectory`, devi includere le seguenti operazioni nella relativa policy:

- `ds:CreateDirectory`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `ec2:CreateSecurityGroup`

- `ec2:CreateNetworkInterface`
- `ec2:DescribeNetworkInterfaces`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:AuthorizeSecurityGroupEgress`

Quando crei o modifichi una policy utilizzando l'editor grafico, ricevi avvisi e richieste che ti aiutano a selezionare tutte le operazioni necessarie per la policy.

Per ulteriori informazioni sulle autorizzazioni necessarie per creare una directory in Servizio di directory AWS, vedere Esempio 2: consentire a un utente di creare una directory.

Esempi di policy per amministrare le risorse IAM

Di seguito sono riportati gli esempi delle policy IAM che consentono agli utenti di eseguire attività associate alla gestione di utenti, gruppi e credenziali IAM. Queste includono le policy che consentono agli utenti di gestire le proprie password, le chiavi di accesso e i dispositivi per la multi-factor authentication (MFA).

Per esempi di policy che consentono agli utenti di eseguire attività con altri AWS servizi, come Amazon S3, Amazon EC2 e DynamoDB, consulta. Esempi di policy basate su identità IAM

Argomenti

- Consentire a un utente di elencare i gruppi, gli utenti e le policy dell'account e altro per scopi di report.
- Consentire a un utente di gestire l'appartenenza del gruppo
- Consentire a un utente di gestire gli utenti IAM
- Consentire agli utenti di impostare una policy per la password dell'account
- Consentire agli utenti di generare e recuperare i report delle credenziali IAM
- Consentire tutte le operazioni IAM (accesso amministratore)

Consentire a un utente di elencare i gruppi, gli utenti e le policy dell'account e altro per scopi di report.

La policy seguente consente all'utente di chiamare qualsiasi operazione IAM che inizi con la stringa `Get` o `List` e generare i report. Per visualizzare la policy di esempio, consulta IAM: consente l'accesso in sola lettura alla console IAM.

Consentire a un utente di gestire l'appartenenza del gruppo

La seguente policy consente all'utente di aggiornare l'appartenenza al gruppo chiamato. MarketingGroup Per visualizzare la policy di esempio, consulta IAM: consente di gestire l'appartenenza di un gruppo a livello di programmazione e nella console.

Consentire a un utente di gestire gli utenti IAM

La policy seguente consente a un utente di eseguire tutte le attività associate alla gestione degli utenti IAM ma non di eseguire le operazioni su altre entità, come ad esempio la creazione di gruppi o policy. Le operazioni consentite includono le seguenti:

- Creazione dell'utente (l'operazione `CreateUser`).
- Eliminazione dell'utente. Questa operazione richiede le autorizzazioni per eseguire tutte le seguenti operazioni: `DeleteSigningCertificate`, `DeleteLoginProfile`, `RemoveUserFromGroup` e `DeleteUser`.
- Elencare gli utenti nell'account e nei gruppi (le operazioni `GetUser`, `ListUsers` e `ListGroupsForUser`).
- Elencare e rimuovere le policy per l'utente (le operazioni `ListUserPolicies`, `ListAttachedUserPolicies`, `DetachUserPolicy` e `DeleteUserPolicy`)
- Rinominare o modificare il percorso per l'utente (l'operazione `UpdateUser`). L'elemento `Resource` deve includere un ARN che copre sia il percorso di origine sia il percorso di destinazione. Per ulteriori informazioni sui percorsi, consultare la pagina Nomi descrittivi e percorsi.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUsersToPerformUserActions",
            "Effect": "Allow",
            "Action": [
                "iam:ListPolicies",
                "iam:GetPolicy",
                "iam:UpdateUser",
                "iam:AttachUserPolicy",
                "iam:ListEntitiesForPolicy",
```

```
                "iam:DeleteUserPolicy",
                "iam:DeleteUser",
                "iam:ListUserPolicies",
                "iam:CreateUser",
                "iam:RemoveUserFromGroup",
                "iam:AddUserToGroup",
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:PutUserPolicy",
                "iam:ListAttachedUserPolicies",
                "iam:ListUsers",
                "iam:GetUser",
                "iam:DetachUserPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUsersToSeeStatsOnIAMConsoleDashboard",
            "Effect": "Allow",
            "Action": [
                "iam:GetAccount*",
                "iam:ListAccount*"
            ],
            "Resource": "*"
        }
    ]
}
```

Un numero di autorizzazioni incluse nella policy precedente consente all'utente di eseguire attività nella Console di gestione AWS. Gli utenti che eseguono attività correlate all'utente dalla AWS CLI, gli SDK AWS o dalle API di query HTTP IAM potrebbero non avere bisogno di determinate autorizzazioni. Ad esempio, se gli utenti conoscono già l'ARN delle policy per distaccarsi da un utente, non hanno bisogno dell'autorizzazione `iam:ListAttachedUserPolicies`. L'elenco esatto delle autorizzazioni che un utente richiede, dipende dalle attività che l'utente deve eseguire durante la gestione di altri utenti.

I seguenti permessi nella policy consentono l'accesso alle attività dell'utente tramite la Console di gestione AWS:

- `iam:GetAccount*`
- `iam:ListAccount*`

Consentire agli utenti di impostare una policy per la password dell'account

Potresti fornire ad alcuni utenti le autorizzazioni per ottenere e aggiornare la password policy (policy della password) del tuo Account AWS. Per visualizzare la policy di esempio, consulta IAM: consente l'impostazione dei requisiti della password dell'account a livello di programmazione e nella console.

Consentire agli utenti di generare e recuperare i report delle credenziali IAM

Puoi concedere agli utenti il permesso di generare e scaricare un rapporto che elenca tutti gli utenti del tuo Account AWS. Il report elenca inoltre lo stato di varie credenziali utente, tra cui le password, le chiavi di accesso, i dispositivi MFA e i certificati di firma. Per ulteriori informazioni sui report delle credenziali, consultare la pagina Genera report sulle credenziali per i tuoi Account AWS. Per visualizzare la policy di esempio, consulta IAM: generazione e recupero di report di credenziali IAM.

Consentire tutte le operazioni IAM (accesso amministratore)

È possibile fornire ad alcuni utenti le autorizzazioni amministrative per eseguire tutte le operazioni in IAM, tra cui la gestione delle password, le chiavi di accesso, i dispositivi MFA e i certificati utente. Il seguente esempio di policy concede queste autorizzazioni:

> **Warning**
>
> Quando concedi a un utente l'accesso completo a IAM, non ci sono limiti alle autorizzazioni che l'utente può concedere him/herself o ad altri. L'utente può creare nuove entità IAM (utenti o ruoli) e concedere a quelle entità l'accesso completo a tutte le risorse nel tuo Account AWS. Quando concedi a un utente l'accesso completo a IAM, stai concedendo effettivamente l'accesso completo a tutte le risorse nel tuo Account AWS. Questo include l'accesso a eliminare tutte le risorse. Dovresti concedere queste autorizzazioni solo agli amministratori attendibili e dovresti applicare la multi-factor authentication (MFA) per questi amministratori.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*",
    "Resource": "*"
```

```
  }
}
```

Delegazione temporanea IAM

Panoramica di

La delega temporanea accelera l'onboarding e semplifica la gestione dei prodotti di Amazon e dei AWS partner che si integrano con i tuoi account. AWS Invece di configurare manualmente più AWS servizi, puoi delegare autorizzazioni temporanee e limitate che consentono al fornitore del prodotto di completare le attività di configurazione per tuo conto in pochi minuti attraverso flussi di lavoro di distribuzione automatizzati. Mantieni il controllo amministrativo con requisiti di approvazione e limiti di autorizzazione, mentre le autorizzazioni dei fornitori di prodotti scadono automaticamente dopo la durata approvata senza che sia necessaria una pulizia manuale. Se il prodotto richiede un accesso persistente per le operazioni in corso, il provider può utilizzare la delega temporanea per creare un ruolo IAM con un limite di autorizzazione che definisce le autorizzazioni massime del ruolo. Tutte le attività del fornitore di prodotti vengono tracciate AWS CloudTrail per il monitoraggio della conformità e della sicurezza.

Per gestire le richieste di delega temporanee, puoi avviare una richiesta e rivederle. Per ulteriori informazioni sul monitoraggio, consulta Notifiche e CloudTrail registrazione.

> **Note**
>
> Le richieste di delega temporanee possono essere create solo dai prodotti Amazon e dai AWS partner qualificati che hanno completato il processo di attivazione delle funzionalità. I clienti esaminano e approvano queste richieste ma non possono crearle direttamente. Se sei un AWS partner che desidera integrare la delega temporanea IAM nel tuo prodotto, consulta la Partner Integration Guide per le istruzioni di onboarding e integrazione.

Come funziona la delega temporanea

La delega temporanea consente ad Amazon e AWS Partners di richiedere un accesso temporaneo e limitato al tuo account. Dopo la tua approvazione, possono utilizzare le autorizzazioni delegate per intraprendere azioni per tuo conto. Le richieste di delega definiscono autorizzazioni specifiche per AWS i servizi e le azioni di cui il fornitore del prodotto ha bisogno per distribuire o configurare le

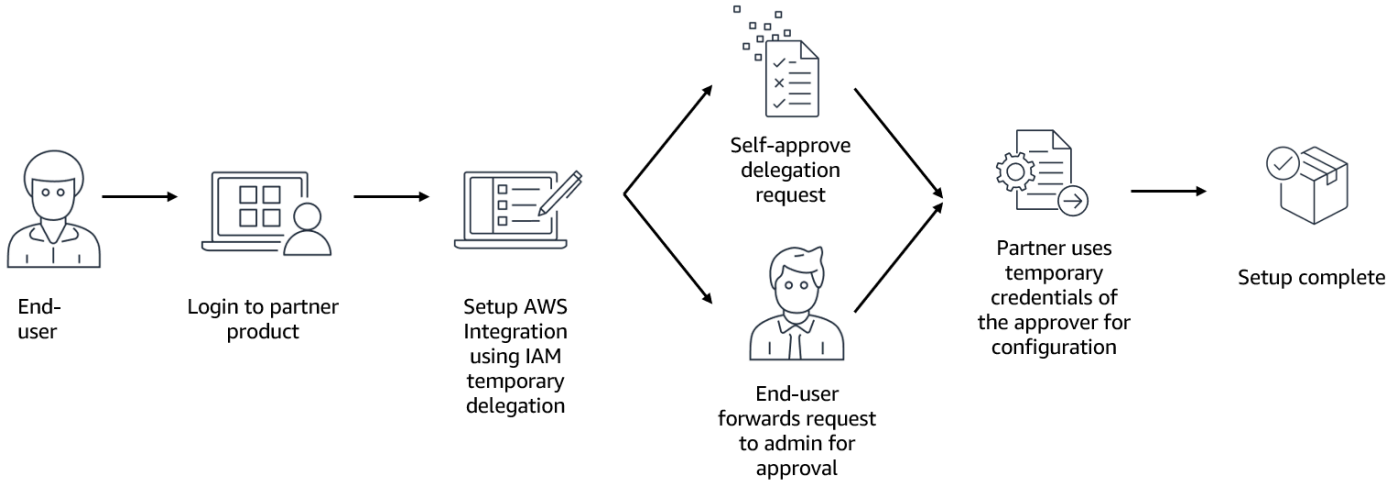
risorse nel tuo account. AWS Queste autorizzazioni sono disponibili solo per un periodo di tempo limitato e scadono automaticamente dopo la durata specificata nella richiesta.

> **Note**
>
> La durata massima per l'accesso delegato è di 12 ore. Tuttavia, gli utenti root possono approvare solo le richieste di delega con una durata pari o inferiore a 4 ore. Se una richiesta specifica più di 4 ore, è necessario utilizzare un'identità non root per approvare la richiesta. Per i dettagli, vedi Funzionalità beta di Permission Simulation.

Per le attività continue, come la lettura da un bucket Amazon S3, le richieste di delega possono includere la creazione di un ruolo IAM che consenta l'accesso continuo alle risorse e alle azioni dopo la scadenza dell'accesso temporaneo. I fornitori di prodotti devono attribuire un limite di autorizzazione a qualsiasi ruolo IAM creato tramite delega temporanea. I limiti di autorizzazione limitano le autorizzazioni massime di un ruolo ma non le concedono autonomamente. Puoi rivedere il limite delle autorizzazioni come parte della richiesta prima di approvarla. Per i dettagli, consulta Limiti delle autorizzazioni.

Di seguito è riportato il procedimento:

1. Accedi a un prodotto Amazon o AWS Partner per integrarlo con il tuo AWS ambiente.
2. Il fornitore del prodotto avvia una richiesta di delega per tuo conto e ti reindirizza alla console di gestione. AWS
3. L'utente esamina le autorizzazioni richieste e decide se approvare, rifiutare o inoltrare la richiesta all'amministratore.
4. Dopo che tu o il tuo amministratore avete approvato la richiesta, il fornitore del prodotto può ottenere le credenziali temporanee dell'approvatore per eseguire le attività richieste.
5. L'accesso del fornitore di prodotti scade automaticamente dopo il periodo di tempo specificato. Tuttavia, qualsiasi ruolo IAM creato tramite la richiesta di delega temporanea persiste oltre questo periodo, consentendo al fornitore del prodotto di continuare ad accedere alle risorse e alle azioni per le attività di gestione in corso.



> **Note**
>
> Puoi delegare le autorizzazioni a un fornitore di prodotti solo se disponi delle autorizzazioni per i servizi e le azioni incluse nella richiesta di delega temporanea. Se non hai accesso ai servizi e alle azioni richiesti, il fornitore del prodotto non riceve queste autorizzazioni quando approvi la richiesta.

Se la verifica delle autorizzazioni mostra che è probabile che abbia esito positivo, puoi approvare la richiesta di delega temporanea e continuare con il flusso di lavoro.

Se la verifica delle autorizzazioni mostra che potresti non disporre di autorizzazioni sufficienti, inoltra la richiesta all'amministratore per l'approvazione. Ti consigliamo di notificare questa richiesta all'amministratore utilizzando il metodo che preferisci, ad esempio un'email o un ticket.

Dopo che l'amministratore ha approvato la richiesta, ciò che accade dopo dipende dalla configurazione del fornitore del prodotto:

- Se il fornitore del prodotto richiede l'accesso immediato, riceve automaticamente le autorizzazioni temporanee e inizia la durata dell'accesso.
- Se il fornitore del prodotto ha richiesto il rilascio da parte del proprietario (destinatario iniziale), è necessario ripetere la richiesta di condivisione esplicita dell'accesso temporaneo all'account prima che inizi la durata dell'accesso. I fornitori di prodotti utilizzano in genere questa opzione quando necessitano di ulteriori input da parte dell'utente, ad esempio la selezione delle risorse o i dettagli di configurazione, per completare l'attività richiesta.

Avviare una richiesta di delega temporanea

Puoi avviare una richiesta di delega temporanea solo da prodotti Amazon o Partner supportati. AWS Durante un flusso di lavoro che supporta la delega temporanea, ti verrà richiesto di concedere al fornitore del prodotto autorizzazioni temporanee e limitate per configurare le risorse richieste AWS nel tuo account. Questo approccio automatizzato offre un'esperienza più semplificata eliminando la necessità di configurare queste risorse manualmente.

Puoi esaminare i dettagli della richiesta di delega, ad esempio i ruoli, le policy e i AWS servizi IAM specifici di cui il fornitore di prodotti ha bisogno prima di concedere l'accesso. Puoi approvare tu stesso la richiesta se disponi di autorizzazioni sufficienti oppure inoltrare la richiesta all'amministratore del tuo account per l'approvazione. Tutti gli accessi ai fornitori di prodotti sono limitati nel tempo e possono essere monitorati e revocati secondo necessità.

Per avviare una richiesta di delega temporanea

1. Accedi alla console di un prodotto supportato da Amazon o dai AWS partner che richiede l'integrazione con il tuo AWS account.
2. Seleziona Implementa con delega temporanea IAM. Tieni presente che il nome dell'opzione può variare a seconda dei prodotti supportati. Per ulteriori informazioni, consulta la documentazione del fornitore del prodotto.

   > **Note**
   >
   > Se non si è già effettuato l'accesso alla AWS Management Console, si apre una nuova finestra nella pagina di AWS accesso. Ti consigliamo di accedere al tuo AWS account prima di avviare la richiesta di delega temporanea dalla console del prodotto. Per ulteriori informazioni su come accedere in base al tipo di utente e alle AWS risorse a cui si desidera accedere, consulta la Guida per l'AWS Sign-In utente.

3. Rivedi i dettagli della richiesta per confermare il nome del prodotto e l' AWS account del fornitore del prodotto. Puoi anche verificare l' AWS identità che il fornitore del prodotto utilizzerà per eseguire azioni per tuo conto.
4. Rivedi i dettagli di accesso per le autorizzazioni che verranno temporaneamente delegate approvando questa richiesta.

- La sezione di riepilogo delle autorizzazioni fornisce una panoramica di alto livello generata dall'intelligenza artificiale che può aiutarti a capire a quali categorie di AWS servizi è possibile accedere e quali tipi di azioni possono essere eseguite in ciascun servizio.
- Scegli Visualizza JSON per esaminare le autorizzazioni specifiche che il fornitore del prodotto deve implementare nel tuo AWS account, inclusi l'ambito di accesso e le limitazioni delle risorse.
- Se il fornitore del prodotto crea un ruolo IAM come parte di una richiesta di delega temporanea, è necessario allegare un limite di autorizzazione al ruolo. Questi ruoli IAM dispongono di autorizzazioni che continuano a consentire l'accesso a risorse e azioni dopo la scadenza della durata di accesso richiesta. Scegli Visualizza dettagli per rivedere il limite delle autorizzazioni, che definisce le autorizzazioni massime che il ruolo può avere. Il fornitore del prodotto applicherà politiche aggiuntive al ruolo durante la creazione che ne definiscono le autorizzazioni effettive. Queste politiche potrebbero apparire più mirate o più ampie rispetto al limite, a seconda di come le definisce il fornitore del prodotto. Tuttavia, il limite delle autorizzazioni garantisce che le autorizzazioni effettive del ruolo non superino mai quelle visualizzate durante l'approvazione della richiesta, indipendentemente dalle politiche associate al ruolo. Per ulteriori informazioni, consulta Limiti delle autorizzazioni.

5. Esamina i risultati della simulazione delle autorizzazioni. La funzionalità di simulazione delle autorizzazioni valuta automaticamente le autorizzazioni della tua identità rispetto a quelle incluse nella richiesta. Sulla base di questa analisi, viene visualizzato un consiglio che indica se approvare la richiesta con l'identità attuale o inoltrare la richiesta a un amministratore. Per i dettagli, vedi Funzionalità beta di Permission Simulation.
6. Nella finestra di dialogo, seleziona come vuoi procedere.

   - Seleziona Consenti l'accesso quando la tua identità dispone di autorizzazioni sufficienti per consentire al fornitore del prodotto di eseguire le procedure di onboarding per tuo conto. Quando si seleziona questa opzione, la durata dell'accesso del fornitore del prodotto inizia dopo aver fornito l'accesso.
   - Seleziona Richiedi approvazione se la tua identità non dispone di autorizzazioni sufficienti per consentire al fornitore del prodotto di eseguire le procedure di onboarding per tuo conto. Quindi, scegli Crea richiesta di approvazione. Quando selezioni questa opzione, viene creato un link temporaneo per la richiesta di delega che puoi condividere con l'amministratore dell'account. L'amministratore può accedere alla console di AWS gestione o utilizzare il link di accesso per esaminare le richieste di delega temporanea per approvare la richiesta e condividere l'accesso temporaneo con il richiedente.

> **Note**
>
> La concessione dell'accesso al fornitore di prodotti richiede due azioni: accettare la richiesta di delega (`AcceptDelegationRequest`) e rilasciare il token di scambio (). `SendDelegationToken` La AWS Management Console esegue entrambi i passaggi automaticamente quando si approva una richiesta. Se si utilizza l'API AWS CLI or, è necessario eseguire entrambi i passaggi separatamente.

Capacità di simulazione delle autorizzazioni -beta

Quando ricevi una richiesta di delega temporanea, puoi approvarla tu stesso o inoltrarla all'amministratore del tuo account per l'approvazione. Puoi delegare le autorizzazioni a un fornitore di prodotti solo se disponi delle autorizzazioni per i servizi e le azioni inclusi nella richiesta di delega temporanea. Se non hai accesso ai servizi e alle azioni richiesti, il fornitore del prodotto non riceverà tali autorizzazioni anche se sono incluse nella richiesta.

Ad esempio, una richiesta di delega temporanea richiede la possibilità di creare un bucket Amazon S3, avviare e interrompere le istanze in Amazon EC2 e assumere un ruolo IAM. L'identità che approva la richiesta può avviare e interrompere le istanze in Amazon EC2 e assumere un ruolo IAM, ma non dispone dell'autorizzazione per creare un bucket Amazon S3. Quando questa identità approva la richiesta, il fornitore del prodotto non è in grado di creare un bucket Amazon S3 anche se queste autorizzazioni sono state incluse nella richiesta di delega temporanea.

Poiché puoi delegare solo le autorizzazioni che già possiedi, è fondamentale valutare se disponi delle autorizzazioni richieste prima dell'approvazione. La funzionalità beta di simulazione delle autorizzazioni aiuta in questa valutazione confrontando le autorizzazioni con quelle incluse nella richiesta. La valutazione indica se è possibile approvare la richiesta con la propria identità attuale o se è necessario inoltrarla a un amministratore. Se l'analisi non è in grado di confermare che disponi di autorizzazioni sufficienti, inoltra la richiesta a un amministratore per la revisione. Questa valutazione si basa su un'analisi simulata delle autorizzazioni e potrebbe differire dall' AWS ambiente live, pertanto rivedi attentamente le autorizzazioni richieste prima di procedere.

Fasi successive

Dopo aver avviato una richiesta di delega temporanea, puoi gestire e monitorare la richiesta per tutto il suo ciclo di vita. Le seguenti procedure consentono di tracciare, approvare e controllare l'accesso temporaneo:

- Esamina le richieste di delega temporanea: monitora lo stato delle tue richieste di accesso e visualizza informazioni dettagliate sulle richieste di approvazione o rifiuto delle richieste di delega temporanea.
- Revoca l'accesso alle deleghe temporanee: termina immediatamente le sessioni di delega temporanea attive prima che scadano naturalmente.

Esamina le richieste di delega temporanea

Dopo aver avviato una richiesta di delega temporanea, puoi monitorare, approvare e rifiutare le richieste nella console IAM. La pagina Richieste di delega temporanea offre una visualizzazione centralizzata di tutte le richieste, incluse quelle in attesa di approvazione, completate o rifiutate. In qualità di amministratore, puoi esaminare queste richieste per concedere ai fornitori di prodotti l'accesso alle AWS risorse o rifiutarle in base alle politiche di sicurezza, ai requisiti aziendali o agli standard di conformità dell'organizzazione. Questa visibilità ti aiuta a tenere traccia del ciclo di vita dell'accesso dei fornitori di prodotti e a mantenere la supervisione delle autorizzazioni temporanee.

> **Note**
>
> È necessario disporre dell'aim: AcceptDelegationRequest autorizzazione ad approvare le richieste di delega temporanea.

Per approvare una richiesta di delega temporanea

1. Accedi alla console di AWS gestione e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, scegli Richieste di delega temporanee.
3. La pagina principale mostra un elenco delle richieste di delega temporanea con le seguenti informazioni:
   - ID richiesta: identificatore univoco della richiesta
   - Stato: stato attuale (in sospeso, approvato, rifiutato, condiviso, scaduto)
   - Richiedente: fornitore del prodotto associato alla richiesta
   - Avviato da: responsabile IAM dell'account che ha avviato la richiesta per il fornitore del prodotto
   - Richiesta creata: quando è stata inviata la richiesta

- La richiesta scade: quando la richiesta è scaduta o scadrà

4. (Facoltativo) Utilizza le opzioni di filtro per visualizzare le richieste in base allo stato:

   - Tutte le richieste: visualizza tutte le tue richieste indipendentemente dallo stato
   - In sospeso: visualizza le richieste in attesa dell'approvazione dell'amministratore
   - Approvato: visualizza le richieste approvate
   - Condiviso: visualizza le richieste per le quali è stato condiviso l'accesso
   - Rifiutate: visualizza le richieste rifiutate con i motivi del rifiuto

5. Per visualizzare informazioni dettagliate su una richiesta specifica o per esaminare una richiesta di approvazione in sospeso, scegli l'ID della richiesta.

6. Rivedi le informazioni dettagliate sulla richiesta:

   - Informazioni sul fornitore del prodotto
   - Motivo e giustificazione della richiesta
   - Durata richiesta
   - AWS Autorizzazioni richieste

7. Se sei un amministratore che esamina una richiesta in sospeso, scegli una delle seguenti opzioni:

   - Per approvare la richiesta, scegli Approva. Nella finestra di dialogo di approvazione, puoi visualizzare i risultati della simulazione delle autorizzazioni. Per ulteriori informazioni, consulta la funzionalità beta di simulazione delle autorizzazioni. Dopo aver confermato la durata dell'accesso e la tua AWS identità, scegli Approva per concedere l'accesso. Se il fornitore del prodotto ha richiesto l'accesso immediato, riceve automaticamente le autorizzazioni temporanee e inizia la durata dell'accesso. Altrimenti, informa la persona che ha avviato la richiesta di concedere l'accesso al fornitore del prodotto.
   - Per rifiutare la richiesta, scegli Rifiuta.
     - Nella finestra di dialogo di rifiuto, fornisci un motivo chiaro del rifiuto per aiutare il richiedente a capire perché la sua richiesta è stata rifiutata.
     - Scegli Rifiuta per negare l'accesso.

8. L'elenco delle richieste si aggiorna automaticamente per mostrare le informazioni sullo stato più aggiornate. Puoi anche aggiornare manualmente la pagina per verificare la presenza di aggiornamenti sullo stato.

Revoca l'accesso temporaneo alla delega

Sebbene le sessioni di accesso dei fornitori di prodotti siano progettate per scadere automaticamente dopo la durata approvata, in determinate situazioni potrebbe essere necessario interrompere immediatamente l'accesso. La revoca dell'accesso attivo del fornitore di prodotti fornisce un meccanismo di controllo di emergenza in caso di problemi di sicurezza, quando il lavoro del fornitore di prodotti viene completato anticipatamente o quando i requisiti aziendali cambiano. Sia i promotori delle richieste che gli amministratori possono revocare l'accesso per mantenere la sicurezza e il controllo operativo.

Per revocare l'accesso temporaneo alla delega

1. Accedi alla console di AWS gestione e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Richieste di delega temporanee.
3. Individua l'ID della richiesta per la sessione di accesso che desideri revocare.
4. Scegli Azioni, quindi scegli Revoca l'accesso.
5. Nella finestra di dialogo, scegli Revoca l'accesso per confermare che desideri interrompere immediatamente la sessione di accesso.

Dopo la revoca dell'accesso, il fornitore del prodotto non sarà più in grado di accedere alle tue risorse. AWS La revoca viene registrata a scopo di controllo. AWS CloudTrail

> **Important**
>
> La revoca dell'accesso interrompe immediatamente la sessione di accesso del fornitore del prodotto. Qualsiasi lavoro o processo in corso che utilizza l'accesso verrà interrotto. Assicurati che la revoca non interrompa le operazioni critiche.

> **Note**
>
> Non è possibile revocare l'accesso per le richieste approvate utilizzando un utente root. AWS consiglia di evitare di utilizzare un utente root per approvare le richieste di delega. Utilizzate invece un ruolo IAM con le autorizzazioni appropriate.

Gestione delle autorizzazioni per le richieste di delega

Gli amministratori possono concedere ai principali IAM le autorizzazioni per gestire le richieste di delega dai fornitori di prodotti. Ciò è utile quando si desidera delegare l'autorità di approvazione a utenti o team specifici dell'organizzazione o quando è necessario controllare chi può eseguire azioni specifiche sulle richieste di delega.

Le seguenti autorizzazioni IAM sono disponibili per la gestione delle richieste di delega:

Autorizzazione	Description
sono: Associate DelegationRequest	Associa una richiesta di delega non assegnata al tuo account AWS
Sono: GetDelega tionRequest	Visualizza i dettagli di una richiesta di delega
sono: UpdateDel egationRequest	Inoltra una richiesta di delega a un amministratore per l'approvazione
sono: AcceptDel egationRequest	Approvare una richiesta di delega
Sono: SendDeleg ationToken	Rilascia il token di scambio al fornitore del prodotto dopo l'approvazione
sono: RejectDel egationRequest	Rifiuta una richiesta di delega
sono: ListDeleg ationRequests	Elenca le richieste di delega per il tuo account

Note

Per impostazione predefinita, ai responsabili IAM che avviano una richiesta di delega vengono automaticamente concesse le autorizzazioni per gestire quella richiesta specifica. Possono associarla al proprio account, visualizzare i dettagli della richiesta, rifiutare una richiesta, inoltrarla a un amministratore per l'approvazione, rilasciare il token di scambio

al fornitore del prodotto dopo l'approvazione dell'amministratore ed elencare le richieste di delega di cui sono proprietari.

Notifications

La delega temporanea IAM si integra con le notifiche AWS utente per aiutarti a rimanere informato sulle modifiche allo stato delle richieste di delega. Le notifiche sono particolarmente utili per gli amministratori che devono esaminare e approvare le richieste di delega.

Con AWS User Notifications, puoi configurare gli avvisi da inviare tramite più canali, tra cui e-mail, Amazon Simple Notification Service (SNS), AWS Chatbot per Slack o Microsoft Teams e Console Mobile Application. AWS Ciò garantisce che le persone giuste vengano avvisate al momento giusto, consentendo una risposta più rapida alle approvazioni in sospeso o la consapevolezza delle modifiche di accesso. Puoi anche personalizzare gli eventi che attivano le notifiche in base alle esigenze e ai requisiti di sicurezza della tua organizzazione.

Eventi di notifica disponibili

Puoi iscriverti per ricevere notifiche per i seguenti eventi di delega temporanea IAM:

- Richiesta di delega temporanea IAM creata
- Richiesta di delega temporanea IAM assegnata
- Richiesta di delega temporanea IAM in attesa di approvazione
- Richiesta di delega temporanea IAM rifiutata
- Richiesta di delega temporanea IAM accettata
- Richiesta di delega temporanea IAM completata
- Richiesta di delega temporanea IAM scaduta

Configurazione delle notifiche

Per configurare le notifiche per gli eventi di delega temporanea IAM:

1. Apri la console AWS User Notifications
2. Crea o aggiorna una configurazione di notifica
3. Seleziona AWS IAM come servizio
4. Scegli per quali eventi di richiesta di delega desideri ricevere notifiche

5. Configura i tuoi canali di distribuzione (email, AWS Chatbot, ecc.)

Per istruzioni dettagliate sulla configurazione delle notifiche AWS utente, inclusa la configurazione dei canali di consegna e la gestione delle regole di notifica, consulta la documentazione sulle notifiche AWS utente.

CloudTrail

Tutte le azioni eseguite dai fornitori di prodotti utilizzando l'accesso delegato temporaneo vengono registrate automaticamente. AWS CloudTrail Ciò fornisce visibilità e verificabilità complete delle attività dei fornitori di prodotti nell'account dell'utente. AWS Puoi identificare quali azioni sono state intraprese dai fornitori di prodotti, quando sono state eseguite e quale account del fornitore di prodotti le ha eseguite.

Per aiutarvi a distinguere tra le azioni intraprese dai vostri responsabili IAM e quelle intraprese dai fornitori di prodotti con accesso delegato, CloudTrail gli eventi includono un nuovo campo chiamato `invokedByDelegate` sotto l'`userIdentity`elemento. Questo campo contiene l'ID dell' AWS account del fornitore del prodotto, semplificando il filtraggio e la verifica di tutte le azioni delegate.

CloudTrail Struttura dell'evento

L'esempio seguente mostra un CloudTrail evento per un'azione eseguita da un fornitore di prodotti utilizzando l'accesso delegato temporaneo:

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:Role-Session-Name",
        "arn": "arn:aws:sts::111122223333:assumed-role/Role-Name/Role-Session-Name",
        "accountId": "111122223333",
        "accessKeyId": "[REDACTED:AWS_ACCESS_KEY]",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "attributes": {
```

```
                "creationDate": "2024-09-09T17:50:16Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedByDelegate": {
            "accountId": "444455556666"
        }
    },
    "eventTime": "2024-09-09T17:51:44Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "GetUserPolicy",
    "awsRegion": "us-east-1",
    "requestParameters": {
        "userName": "ExampleIAMUserName",
        "policyName": "ExamplePolicyName"
    },
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Il `invokedByDelegate` campo contiene l'ID dell' AWS account del fornitore del prodotto che ha eseguito l'azione utilizzando l'accesso delegato. In questo esempio, l'account 444455556666 (il fornitore del prodotto) ha eseguito un'azione nell'account 111122223333 (l'account del cliente).

Delegazione temporanea IAM per AWS Partner

Panoramica di

La delega temporanea IAM consente AWS ai clienti di and/or integrare senza problemi i prodotti dei AWS partner nel loro AWS ambiente attraverso flussi di lavoro interattivi e guidati. I clienti possono concedere ai AWS partner un accesso limitato e temporaneo per configurare i AWS servizi richiesti, riducendo le difficoltà legate all'onboarding e accelerando il time-to-value.

Per integrare la delega temporanea nel tuo prodotto, scopri come comprendere l'integrazione, rivedi le autorizzazioni di comprensione e segui le linee guida per la valutazione delle politiche. ??? Per esempi di policy IAM, consulta i modelli di policy. Quando sei pronto per l'implementazione, vedi come creare la tua integrazione.

La delega temporanea IAM consente ai partner di:

- Semplifica l'onboarding dei clienti con il provisioning automatico delle risorse

- Riduci la complessità dell'integrazione eliminando i passaggi di configurazione manuali
- Crea fiducia attraverso autorizzazioni trasparenti e approvate dai clienti
- Abilita le operazioni continue con modelli di accesso a lungo termine utilizzando i limiti delle autorizzazioni

Come funziona

1. Il partner crea una richiesta di delega: i partner creano una richiesta specificando di quali autorizzazioni hanno bisogno e per quanto tempo
2. Recensioni dei clienti in AWS Console: il cliente vede esattamente quali autorizzazioni richiede il partner e perché
3. Il cliente approva: il cliente approva la richiesta e rilascia un token di scambio. Il token viene inviato al partner su questo argomento SNS specificato.
4. Il partner riceve credenziali temporanee: i partner scambiano il token con credenziali temporanee AWS
5. Il partner configura le risorse: i partner utilizzano le credenziali per configurare le risorse richieste nell'account del cliente

Qualifica del partner

Per qualificarsi per l'integrazione temporanea della delega, un partner deve soddisfare i seguenti requisiti:

- Partecipazione ISV Accelerate: è necessario essere iscritti al programma ISV Accelerate (ISVA).
- AWS Inserzione nel Marketplace: il prodotto deve essere pubblicato nel AWS Marketplace con il badge «Deployed on». AWS

Procedura di onboarding

Completa i passaggi seguenti per integrare la delega temporanea nel tuo prodotto:

1. Fase 1: Rivedi i requisiti

   Consulta questa documentazione per comprendere i requisiti di qualificazione e completa il questionario per i partner riportato di seguito.

2. Fase 2: Invia la tua richiesta di onboarding

   Invia un'email a aws-iam-partner-onboarding@amazon.com o contatta il tuo AWS rappresentante. Includi il questionario per i partner compilato con tutti i campi obbligatori della tabella seguente.

3. Fase 3: AWS convalida e revisione

   AWS sarà:

   - Conferma di soddisfare i criteri di qualificazione
   - Rivedi i modelli di policy e i limiti delle autorizzazioni
   - Fornisci un feedback sugli artefatti inviati

4. Fase 4: Perfeziona le tue politiche

   Rispondi al AWS feedback e invia modelli di policy aggiornati o limiti di autorizzazione, se necessario.

5. Fase 5: Completa la registrazione

   Dopo l'approvazione, AWS dovrà:

   - Abilita l'accesso API per gli account specificati
   - Condividi gli ARN per il modello di policy e il limite delle autorizzazioni (se applicabile)

   Riceverai una conferma al termine dell'onboarding. Potrai quindi accedere alle API di delega temporanee e ai tuoi account registrati CreateDelegationRequest e iniziare GetDelegatedAccessToken a integrare i flussi di lavoro delle richieste di delega nel tuo prodotto.

Questionario per i partner

La tabella seguente elenca le informazioni richieste per l'onboarding dei partner:

Informazioni	Description	Richiesto
ID account Partner Central	ID dell' AWS account registrato su AWS Partner Central.	Sì
PartnerId	ID partner fornito da AWS Partner Central.	No

Informazioni	Description	Richiesto
AWS ID prodotto Marketplace	ID del prodotto fornito da AWS Partner Central.	Sì
AWS ID dell'account	L'elenco degli ID AWS account che desideri utilizzare per chiamare le API di delega temporanee. Questo dovrebbe includere sia gli account di produzione che quelli non utilizzati. production/test	Sì
Nome del partner	Questo nome viene visualizzato dai clienti nella Console di AWS gestione quando esaminano la richiesta di delega temporanea.	Sì
Email di contatto	Uno o più indirizzi email per contattarti in merito alla tua integrazione.	Sì
Dominio richiedente	Il tuo dominio (ad esempio, www.example.com)	Sì
Integration description (Descrizione dell'integrazione)	Breve descrizione del caso d'uso che desideri risolvere utilizzando questa funzionalità. Puoi includere link di riferimento alla tua documentazione o ad altro materiale pubblico.	Sì
Diagramma architetturale	Diagramma dell'architettura che illustra i casi d'uso dell'integrazione.	No
Modello di policy	È necessario registrare almeno un modello di policy per questa funzionalità. Il modello di policy definisce le autorizzazioni temporanee da richiedere negli account dei clienti AWS . Per ulteriori informazioni, consulta la sezione Modello di policy.	Sì
Nome del modello di policy	Nome del modello di policy che desideri registrare.	Sì

Informazioni	Description	Richiesto
Limite delle autorizzazioni	Se desideri creare ruoli IAM negli account dei clienti utilizzando autorizzazioni temporanee, devi registrare un limite di autorizzazione con IAM. I limiti delle autorizza zioni verranno associati ai ruoli IAM creati per limitare le autorizzazioni massime sul ruolo. Puoi utilizzare le policy AWS gestite selezionate come limite di autorizzazione o registrare un nuovo limite di autorizzazione personali zzato (JSON). Per ulteriori informazioni, consulta la sezione Limiti delle autorizzazioni.	No
Nome del limite di autorizzazione	Il nome del limite di autorizzazione. Il formato è: arn:aws:i am: :partner:policy/permission_boundary/<partner_ domain>/<policy_name>_ <date>Il nome della policy deve includere la data di creazione come suffisso. Il nome non può essere aggiornato dopo la creazione del limite di autorizzazione. Se stai utilizzando una policy AWS gestita esistente, fornisci invece l'ARN della policy gestita.	No
Descrizione dei limiti di autorizza zione	Descrizione del limite delle autorizzazioni. Questa descrizione non può essere aggiornata dopo la creazione del limite di autorizzazione.	No

Comprendere la tua integrazione

Dopo aver completato il processo di onboarding, puoi creare la tua integrazione con la delega temporanea IAM. Un'integrazione completa comporta in genere tre categorie principali di lavoro:

1. Esperienza utente e progettazione del flusso di lavoro

Crea un'esperienza front-end nell'applicazione partner che guidi i clienti attraverso il flusso di lavoro di delega temporanea. La candidatura per i partner deve:

- Presentare un chiaro flusso di onboarding o configurazione in cui i clienti possano concedere l'accesso temporaneo. Etichetta chiaramente questa azione, ad esempio «Implementa con delega temporanea IAM».

- Reindirizza i clienti alla console di AWS gestione per esaminare e approvare la richiesta di delega utilizzando il link alla console restituito dall'API CreateDelegationRequest
- Fornisci messaggi appropriati su quali autorizzazioni vengono richieste e perché. I clienti possono visualizzare questo messaggio nella pagina dei dettagli della richiesta di delega.
- Gestisci il ritorno del cliente alla tua domanda dopo aver completato l'approvazione. AWS

Procedure consigliate per le richieste di delega temporanea

Quando implementi la delega temporanea per la tua applicazione partner, segui le best practice definite di seguito per aiutare i clienti a verificare l'autenticità e l'accuratezza delle richieste.

1. Includi il User-Identifiable contesto nei messaggi di richiesta

È necessario includere il contesto specifico dell'utente nel messaggio di richiesta di delega. Queste informazioni aiutano gli utenti a identificare le richieste dal flusso di lavoro esistente dei partner e a distinguere le richieste legittime. Informazioni suggerite nel messaggio di richiesta:

- Identificativo dell'account o nome utente del cliente nel servizio
- Nome dell'area di lavoro, ID dell'abbonamento o identificatore dell'organizzazione
- Risorsa specifica a cui si accede (nome del cluster, nome del progetto, ambiente)
- Identificatore univoco della transazione o della richiesta generato per questo tentativo di delega

Esempio di messaggio di richiesta:

```
Request from Partner A workspace "production-analytics"
Account: user@example.com
Workspace ID: 1234ABCD
Cluster: ml-training-cluster-01
Request ID: 1111-2222-3333-4444
```

2. Facoltativo: includere l'ID AWS dell'account quando si avvia la delega

Se l'ID dell'account cliente è disponibile, includilo nella richiesta di delega. Questa fase di verifica crea un legame aggiuntivo tra l'intento del cliente e il token di delega.

3. Messaggi di richiesta di progettazione per la verifica della sicurezza

Struttura i messaggi di richiesta per consentire ai clienti di verificare con sicurezza la legittimità prima di concedere l'accesso. Requisiti relativi all'esperienza utente:

- Visualizza il messaggio di richiesta in modo ben visibile nell'interfaccia dell'applicazione prima di reindirizzarlo a AWS
- Utilizza un linguaggio chiaro e descrittivo che si colleghi direttamente all'azione corrente del cliente
- Evita i messaggi generici che potrebbero essere applicati a qualsiasi richiesta di delega
- Includi dettagli sufficienti per consentire ai clienti di verificare che la richiesta corrisponda al flusso di lavoro previsto
- Visualizza l'ID AWS dell'account che riceverà l'accesso delegato
- Fornisci una spiegazione chiara delle autorizzazioni richieste dalla tua applicazione

Includi un avviso di conferma nella tua domanda prima di reindirizzare i clienti alla AWS schermata di consenso:

```
## Redirecting to IAM Temporary Delegation
You are about to grant [Your Service Name] temporary access to your AWS account.

Before clicking "Allow" on the AWS consent screen:
• Verify the request details match your current action
• Confirm the AWS account ID matches your intended account
• Make sure you initiated this request from [Your Service Name]
```

4. Consigli vincolanti per le sessioni

Mentre AWS IAM gestisce il flusso di autorizzazione principale, implementa queste pratiche nella tua applicazione per rafforzare l'integrità della sessione:

- Genera identificatori di richiesta univoci e monouso per ogni tentativo di delega
- Associa le richieste di delega alla sessione applicativa attiva del cliente
- Implementa la scadenza appropriata delle richieste in base al processo di delega
- Verifica che i callback di delega corrispondano al contesto della richiesta di origine
- Registra tutte le iniziazioni e i completamenti delle richieste di delega per il monitoraggio della sicurezza

Seguendo queste best practice, contribuisci a creare un'esperienza di delega temporanea più sicura.

2. Integrazione con le API

Usa le API di delega temporanea di IAM per inviare e gestire le richieste di delega. Dopo aver registrato AWS i tuoi account, puoi accedere alle seguenti API:

- IAM CreateDelegationRequest: crea una richiesta di delega per l'account di un cliente AWS . Questa API restituisce un link alla console a cui reindirizzare i clienti per esaminare e approvare la richiesta.
- AWS STS GetDelegatedAccessToken— Recupera AWS le credenziali temporanee dopo che un cliente ha approvato la richiesta di delega. Utilizza queste credenziali per eseguire azioni nell'account del cliente.

L'integrazione dovrebbe gestire l'intero ciclo di vita delle richieste di delega, inclusa la creazione di richieste, il monitoraggio del loro stato e il recupero delle credenziali temporanee una volta approvate.

3. Configurazione e orchestrazione delle risorse

Dopo aver ottenuto le credenziali temporanee, orchestrate i flussi di lavoro necessari per configurare le risorse nell'account del cliente. AWS Ciò potrebbe includere:

- Chiamare direttamente le API AWS del servizio per creare e configurare le risorse
- Implementazione dell'infrastruttura tramite modelli AWS CloudFormation
- La creazione di ruoli IAM per l'accesso continuo (richiede l'utilizzo dei limiti di autorizzazione). Come misura di difesa approfondita, ti consigliamo di utilizzare nomi di ruolo dinamici, ad esempio un nome con un suffisso generato casualmente. Questo approccio rende il nome del ruolo in ogni account cliente unico e imprevedibile.

La logica di orchestrazione dovrebbe essere idempotente e gestire i guasti con garbo, poiché i clienti potrebbero dover riprovare o modificare le approvazioni delle deleghe.

Comprendere le autorizzazioni

Come parte del processo di integrazione delle funzionalità, dovrai registrare le policy con IAM che definiscano le autorizzazioni che desideri richiedere negli account dei clienti. AWS Il processo di registrazione offre un'esperienza più coerente per i clienti e aiuta a evitare le insidie più comuni nella redazione delle policy.

Durante la registrazione, AWS valuta le tue politiche rispetto a una serie di convalide. Queste convalide hanno lo scopo di standardizzare la formattazione e la struttura delle politiche e fornire protezioni di base contro gli anti-pattern noti. Le convalide riducono inoltre il rischio di escalation dei privilegi, accessi involontari tra account e ampio accesso a risorse di alto valore negli account dei clienti.

Tipi di autorizzazione

AWS prenderà in considerazione due categorie di autorizzazioni: temporanee e a lungo termine.

Autorizzazioni temporanee

Le autorizzazioni temporanee limitano le autorizzazioni assegnate a qualsiasi sessione temporanea di accesso delegato. Le autorizzazioni temporanee sono descritte nei modelli di policy applicati alla sessione delegata. I modelli supportano i parametri forniti quando si crea una richiesta di delega. Questi valori dei parametri vengono quindi associati alla sessione. Le autorizzazioni temporanee funzionano allo stesso modo delle politiche di sessione disponibili AWS STS oggi: le politiche limitano la capacità dell'utente sottostante, ma non concedono alcun accesso aggiuntivo. Per ulteriori informazioni, consulta la AWS STS documentazione sulle politiche di sessione.

Long-Term Autorizzazioni

Long-term le autorizzazioni limitano le autorizzazioni di tutti i ruoli creati o gestiti tramite accesso temporaneo. Long-term le autorizzazioni sono implementate come limiti delle autorizzazioni IAM. È possibile inviare uno o più limiti di autorizzazione a AWS come parte dell'onboarding. Dopo l'approvazione, AWS condivideremo con te un ARN della politica a cui potrai fare riferimento nelle tue politiche.

Queste politiche di delimitazione hanno due caratteristiche degne di nota. Innanzitutto, sono immutabili. Se desideri aggiornare le autorizzazioni, puoi registrare un nuovo limite di autorizzazioni. Puoi quindi collegare il nuovo limite di autorizzazioni ai ruoli dei tuoi clienti inviando una nuova richiesta di delega. In secondo luogo, le politiche non sono basate su modelli. Poiché la stessa politica dei confini è condivisa a livello globale, non può essere modificata in base al cliente.

> **Important**
>
> I limiti delle autorizzazioni hanno un limite massimo di 6.144 caratteri.

> **Note**
>
> Se desideri aggiornare un limite di autorizzazione o un modello di policy, contatta IAM all'indirizzo aws-iam-partner-onboarding@amazon.com. Dopo aver registrato il nuovo limite di autorizzazione, puoi inviare una richiesta di delega ai clienti per aggiornare il ruolo IAM e allegare il limite di autorizzazione appena registrato. Per maggiori dettagli, consulta la sezione Esempi.

Esempio di caso d'uso: carico di lavoro di elaborazione dati

Prendiamo in considerazione un fornitore di prodotti che esegue un carico di lavoro di elaborazione dati negli account dei clienti. Il provider deve configurare l'infrastruttura durante l'onboarding iniziale, ma richiede anche un accesso continuo per gestire il carico di lavoro.

Autorizzazioni temporanee (per la configurazione iniziale):

- Crea istanze Amazon EC2, VPC e gruppi di sicurezza
- Crea un bucket Amazon S3 per i dati elaborati
- Crea un ruolo IAM per le operazioni continue
- Allega un limite di autorizzazione al ruolo IAM

Long-term autorizzazioni (ruolo IAM con limite di autorizzazione per le operazioni in corso):

- Avvia e arresta le istanze Amazon EC2 per eseguire processi di elaborazione
- Leggi i dati di input dal bucket Amazon S3
- Scrivi i risultati elaborati nel bucket Amazon S3

Le autorizzazioni temporanee vengono utilizzate una volta durante l'onboarding per configurare l'infrastruttura. Il ruolo IAM creato durante questo processo ha un limite di autorizzazioni che limita le autorizzazioni massime alle sole operazioni necessarie per la gestione continua del carico di lavoro. Ciò garantisce che, anche se le politiche del ruolo vengono modificate, non possono superare le autorizzazioni definite nel limite.

Linee guida di valutazione delle politiche

AWS valuterà le politiche presentate rispetto a una serie di linee guida. Le stesse linee guida di valutazione si applicano sia ai modelli di policy che ai limiti delle autorizzazioni, con lievi differenze rilevate ove appropriato.

Ai fini della valutazione, i servizi sono suddivisi in gruppi distinti. La distinzione più importante riguarda i servizi sensibili alla sicurezza, che gestiscono l'accesso, le credenziali e le chiavi. Le politiche che concedono l'accesso a questi servizi devono essere strettamente incentrate sul lavoro svolto. Security-sensitive i servizi includono: AWS Identity and Access Management (IAM), AWS Key Management Service (KMS), AWS Resource Access Manager (RAM), AWS IAM Identity Center, AWS Organizations e Secrets Manager. AWS

Una distinzione secondaria sono i servizi che possono accedere ai dati attraverso i confini dell'account. Le politiche per questi servizi devono includere protezioni per impedire l'accesso involontario tra account.

Convalide comuni

Tutte le dichiarazioni politiche devono seguire queste linee guida:

- Tutte le dichiarazioni devono includere i campi Effetto, Azione (o NotAction), Risorsa e Condizione in questo ordine
- Tutte le azioni all'interno di una singola dichiarazione devono essere elencate in ordine alfabetico
- Tutti gli ARN inclusi nella policy devono seguire la sintassi definita nella documentazione pubblica per i servizi pertinenti
- NotAction i campi possono essere utilizzati solo nelle dichiarazioni Deny
- Le azioni nelle istruzioni Allow devono includere un codice di servizio. I caratteri jolly generici («*») non sono consentiti

Security-sensitive restrizioni di servizio

Le seguenti restrizioni si applicano ai servizi sensibili alla sicurezza sopra menzionati:

- Le azioni nelle istruzioni Allow devono essere più specifiche di [service] :*
- Le azioni nelle istruzioni Allow per i modelli di policy di accesso temporaneo non devono contenere caratteri jolly

- Le azioni sensibili, come iam: PassRole o iam:CreateServiceLinkedRole, richiedono un ambito aggiuntivo, ad esempio risorse specifiche o controlli condizionali. Queste azioni includono:
  - Passaggio di ruoli IAM
  - Azioni di modifica del ruolo IAM
  - Azioni di modifica delle politiche IAM
  - AWS Operazioni di scrittura o crittografia KMS
  - AWS Operazioni di scrittura o condivisione della RAM
  - AWS Operazioni di Secrets Manager per recuperare o modificare i segreti o modificare le politiche delle risorse
- Altre azioni potrebbero utilizzare una risorsa con caratteri jolly, come iam: o iam: ListUsers GetPolicy
- Le azioni che gestiscono le credenziali, come iam:CreateAccessKey, sono bloccate

IAM-specific restrizioni

Per IAM:

- Per i ruoli e le policy IAM sono consentite solo operazioni di scrittura limitate. Non è possibile richiedere autorizzazioni su altre risorse IAM come utenti, gruppi e certificati.
- L'allegato delle policy o le azioni di gestione delle policy in linea sono limitate ai ruoli con un limite di autorizzazioni. I limiti delle autorizzazioni devono essere forniti dal partner o inclusi in un elenco di politiche gestite consentite. AWS AWS le politiche gestite potrebbero essere consentite se non concedono autorizzazioni amministrative o altamente privilegiate. Ad esempio, le politiche AWS gestite per specifiche funzioni lavorative o la SecurityAudit politica potrebbero essere accettabili. AWS esaminerà ogni politica AWS gestita caso per caso durante il processo di onboarding.
- La gestione delle policy è consentita solo per le policy con un percorso specifico per il partner: arn:aws:iam: :@ {}: .com/ [feature] * AccountId policy/partner_domain
- I tag possono essere applicati solo durante la creazione delle risorse e solo per ruoli e policy
- iam: PassRole i controlli devono corrispondere a un nome o a un prefisso di percorso specifico

AWS STS-restrizioni specifiche

Per AWS STS:

- sts: AssumeRole deve avere come ambito un ARN di ruolo specifico, un prefisso ARN del ruolo o limitato a un insieme di account o unità organizzative ID/organizational

Restrizioni dello IAM Identity Center

Per AWS IAM Identity Center, le seguenti azioni sono bloccate:

- Tutte le azioni relative alla gestione delle autorizzazioni (ad esempio, sso:) AttachCustomerManagedPolicyReferenceToPermissionSet
- Modifiche a utenti, gruppi e appartenenze per Identity Store AWS
- Gestione dei tag

AWS Restrizioni delle organizzazioni

Per AWS le organizzazioni, saranno consentite solo le azioni di lettura.

Convalide aggiuntive specifiche del servizio

- Le azioni che acquisiscono segreti o credenziali, come glue: GetConnection o redshift:, devono avere condizioni che corrispondano agli ARN completiGetClusterCredentials, ai prefissi ARN o ai tag
- Per Amazon Redshift: redshift: GetClusterCredentials è consentito solo su un nome di database specifico e redshift: è consentito solo su un nome di gruppo di lavoro specifico GetClusterCredentialsWithIAM

> **Note**
>
> Quando gestisci le risorse IAM nell'account, ti consigliamo di utilizzare un percorso che indichi il tuo nome, ad esempio arn:aws:iam: :111122223333:. role/partner com/rolename. Ciò contribuirà a differenziare le risorse associate all'integrazione e a facilitare la scoperta, l'audit e l'analisi per i clienti.

Cross-account requisiti di accesso

Le dichiarazioni che potenzialmente consentono l'accesso tra più account devono includere almeno uno dei seguenti:

- Una condizione che specifica l'account o l'organizzazione della risorsa (ad esempio, aws: ResourceOrgId corrispondente a uno o più valori previsti)
- Un campo Risorsa che include un account specifico (ad esempio, arn:aws:sqs: *:111122223333: *)
- Un campo Resource che include un account non wildcard e un nome completo della risorsa (ad esempio, arn:aws:s3: ::full-bucket-name)

> **Note**
>
> Cross-account l'accesso è una funzionalità sensibile che richiede una chiara giustificazione aziendale. AWS esaminerà attentamente la necessità dell'accesso tra più account durante il processo di onboarding.

Modelli di policy

I modelli di policy sono un nuovo costrutto IAM progettato per definire le autorizzazioni temporanee richieste dai partner negli account dei clienti. Come le normali politiche IAM, definiscono le autorizzazioni utilizzando istruzioni con elementi Effect, Action, Resource e Condition. La differenza fondamentale è che i modelli di policy includono parametri (come @ {bucketName}) che vengono sostituiti con valori effettivi quando si crea una richiesta di delega.

Come funzionano i modelli di policy

Come parte del processo di onboarding, registri i tuoi modelli di polizza con AWS. AWS Assegna a ciascun modello un ARN univoco a cui si fa riferimento durante la creazione delle richieste di delega.

Quando si crea una richiesta di delega, si specifica:

- Il modello di policy ARN
- Valori dei parametri da sostituire nel modello

AWS combina il modello con i valori dei parametri per generare una policy IAM standard. I clienti esaminano questa policy renderizzata finale quando approvano la richiesta di delega, per vedere esattamente quali autorizzazioni verranno concesse.

> **Note**
>
> La policy renderizzata finale ha un limite massimo di 2048 caratteri.

Ecco un semplice esempio che mostra come funziona la sostituzione dei template.

Modello di politica:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::@{bucketName}/*"
        }
    ]
}
```

Parametri forniti nella richiesta di delega:

```
{
    "Name": "bucketName",
    "Values": ["customer-data-bucket"],
    "Type": "String"
}
```

Politica resa finale (cosa vedono i clienti):

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
```

```
            "Resource": "arn:aws:s3:::customer-data-bucket/*"
        }
    ]
}
```

Sintassi del modello

I modelli di policy utilizzano due funzionalità chiave per fornire flessibilità: sostituzione dei parametri e dichiarazioni condizionali. La sostituzione dei parametri consente di definire dei segnaposto nel modello che vengono sostituiti con valori effettivi durante la creazione di una richiesta di delega. Le dichiarazioni condizionali consentono di includere o escludere intere dichiarazioni politiche in base ai valori dei parametri.

Sostituzione e tipi di parametri

Usa la sintassi @ {parameterName} per definire i parametri nel tuo modello di policy. Quando si crea una richiesta di delega, è necessario specificare il tipo per ogni parametro.

Stringa

Un singolo valore che viene sostituito direttamente nel modello.

Modello:

```
"Resource": "arn:aws:s3:::@{bucketName}/*"
```

Parametri:

```
{
    "Name": "bucketName",
    "Values": ["my-bucket"],
    "Type": "String"
}
```

Risultato renderizzato:

```
"Resource": "arn:aws:s3:::my-bucket/*"
```

StringList

Valori multipli che generano più voci di risorse. Quando un StringList parametro viene utilizzato in un ARN di risorsa, si espande per creare voci di risorse separate per ciascun valore.

Modello:

```
"Resource": "arn:aws:s3:::@{bucketNames}/*"
```

Parametri:

```
{
    "Name": "bucketNames",
    "Values": ["bucket-1", "bucket-2"],
    "Type": "StringList"
}
```

Risultato renderizzato:

```
"Resource": [
    "arn:aws:s3:::bucket-1/*",
    "arn:aws:s3:::bucket-2/*"
]
```

Cross-Product Comportamento

Quando vengono utilizzati più parametri nella stessa risorsa ARN, StringList i parametri creano un prodotto incrociato di tutte le combinazioni.

Modello:

```
"Resource": "arn:aws:s3:::@{bucketNames}/@{prefix}/*"
```

Parametri:

```
[
    {
        "Name": "bucketNames",
        "Values": ["bucket-1", "bucket-2"],
        "Type": "StringList"
    },
    {
        "Name": "prefix",
        "Values": ["data"],
        "Type": "String"
```

```
    }
]
```

Risultato renderizzato:

```
"Resource": [
    "arn:aws:s3:::bucket-1/data/*",
    "arn:aws:s3:::bucket-2/data/*"
]
```

Dichiarazioni condizionali

Utilizzate la direttiva @Enabled per includere o escludere condizionalmente intere istruzioni in base ai valori dei parametri.

Sintassi:

- @Enabled: «parameterName»: include l'istruzione quando il valore del parametro è «True»
- @Enabled: «! parameterName»: include l'istruzione quando il valore del parametro NON è «True» (negazione)

Modello:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["s3:GetObject"],
            "Resource": "*"
        },
        {
            "@Enabled": "ENABLE_S3_WRITE",
            "Effect": "Allow",
            "Action": ["s3:PutObject"],
            "Resource": "arn:aws:s3:::@{bucketName}/*"
        }
    ]
}
```

Parametri (quando ENABLE_S3_WRITE è «True»):

```
[
    {
        "Name": "bucketName",
        "Values": ["my-bucket"],
        "Type": "String"
    },
    {
        "Name": "ENABLE_S3_WRITE",
        "Values": ["True"],
        "Type": "String"
    }
]
```

Risultato renderizzato:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["s3:GetObject"],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": ["s3:PutObject"],
            "Resource": "arn:aws:s3:::my-bucket/*"
        }
    ]
}
```

Parametri (quando ENABLE_S3_WRITE è «False»):

```
[
    {
        "Name": "bucketName",
        "Values": ["my-bucket"],
        "Type": "String"
    },
    {
        "Name": "ENABLE_S3_WRITE",
        "Values": ["False"],
```

```
        "Type": "String"
    }
]
```

Risultato renderizzato:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["s3:GetObject"],
            "Resource": "*"
        }
    ]
}
```

Quando ENABLE_S3_WRITE è impostato su «True», l'istruzione condizionale è inclusa. Se è impostata su «False», l'istruzione viene esclusa dalla policy renderizzata.

Esempi aggiuntivi

Gli esempi seguenti illustrano i modelli comuni per l'utilizzo dei modelli di policy nelle deleghe temporanee. Si concentrano sulla creazione di ruoli IAM con limiti di autorizzazione per l'accesso a lungo termine e mostrano diverse strategie per definire l'ambito delle autorizzazioni per risorse specifiche. Questi esempi illustrano come bilanciare flessibilità e sicurezza utilizzando tecniche come i prefissi ARN, l'etichettatura delle risorse e gli aggiornamenti dei limiti di autorizzazione.

Esempio 1: concessione dell'accesso a risorse specifiche Long-Term

Il seguente limite di autorizzazione viene inviato come "SQSAccessorBoundary" per «partner.com»:

```
{
    "Effect": "Allow",
    "Action": [
        "sqs:DeleteMessage",
        "sqs:ReceiveMessage",
        "sqs:SendMessage"
    ],
    "Resource": "arn:aws:sqs:*:*:*",
    "Condition": {
        "StringEquals": {
```

```
            "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
    }
}
```

> **Note**
>
> Ciò include una condizione relativa allo stesso account per evitare di concedere l'accesso alle code in altri account con criteri relativi alle risorse aperte. Non è possibile includere un riferimento diretto all'ID dell'account del cliente perché il limite è condiviso tra tutti i clienti e non può essere utilizzato come modello.

Poiché questa è la prima versione di questa politica, il suo ARN è arn:aws:iam: :partner: -. policy/permissions boundary/partner com/SQSAccessorBoundary_2025_01_15

Il seguente modello di policy viene inviato per le autorizzazioni di accesso temporanee:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sqs:ListQueues"
            ],
            "Resource": "arn:aws:sqs:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:PutRolePermissionsBoundary",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::@{AccountId}:role/partner.com/SQSAccessor",
            "Condition": {
                "StringEquals": {
                    "iam:PermissionsBoundary": "arn:aws:iam::partner:policy/
permissions-boundary/partner.com/SQSAccessorBoundary_2025_01_15"
                }
            }
```

```
        }
    ]
}
```

Esempio 2: utilizzo dei prefissi ARN

Il limite delle autorizzazioni può specificare un prefisso ARN della risorsa per limitare l'accesso:

```
"Resource": "arn:aws:sqs:*:@{AccountId}:PartnerPrefix*"
```

Ciò limita l'accesso solo alle risorse con quel prefisso, riducendo l'ambito delle risorse accessibili.

Esempio 3: utilizzo dei tag per il controllo dell'accesso alle risorse

È possibile etichettare le risorse durante l'accesso temporaneo delegato e fare affidamento su tali tag per il controllo degli accessi a lungo termine.

Limite di autorizzazione che consente l'accesso alle risorse contrassegnate:

```
{
    "Effect": "Allow",
    "Action": [
        "sqs:DeleteMessage",
        "sqs:ReceiveMessage",
        "sqs:SendMessage"
    ],
    "Resource": "arn:aws:sqs:*:*:*",
    "Condition": {
        "Null": {
            "aws:ResourceTag/ManagedByPartnerDotCom": "false"
        },
        "StringEquals": {
            "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
    }
}
```

Modello di policy per etichettare le nuove code al momento della creazione:

```
{
    "Effect": "Allow",
    "Action": [
```

```
        "sqs:CreateQueue",
        "sqs:TagQueue"
    ],
    "Resource": "arn:aws:sqs:*:*:*",
    "Condition": {
        "Null": {
            "aws:RequestTag/ManagedByPartnerDotCom": "false"
        }
    }
}
```

Modello di policy per etichettare le code preesistenti e creare il ruolo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sqs:TagQueue"
            ],
            "Resource": "arn:aws:sqs:*:@{AccountId}:@{QueueName}",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "ManagedByPartnerDotCom"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:PutRolePermissionsBoundary",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::@{AccountId}:role/partner.com/SQSAccessor",
            "Condition": {
                "StringEquals": {
                    "iam:PermissionsBoundary": "arn:aws:iam::partner:policy/
permissions-boundary/partner.com/SQSAccessorBoundary_2025_01_15"
                }
            }
        }
```

```
    ]
}
```

Questo approccio consente ai clienti di confermare esplicitamente a quali risorse specifiche è possibile accedere a lungo termine.

Esempio 4: aggiornamento del limite delle autorizzazioni

Per aggiornare un limite di autorizzazione, registra una nuova versione con un nuovo suffisso di data e richiedi l'autorizzazione per sostituirlo.

Limite di autorizzazione aggiornato con autorizzazione aggiuntiva:

```
{
    "Effect": "Allow",
    "Action": [
        "sqs:DeleteMessage",
        "sqs:PurgeQueue",
        "sqs:ReceiveMessage",
        "sqs:SendMessage"
    ],
    "Resource": "arn:aws:sqs:*:*:*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
    }
}
```

Come seconda versione, questa policy ha l'ARN: arn:aws:iam: :partner: -. policy/permissions boundary/partner com/SQSAccessorBoundary_2025_01_20

Modello di policy per aggiornare il limite delle autorizzazioni sul ruolo esistente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:PutRolePermissionsBoundary"
```

```
            ],
            "Resource": "arn:aws:iam::@{AccountId}:role/partner.com/SQSAccessor",
            "Condition": {
                "StringEquals": {
                    "iam:PermissionsBoundary": "arn:aws:iam::partner:policy/
permissions-boundary/partner.com/SQSAccessorBoundary_2025_01_20"
                }
            }
        }
    ]
}
```

I clienti devono approvare questa richiesta di delega per aggiornare il limite di autorizzazione sul ruolo esistente.

Sviluppa la tua integrazione

Comprendere il ciclo di vita delle richieste

Prima di creare l'integrazione, è importante capire in che modo le richieste di delega procedono dalla creazione al completamento.

Stati della richiesta

Una richiesta di delega procede attraverso i seguenti stati:

Stato	Description
Non assegnato	Richiesta creata ma non ancora associata a un account cliente e a un responsabile IAM. La richiesta potrebbe essere stata creata senza specificare un account di destinazione o con un ID account di destinazione ma non ancora richiesta dal proprietario dell'account.
Assegnata	Richiesta associata a un account cliente e in attesa di revisione
In attesa di approvazione	Il cliente ha inoltrato la richiesta a un amministratore per l'approvazione
Accettato	Richiesta approvata dal cliente ma token di scambio non ancora rilasciato

Stato	Description
Finalizzato	Token di scambio rilasciato al fornitore del prodotto. Il periodo di delega (validità del token di scambio) inizia quando la richiesta raggiunge lo stato Finalizzato
Rifiutato	Richiesta rifiutata dal cliente
Scaduto	Richiesta scaduta per inattività o timeout

Transizioni di stato

Flusso normale (percorso di approvazione)

- Non assegnato → Assegnato: il cliente associa la richiesta al proprio account
- Assegnata → Accettata O Assegnata → In attesa di approvazione: il cliente approva la richiesta direttamente OPPURE la inoltra all'amministratore per la revisione
- In attesa di approvazione → Accettata: l'amministratore approva la richiesta
- Accettato → Finalizzato: il cliente rilascia il token di scambio

Percorso di rifiuto

- Assegnato → Rifiutato: il cliente rifiuta la richiesta
- In attesa di approvazione → Rifiutata: l'amministratore rifiuta la richiesta
- Accettato → Rifiutato: il cliente revoca l'approvazione prima di rilasciare il token

Percorso di scadenza

Le richieste scadono automaticamente se non viene intrapresa alcuna azione entro il periodo di tempo specificato:

- Non assegnato → Scaduto (1 giorno)
- Assegnato → Scaduto (7 giorni)
- In attesa di approvazione → Scaduto (7 giorni)
- Accettato → Scaduto (7 giorni)
- Rifiutato → Scaduto (7 giorni)

- Finalizzato → Scaduto (7 giorni)

Stati terminali

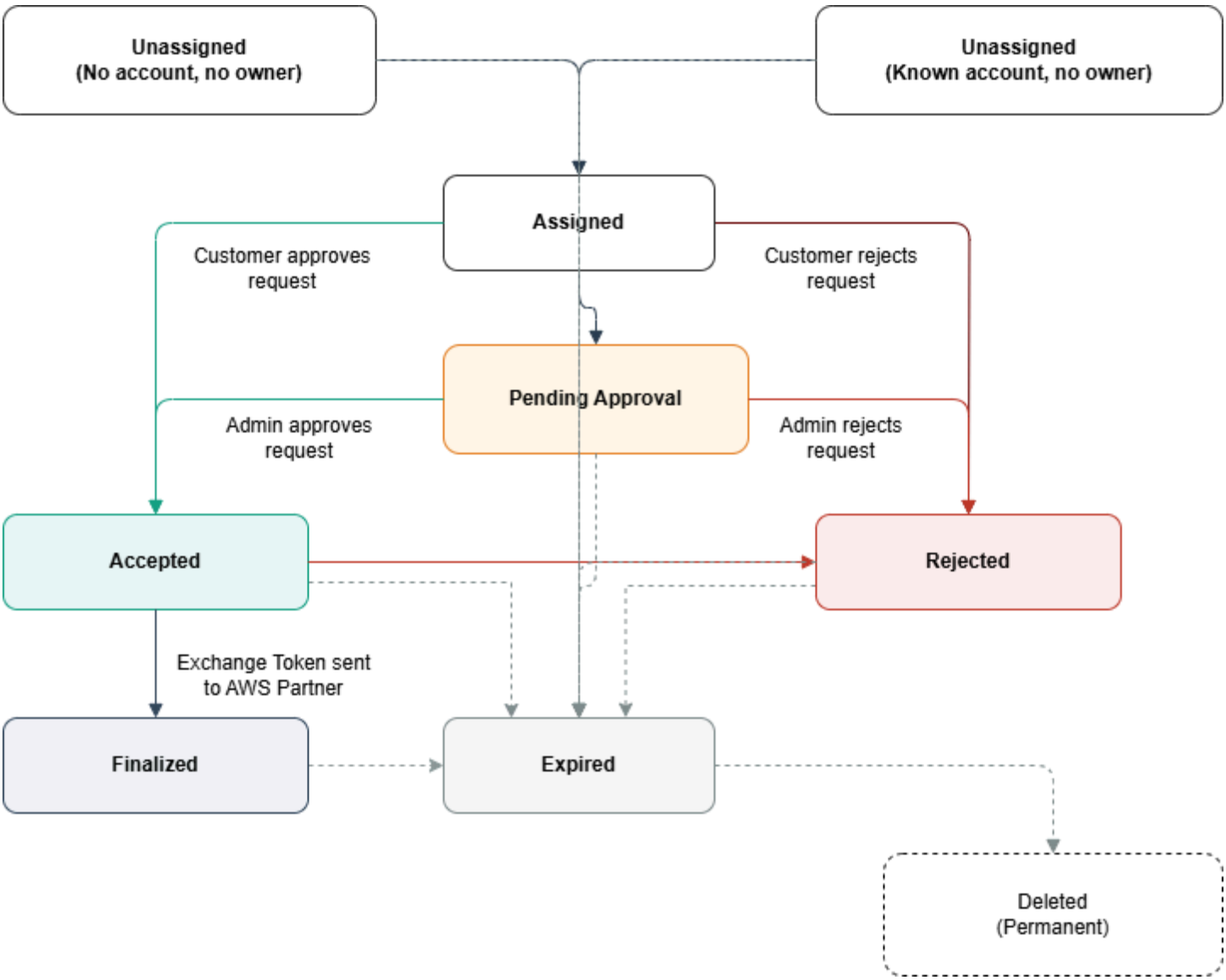
I seguenti stati sono terminali (senza ulteriori transizioni):

- Finalizzato: Exchange Token inviato
- Rifiutata: la richiesta è stata respinta
- Scaduta: la richiesta è scaduta o il periodo di delega è terminato

Le richieste scadute vengono infine eliminate dal sistema dopo il periodo di conservazione.

Gestione degli stati delle richieste di delega nell'applicazione

In qualità di partner, devi tenere traccia degli stati delle richieste di delega nel tuo sistema e mostrarli ai tuoi clienti. Quando ricevete notifiche SNS sulle modifiche di stato, archiviate questi aggiornamenti nel backend e rifletteteli nell'interfaccia utente rivolta ai clienti. Presta particolare attenzione allo stato In attesa di approvazione: quando un cliente inoltra una richiesta a un amministratore per la revisione, ti invia una notifica in sospeso. AWS Le richieste possono rimanere in questo stato per un massimo di 7 giorni in attesa dell'intervento dell'amministratore. Durante questo periodo, mostra ai clienti che la loro richiesta è in attesa dell'approvazione dell'amministratore nella tua applicazione. Prendi in considerazione la possibilità di fornire un collegamento diretto alla AWS Console in cui i clienti possano verificare lo stato della richiesta o contattare il proprio amministratore. Per una buona esperienza di integrazione è importante gestire correttamente la macchina a stati nel backend e fornire ai clienti le informazioni sullo stato corrette in ogni fase.



Configurazione delle notifiche

IAM utilizza Amazon Simple Notification Service (SNS) per comunicarti le modifiche allo stato delle richieste di delega. Quando crei una richiesta di delega, devi fornire un argomento SNS ARN dal tuo account registrato. AWS IAM pubblicherà messaggi su questo argomento per eventi importanti, ad esempio quando i clienti approvano o rifiutano le richieste e quando il token di scambio è pronto.

> **Note**
>
> Gli argomenti SNS non possono essere inclusi nelle regioni opt-in. AWS L'argomento SNS deve essere in una AWS regione abilitata per impostazione predefinita. Per un elenco delle

regioni opt-in, consulta Gestione delle AWS regioni nella guida alla gestione degli AWS account.

Configurazione degli argomenti SNS

Per ricevere notifiche di richiesta di delega, devi configurare l'argomento SNS per concedere le autorizzazioni IAM per pubblicare messaggi su di esso. Aggiungi la seguente dichiarazione politica alla tua policy sull'argomento SNS:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIAMServiceToPublish",
            "Effect": "Allow",
            "Principal": {
                "Service": "iam.amazonaws.com"
            },
            "Action": "SNS:Publish",
            "Resource": "arn:aws:sns:us-east-1:111122223333:TOPIC-NAME"
        }
    ]
}
```

Important

L'argomento SNS deve essere presente in uno dei tuoi account registrati AWS . IAM non accetterà argomenti SNS da altri account. Se la policy sull'argomento non è configurata correttamente, non riceverai notifiche di modifica dello stato o il token di scambio.

Tipi di notifica

IAM invia due tipi di notifiche:

StateChange Notifiche

Inviata quando una richiesta di delega passa a un nuovo stato (Assegnata, In attesa di approvazione, Accettata, Finalizzata, Rifiutata, Scaduta).

ExchangeToken Notifiche

Inviato quando un cliente rilascia il token di delega (stato Finalizzato). Questa notifica include il token di scambio necessario per ottenere le credenziali.

Stati di notifica

Riceverai notifiche per i seguenti stati di richiesta di delega:

Stato	Tipo di notifica	Description
ASSEGNATO	StateChange	La richiesta è stata associata a un account cliente
IN ATTESA DI APPROVAZIONE	StateChange	Il cliente ha inoltrato la richiesta a un amministratore per l'approvazione
ACCETTATO	StateChange	Il cliente ha approvato la richiesta ma non ha ancora rilasciato il token
FINALIZZATO	StateChange	Il cliente ha rilasciato il token di scambio
FINALIZZATO	ExchangeToken	Questa notifica contiene l'Exchange Token
REJECTED	StateChange	Il cliente ha rifiutato la richiesta
SCADUTO	StateChange	Richiesta scaduta prima del completamento

Formato del messaggio di notifica

IAM pubblica notifiche SNS standard. Le informazioni sulla richiesta di delega sono contenute nel campo Message come stringa JSON.

Campi comuni (tutte le notifiche)

Campo	Tipo	Description
Tipo	Stringa	O "StateChange" o "ExchangeToken»
RequestId	Stringa	L'ID della richiesta di delega IAM

Campo	Tipo	Description
RequestorWorkflowId	Stringa	L'ID del flusso di lavoro fornito durante la creazione della richiesta
Stato	Stringa	Stato attuale della richiesta
OwnerAccountId	Stringa	ID dell' AWS account del cliente
UpdatedAt	Stringa	Timestamp in cui lo stato è cambiato (formato ISO 8601)

Campi aggiuntivi (solo notifiche) ExchangeToken

Campo	Tipo	Description
ExchangeToken	Stringa	Il token da scambiare per le credenziali tramite API AWS STS GetDelegatedAccessToken
ExpiresAt	Stringa	Alla scadenza dell'accesso delegato (formato ISO 8601)

Esempi di notifiche

StateChange Notifica

```
{
  "Type": "Notification",
  "MessageId": "61ee8ad4-6eec-56b5-8f3d-eba57556aa13",
  "TopicArn": "arn:aws:sns:us-east-1:123456789012:partner-notifications",
  "Message": "{\"RequestorWorkflowId\":\"workflow-12345\",\"Type\":\"StateChange\",
\"RequestId\":\"dr-abc123\",\"State\":\"ACCEPTED\",\"OwnerAccountId\":\"111122223333\",
\"UpdatedAt\":\"2025-01-15T10:30:00.123Z\"}",
  "Timestamp": "2025-01-15T10:30:00.456Z",
  "SignatureVersion": "1",
  "Signature": "...",
  "SigningCertURL": "...",
  "UnsubscribeURL": "..."
}
```

ExchangeToken Notifica

```
{
  "Type": "Notification",
  "MessageId": "e44e5435-c72c-5333-aba3-354406782f5b",
  "TopicArn": "arn:aws:sns:us-east-1:123456789012:partner-notifications",
  "Message": "{\"RequestId\":\"dr-abc123\",\"RequestorWorkflowId\":\"workflow-12345\",
\"State\":\"FINALIZED\",\"OwnerAccountId\":\"111122223333\",\"ExchangeToken\":
\"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...\",\"ExpiresAt\":\"2025-01-15T18:30:00.123Z\",
\"UpdatedAt\":\"2025-01-15T10:30:00.456Z\",\"Type\":\"ExchangeToken\"}",
  "Timestamp": "2025-01-15T10:30:00.789Z",
  "SignatureVersion": "1",
  "Signature": "...",
  "SigningCertURL": "...",
  "UnsubscribeURL": "..."
}
```


Token di scambio

Un token di scambio o un token di permuta viene emesso da IAM quando un cliente accetta e finalizza una richiesta di delega. Il fornitore del prodotto utilizza questo token di scambio o permuta per chiamare l' AWS STS GetDelegatedAccessToken API e ottenere AWS credenziali temporanee con le autorizzazioni approvate dai clienti. Il token di scambio di per sé non garantisce l'accesso alle tue AWS risorse; deve essere scambiato con credenziali effettive tramite AWS STS.

Il token di scambio può essere riscattato solo dall'account del fornitore del prodotto che ha creato la richiesta di delega. L'account richiedente è incorporato nel token, garantendo che solo il fornitore di prodotti autorizzato possa ottenere le credenziali per accedere all'account del cliente.

Durata dell'accesso

Il periodo di delega inizia quando il cliente rilascia il token di scambio, non quando il fornitore del prodotto lo riscatta. Dopo che il cliente ha rilasciato il token:

- Il fornitore del prodotto riceve il token tramite una notifica SNS
- Possono scambiarlo immediatamente con credenziali
- Le credenziali scadono a: ora di rilascio + durata approvata
- Il fornitore del prodotto può scambiare il token più volte prima della scadenza per ottenere nuove credenziali, se necessario

Rimborsi multipli

I fornitori di prodotti possono scambiare il token più volte durante il periodo di validità per ottenere nuove credenziali. Tuttavia, tutte le credenziali ottenute dallo stesso token di scambio scadono contemporaneamente, in base alla data di rilascio del token.

Esempio: se approvi una richiesta di delega di 2 ore e rilasci il token alle 10:00:

Ora di rilascio del token	Tempo di scambio dei token	Scadenza delle credenziali	Tempo utilizzabile
10:00	10:00 del mattino	12:00	2 ore
10:00 del mattino	10:20 del mattino	12:00	1 ora e 40 minuti
10:00 del mattino	11:40 del mattino	12:00	20 minuti
10:00 del mattino	12:10	Fallito (token scaduto)	0 minuti

Come mostrato nella tabella, lo scambio del token più avanti nel periodo di validità comporta una riduzione del tempo utilizzabile per il fornitore del prodotto.

Gestore dell'accesso all'account

Cos'è l'account access manager?

Account Access Manager ti consente di assegnare Account AWS l'accesso a utenti e gruppi nell'istanza dell'organizzazione IAM Identity Center. Assegni l'accesso utilizzando i ruoli IAM nel tuo Account AWS. Puoi utilizzare l'account access manager insieme ai set di autorizzazioni IAM Identity Center o separatamente. Sia l'account access manager che i set di autorizzazioni IAM Identity Center utilizzano gli utenti e i gruppi che hai sincronizzato dalla tua fonte di identità o creati direttamente in IAM Identity Center.

Perché usare l'account access manager?

Account Access Manager ti consente di assegnare i ruoli IAM esistenti agli utenti e ai gruppi che hai sincronizzato in IAM Identity Center dalla tua fonte di identità aziendale. L'utilizzo dei ruoli IAM esistenti offre una maggiore flessibilità per personalizzare l'accesso degli utenti, tra cui:

- Configura i tag dei ruoli IAM e usa i tag di sessione per il controllo degli accessi basato sugli attributi (ABAC).
- Configura la politica di fiducia dei ruoli IAM per limitare l'assunzione di ruoli o utilizzare le IdP-asserted attestazioni come condizioni della politica di fiducia.
- Configura i percorsi dei ruoli IAM per raggruppare i ruoli in modo logico e semplifica l'amministrazione. Per maggiori informazioni, consulta Configurare i percorsi dei ruoli IAM per raggruppare i ruoli in modo logico e semplificare l'amministrazione.

La tabella seguente confronta le tre opzioni di accesso all'account:

	Ruoli IAM + gestore degli accessi agli account	Set di autorizzazioni IAM Identity Center	Ruoli IAM + federazione IAM (assunzione diretta del ruolo)
Ideale per	Ruoli IAM personalizzati che variano a seconda degli account	Accesso umano di base uniforme, ad esempio in sola lettura e con provision	Adatto per carichi di lavoro

	Ruoli IAM + gestore degli accessi agli account	Set di autorizzazioni IAM Identity Center	Ruoli IAM + federazione IAM (assunzione diretta del ruolo)
		ing amministrativo come ruoli IAM immutabili	
Scope (Ambito)	Account multipli (AWS organizzazione)	Account multipli (AWS organizzazione)	Per account
Assegnazione dei ruoli IAM	Richiede l'infrastruttura come codice (IaC) o il provisioning manuale tramite la console IAM AWS	Fornito automaticamente	Richiede l'infrastruttura come codice (IaC) o il provisioning manuale tramite console AWS IAM e IdP esterne
Role-to-identity la mappatura risiede in	Gestore dell'accesso all'account	Centro identità IAM	IdP esterno
Esperienza dell'utente finale	Gli utenti accedono agli AWS account tramite l'URL del portale di accesso all' AWS account dopo aver effettuato l'accesso AWS come utenti IAM Identity Center.	Gli utenti accedono agli AWS account tramite il portale di AWS accesso dopo aver effettuato l'accesso AWS come utenti dello IAM Identity Center.	Gli utenti accedono agli AWS account tramite applicazioni SAML specifiche dell'account (icone) nel loro portale IdP.

	Ruoli IAM + gestore degli accessi agli account	Set di autorizzazioni IAM Identity Center	Ruoli IAM + federazione IAM (assunzione diretta del ruolo)
AWS CLI esperienza utente	1. L'utente accede AWS e quindi accede a un account con un ruolo specifico in un browser web. 2. L'utente digita aws login sulla riga di comando per recuperare le credenziali della sessione del ruolo IAM dal browser. Per tutta la durata della sessione del ruolo IAM, l'utente può continuare a lavorare con AWS CLI senza dover ripetere l'autenticazione nel passaggio 1.	L'utente configura un profilo che include l'account e la coppia di ruoli desiderati digitando. aws configure sso L'utente digita aws sso login per avviare una nuova sessione utilizzando un profilo. Se non è presente alcuna sessione IAM Identity Center attiva, l'utente si autentica in un browser e ne autorizza l'accesso . Consulta la Guida per Centro identità AWS IAM l'utente per maggiori dettagli.	Integrazioni personalizzate tramite provider di AWS credenziali SDK

Funzionalità chiave

Portale di accesso all'account

Account Access Manager fornisce agli utenti un portale di accesso all'account in cui possono visualizzare e accedere ai Account AWS dati loro assegnati tramite Account Access Manager.

Accesso alla CLI

La tua forza lavoro può accedere Account AWS AWS CLI utilizzando il aws login comando dopo aver effettuato l'accesso AWS in un browser.

API di accesso all'account

L'Account Access Manager dispone di un proprio namespace API`account-access`, separato dall'API IAM. Questo namespace viene utilizzato negli SDK e. AWS CLI

Guida introduttiva all'account access manager

Di seguito viene descritto come iniziare a utilizzare Account Access Manager. Ciò presuppone che i ruoli IAM che desideri assegnare esistano già nei rispettivi ruoli. Account AWS Se questi ruoli non esistono, consulta Creare un ruolo utilizzando politiche di fiducia personalizzate per informazioni su come creare nuovi ruoli.

Per iniziare con Account Access Manager

1. Assicurati di avere un'istanza organizzativa di IAM Identity Center abilitata e di aver effettuato il provisioning degli utenti e dei gruppi della tua forza lavoro. Per ulteriori informazioni, consulta la sezione Guida introduttiva a IAM Identity Center nella Guida per Centro identità AWS IAM l'utente. Vedi the section called "Prerequisiti e considerazioni" più avanti in questo argomento per maggiori dettagli.

   > **Note**
   >
   > Se utilizzi IAM Identity Center replicato in più AWS regioni o con una chiave gestita dal cliente (CMK), devi aggiungere l'`kms:Decrypt`autorizzazione all'account access manager per utilizzare questa CMK quando ti connetti a IAM Identity Center. Per ulteriori informazioni, consulta la politica chiave di Baseline KMS nella Guida per l'utente. Centro identità AWS IAM

2. Assicurati di disporre delle autorizzazioni per abilitare l'account access manager e gestire le assegnazioni degli account al suo interno. Per ulteriori informazioni, consulta the section called "Gestione delle identità e degli accessi per l'account access manager".
3. Abilita l'account access manager tramite la console o l'API.
4. Aggiorna i firewall e i gateway per accedere al portale di accesso all'account.

5. Concedi all'account access manager l'accesso ai ruoli IAM aggiornando la loro politica di fiducia.
6. Inizia ad assegnare l'accesso all'account agli utenti e ai gruppi della tua forza lavoro da IAM Identity Center.
7. Fai sapere ai tuoi utenti come accedere Account AWS utilizzando le assegnazioni dei ruoli di Account Access Manager.

Prerequisiti e considerazioni

Si applicano i seguenti prerequisiti e considerazioni:

- Account Access Manager è disponibile nelle aree AWS commerciali abilitate per impostazione predefinita. Opt-in Le regioni commerciali GovCloud (USA), la Cina e altre regioni non commerciali non sono supportate.
- Al momento dell'abilitazione, l'account access manager si connette a una singola istanza organizzativa di IAM Identity Center. È necessario abilitare preventivamente un' AWS organizzazione e un'istanza organizzativa di IAM Identity Center. Le istanze di account di IAM Identity Center non sono supportate.
- Ti consigliamo di abilitare l'account access manager nella stessa AWS regione in cui hai abilitato l'istanza IAM Identity Center. Se non l'hai già fatto, consulta Considerazioni per la scelta di una AWS regione nella Guida per l' Centro identità AWS IAM utente.
- Sia l'istanza organizzativa di IAM Identity Center che l'account access manager devono essere abilitati nell'account di gestione dell'organizzazione.

Abilita l'account access manager

Dopo aver creato un'istanza organizzativa di IAM Identity Center, puoi abilitare l'account access manager tramite la relativa console o API. È necessario eseguire questa operazione nella stessa Account AWS area geografica dell'istanza di IAM Identity Center.

Prima di abilitare l'account access manager, esegui il seguente comando dall'account di gestione dell' AWS organizzazione per concedere l'accesso all'account access manager alla tua organizzazione:

```
aws organizations enable-aws-service-access --service-principal account-
access.amazonaws.com
```

> **Note**
>
> Sebbene la console dell'account access manager faccia parte della console IAM, la relativa API risiede in un namespace separato.

Utilizza uno dei seguenti metodi per abilitare l'account access manager:

Console

Per abilitare l'account access manager

1. Accedi all'account di gestione della tua organizzazione.
2. Nel pannello di navigazione della AWS Identity and Access Management console, scegli Account access manager.
3. Verifica che la regione della console sia impostata sulla regione principale della tua istanza IAM Identity Center.
4. Se non è presente alcuna istanza organizzativa di IAM Identity Center, ti verrà richiesto di abilitarne una ora prima di continuare. Altrimenti, scegli Abilita il gestore degli accessi agli account.

AWS CLI

Per prima cosa trova l'ARN dell'istanza di IAM Identity Center e verifica che il codice regionale corrisponda alla regione principale di IAM Identity Center. Quindi, esegui il seguente comando:

```
aws account-access create-application \
  --region <Region> \
  --identity-source '{
    "identityCenter": {
      "instanceArn": "<IAM_IDENTITY_CENTER_INSTANCE_ARN>"
    }
  }'
```

Aggiorna firewall e gateway

Il portale di accesso all'account fornisce agli utenti l'accesso Single Sign-on ai dati loro Account AWS assegnati tramite Account Access Manager.

Se si filtra l'accesso a AWS domini o endpoint URL specifici utilizzando una soluzione di filtraggio dei contenuti Web come i firewall di nuova generazione (NGFW) o i Secure Web Gateway (SWG), è necessario inserire nella lista consentita i domini e gli endpoint URL associati al portale di accesso all'account.

L'elenco seguente fornisce i domini dual-stack e gli endpoint URL da aggiungere agli elenchi consentiti della soluzione di filtraggio dei contenuti web. L' IPv4-only account access manager non ha endpoint.

Dual-stack elenco degli indirizzi consentiti

L'URL del portale di accesso all'account:

- `https://[Tenant-ID].account-access.[Region].app.aws`

Le API chiamate dal portale di accesso all'account:

- `[Tenant-ID].account-access.[Region].app.aws/api/*`
- `[Tenant-ID].account-access.[Region].app.aws/auth/*`

Tenant-ID è un identificatore casuale univoco associato all'account access manager. Per trovarne il valore, accedi alla scheda Impostazioni nella console di gestione degli accessi all'account e cerca l'URL dell'applicazione nella sezione Dettagli dell'istanza. Tenant-ID è il primo sottodominio nell'URL.

Gli endpoint del servizio di gestione degli accessi all'account:

- Control-plane: `account-access.[Region].api.aws`
- Runtime: `account-access-runtime.[Region].api.aws`
- Console: `console.aws.amazon.com/account-access`

Se utilizzi il AWS CLI, puoi trovare l'ID del tenant eseguendo il seguente comando CLI nell'account di gestione o di amministratore delegato della tua organizzazione.

```
aws account-access list-applications
```

La risposta contiene informazioni sull'account access manager, inclusi l'ARN dell'applicazione e l'ID del tenant.

```
{
    "applications": [
        {
            "applicationArn": "arn:aws:account-access:us-
west-2:123456789012:application/1234567890abcdef",
            "tenantId": "aa-gyxmap389",
            "createdAt": "2026-03-27T18:31:19+00:00",
            "updatedAt": "2026-03-27T18:31:19+00:00"
        }
    ]
}
```

Considerazioni per i domini consentiti e gli endpoint URL

Oltre ai requisiti della lista consentita per il portale di accesso all'account, gli altri servizi e applicazioni utilizzati potrebbero richiedere l'elenco dei domini consentiti.

- È necessario inserire nella lista consentita il portale di AWS accesso e le relative dipendenze. Per ulteriori informazioni, vedere Aggiornamento di firewall e gateway per consentire l'accesso al portale di accesso nella Guida per AWS l'utente. Centro identità AWS IAM
- Per accedere alla Account AWS Console di gestione AWS, e alla console di gestione degli accessi dell'account, devi inserire nell'elenco dei domini aggiuntivi consentiti. Per un elenco dei domini, consulta la sezione Risoluzione dei problemi nella Guida Console di gestione AWS introduttiva. Console di gestione AWS

Elimina il gestore degli accessi agli account

Quando l'account access manager viene eliminato, tutti i dati in esso contenuti vengono eliminati e non possono essere recuperati.

Utilizza la seguente procedura per eliminare l'account access manager.

Console

Per eliminare l'istanza dell'account access manager

1. Accedi all'account di gestione della tua organizzazione.
2. Nel pannello di navigazione della AWS Identity and Access Management console, scegli Account access manager.

3. Verifica che la regione della console sia impostata sulla regione principale dell'istanza dell'account access manager.
4. Seleziona la scheda Impostazioni.
5. Nella sezione Elimina il gestore dell'accesso all'account, scegli Elimina.
6. La finestra di dialogo richiede di confermare. Quando sei pronto, scegli Elimina.

AWS CLI

Esegui il comando seguente per eliminare l'account access manager:

```
aws account-access delete-application \
  --region "<Region>" \
  --application-arn "<account_access_manager_ARN>"
```

Warning

L'eliminazione dell'account access manager è irreversibile e comporta le seguenti conseguenze:

- Tutte le assegnazioni di ruolo vengono rimosse definitivamente. I tuoi utenti perdono immediatamente la possibilità di accedere Account AWS con i ruoli IAM assegnati tramite l'account access manager.
- L'account access manager viene rimosso dal portale di AWS accesso per tutti gli utenti.
- Le sessioni attive dei ruoli IAM assegnati tramite l'account access manager continuano fino alla scadenza, ma gli utenti non possono iniziare nuove sessioni.
- Se riattivi l'account access manager in un secondo momento, devi ricreare tutte le assegnazioni da zero. Le assegnazioni precedenti non sono recuperabili.

Prima dell'eliminazione, valuta la possibilità di disabilitare l'accesso degli utenti nei singoli account per limitare temporaneamente l'accesso.

Gestisci l'accesso a Account AWS

L'Account Access Manager è integrato con IAM Identity Center e AWS Organizations consente di gestire centralmente l'assegnazione dei ruoli IAM esistenti su più ruoli Account AWS senza configurare ogni account singolarmente. Con l'account access manager, assegni i ruoli IAM esistenti nei tuoi utenti e gruppi Account AWS di IAM Identity Center per controllarne l'accesso a determinati utenti. Account AWS

Con l'account access manager, i tuoi team possono creare i propri ruoli IAM personalizzati e tu Account AWS, in qualità di amministratore, puoi utilizzare l'account access manager per gestire centralmente le assegnazioni di utenti e gruppi IAM Identity Center a questi ruoli IAM.

> **Note**
>
> Nel contesto dell'account access manager, i termini utenti e gruppi si riferiscono esclusivamente agli utenti e ai gruppi della forza lavoro in IAM Identity Center.

Amministrazione delegata

Quando abiliti l'account access manager, lo AWS crea nell'account di gestione AWS Organizations perché l'account access manager gestisce le assegnazioni dei ruoli IAM per un'intera AWS organizzazione. Puoi scegliere di delegare l'amministrazione dell'account access manager a un account membro in. AWS Organizations

Se si sceglie di registrare un account membro come amministratore delegato per l'account access manager, gli utenti di questo account possono eseguire la maggior parte delle attività amministrative di Account Access Manager. La tabella seguente indica se un'attività amministrativa di Account Access Manager può essere eseguita dagli utenti di un account di gestione dell'organizzazione, di un account membro dell'amministrazione delegata o di un altro account membro. La gestione dei ruoli IAM è esclusa dall'elenco delle attività amministrative perché avviene all'esterno dell'account access manager.

Attività amministrative del gestore degli accessi agli account	Gestione dell'account	Account amministratore delegato	Account membro
Gestisci le assegnazioni dei ruoli IAM*	Sì	Sì	No
Abilita il gestore degli accessi agli account	Sì	No	No
Elimina il gestore degli accessi all'account	Sì	No	No
Abilita o disabilita l'accesso degli utenti nell'account di gestione	Sì	No	No
Abilita o disabilita l'accesso utente in un account membro	Sì	Sì	No
Registra o annulla la registrazione di un account membro come amministratore delegato	Sì	No	No

*Fai riferimento alle best practice per l'amministrazione delegata per quanto riguarda le assegnazioni di utenti e gruppi all'account di gestione.

Registra un account membro

Per configurare l'amministrazione delegata, devi prima registrare un account membro nella tua organizzazione come amministratore delegato. Gli utenti di quell'account membro che dispongono di autorizzazioni sufficienti avranno accesso amministrativo all'account access manager.

Account Access Manager supporta la registrazione di un solo account membro alla volta come amministratore delegato. È possibile registrare un account membro solo dopo aver effettuato l'accesso con le credenziali dell'account di gestione.

Utilizza la seguente procedura per registrare un account membro nella tua AWS organizzazione come amministratore delegato.

Per registrare un account membro

1. Accedi a Console di gestione AWS utilizzando le credenziali del tuo account di gestione. AWS Organizations Le credenziali dell'account di gestione sono necessarie per eseguire l'RegisterDelegatedAdministratorAPI.
2. Seleziona la regione in cui è stato abilitato l'account access manager, quindi apri la AWS Identity and Access Management console.
3. Scegli Account access manager, quindi scegli la scheda Impostazioni.
4. Nella sezione Amministratore delegato, scegli Registrati.
5. Nella pagina Registra amministratore delegato, seleziona l'amministratore delegato Account AWS che desideri registrare, quindi scegli Registra.

Annullamento della registrazione di un account membro

Puoi annullare la registrazione di un account membro solo se hai effettuato l'accesso con le credenziali dell'account di gestione.

Utilizza la procedura seguente per annullare la registrazione di un account membro AWS dell'organizzazione che in precedenza era stato designato come amministratore delegato.

> **Important**
>
> Quando si annulla la registrazione di un account, si rimuove la possibilità per tutti gli utenti amministratori di gestire l'account access manager da quell'account. Non possono più amministrare Account AWS l'accesso da questo account.
> Questa operazione non influisce sulle autorizzazioni o sulle assegnazioni configurate in Account Access Manager. Gli utenti finali continuano ad accedere ai propri Account AWS dal portale di accesso all'account.

Per annullare la registrazione di un account membro

1. Accedi a Console di gestione AWS utilizzando le credenziali del tuo account di gestione. AWS Organizations Le credenziali dell'account di gestione sono necessarie per eseguire l'DeregisterDelegatedAdministratorAPI.
2. Seleziona la regione in cui è abilitato l'account access manager, quindi apri la AWS Identity and Access Management console.
3. Scegli Account access manager, quindi scegli la scheda Impostazioni.
4. Nella sezione Amministratore delegato, scegli Annulla registrazione account.
5. Nella finestra di dialogo Annulla registrazione amministratore delegato, esamina le implicazioni sulla sicurezza e conferma di aver compreso.
6. Scegli Annulla registrazione.

Visualizza l'account amministratore delegato registrato

Utilizza la procedura seguente per scoprire quale account membro del tuo AWS Organizations è stato configurato come amministratore delegato per l'account access manager.

Per visualizzare il tuo account utente registrato

1. Accedi a Console di gestione AWS utilizzando le credenziali del tuo account di gestione. AWS Organizations
2. Seleziona la regione in cui è stato abilitato l'account access manager, quindi apri la AWS Identity and Access Management console.
3. Scegli Account access manager, quindi scegli la scheda Impostazioni.
4. Nella sezione Amministratore delegato, puoi visualizzare i dettagli dell'account amministratore delegato corrente.

Prepara i tuoi ruoli IAM

Prima di iniziare ad assegnare ruoli IAM agli utenti e ai gruppi di IAM Identity Center, devi:

1. Identifica i ruoli IAM pertinenti negli account di interesse e
2. Assicurati che la politica di fiducia dei ruoli IAM contenga la dichiarazione fornita di seguito.

Se il ruolo ha una politica di fiducia esistente, aggiungi questa dichiarazione al relativo array Statement. Se stai creando un nuovo ruolo, puoi utilizzare l'intero documento di policy. Questa dichiarazione politica consente all'account access manager di assumere il ruolo e impostare il contesto per conto degli utenti. Il `Condition` blocco fornisce una protezione in caso di confusione assicurando che l'account access manager possa assumere il ruolo solo per conto del tuo account e della tua applicazione di gestione degli accessi all'account. Per ulteriori informazioni, consulta Aggiornare una politica di fiducia dei ruoli.

Sostituiscila `<your-account-id>` con il tuo Account AWS ID, `<region>` con la regione principale del tuo IAM Identity Center e `<application-id>` con l'ID dell'applicazione dell'account access manager. Puoi trovare l'ARN dell'applicazione nella pagina Impostazioni della console dell'account access manager.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccountAccessManagerIAMRoleTrustPolicyStatement",
      "Effect": "Allow",
      "Principal": {
        "Service": "account-access.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetContext"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<your-account-id>",
          "aws:SourceArn": "arn:aws:account-access:<region>:<your-account-
id>:application/<application-id>"
        }
      }
    }
  ]
}
```

Opzionale: **`sts:TagSession`**. Aggiungilo `sts:TagSession` all'`Action`elenco se desideri che gli attributi di asserzione originari del tuo provider di identità (ad esempio, appartenenze a gruppi, reparto, centro di costo) si propaghino come tag principali nella sessione emessa da Account Access Manager. Senza questa azione, l'assunzione del ruolo ha comunque esito positivo; viene soppressa

solo la propagazione dei tag di sessione. Aggiungilo quando le policy IAM downstream dell'account di destinazione utilizzano i tag principal ()`aws:PrincipalTag/<key>`; altrimenti lascialo fuori. Mantieni lo stesso `Condition` blocco.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccountAccessManagerIAMRoleTrustPolicyStatement",
      "Effect": "Allow",
      "Principal": {
        "Service": "account-access.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetContext",
        "sts:TagSession"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<your-account-id>",
          "aws:SourceArn": "arn:aws:account-access:<region>:<your-account-
id>:application/<application-id>"
        }
      }
    }
  ]
}
```

> **Note**
>
> Ti consigliamo di aggiornare i flussi di lavoro per la creazione dei ruoli IAM per aggiungere automaticamente questa dichiarazione sulla politica di fiducia dove necessario.

Assegna e rimuovi l'accesso

Assegna l'accesso di un utente o di gruppo a un Account AWS

Puoi assegnare l'accesso di un utente o di gruppo a un Account AWS creando un'assegnazione tra l'utente o il gruppo, l'account e il ruolo IAM che l'utente o il gruppo può assumere nell'account.

Pertanto, una singola assegnazione è una tripletta composta dall'ID utente o gruppo, dall'ID dell'account e dall'ARN del ruolo IAM.

Se si desidera eseguire assegnazioni in blocco, ad esempio l'assegnazione dello stesso ruolo a più utenti e gruppi, è necessario creare ciascuna assegnazione singolarmente. Puoi scrivere questa procedura utilizzando l'API di accesso all'account e il. AWS SDK/CLI

Console

La console offre diversi modi per assegnare un nuovo accesso a seconda che si desideri identificare prima l'account, l'utente o il gruppo.

> **Note**
>
> La console di gestione degli accessi agli account fornisce una funzionalità di ricerca per utenti, gruppi e account. Per identificare il ruolo che desideri assegnare, devi prima trovarne il nome (o ARN). La console dell'account access manager non è in grado di rilevare i ruoli IAM in. Account AWS

Per tutte le procedure in questa sezione, accedi all'account di gestione (o all'account amministratore delegato), seleziona la regione in cui è abilitato l'account access manager e vai a Account access manager nella console IAM.

Accounts tab

Per assegnare l'accesso a utenti o gruppi dalla scheda Account

1. Scegliere la scheda Accounts (Account).
2. Se conosci già l'ID dell'account, scegli Assegna nuovo accesso. Nella pagina Assegna accesso all'account, cerca l'utente o il gruppo desiderato, inserisci l'ID dell'account e il nome del ruolo IAM, quindi scegli Assegna accesso.
3. Se devi prima trovare l'account, puoi cercare quello desiderato in Account AWS base al nome, all'ID o all'indirizzo email. In alternativa, puoi consultare la struttura organizzativa o l'elenco degli account per trovare quelli pertinenti Account AWS. Scegli il pulsante di opzione davanti all'account e poi Assegna nuovo accesso. Nella pagina Assegna l'accesso all'account, cerca l'utente o il gruppo desiderato, inserisci il nome del ruolo IAM e quindi scegli Assegna accesso.

4. La struttura organizzativa o l'elenco degli account consente di scegliere un account per visualizzare i dettagli dell'account e le assegnazioni di accesso all'account. Puoi scegliere Assegna nuovo accesso anche da lì.

Users tab

Per assegnare l'accesso agli utenti dalla scheda Utenti

1. Scegli la scheda Users (Utenti);
2. È possibile cercare gli utenti per nome utente e nome visualizzato o trovarli nell'elenco degli utenti visualizzato.
3. Scegli l'utente a cui vuoi assegnare un nuovo accesso.
4. La pagina successiva mostra gli attributi dell'utente, le assegnazioni degli account e le appartenenze ai gruppi.
5. Scegli Assegna nuovo accesso.
6. Nella pagina Assegna accesso all'account, inserisci l'ID dell'account e il nome del ruolo IAM, quindi scegli Assegna accesso.

Groups tab

Per assegnare l'accesso ai gruppi dalla scheda Gruppi

1. Nel riquadro di navigazione, scegli Account access manager.
2. Scegliere la scheda Groups (Gruppi).
3. Puoi cercare i gruppi in base al nome del gruppo o trovarli nell'elenco dei gruppi visualizzato.
4. Scegli il gruppo a cui vuoi assegnare un nuovo accesso.
5. La pagina successiva mostra gli attributi del gruppo, le assegnazioni degli account e i membri degli utenti.
6. Scegli Assegna nuovo accesso.
7. Nella pagina Assegna accesso all'account, inserisci l'ID dell'account e il nome del ruolo IAM, quindi scegli Assegna accesso.

AWS CLI

Il AWS CLI comando seguente può essere utilizzato per creare l'assegnazione di un account per un utente:

```
aws account-access create-entitlement \
  --region <Region> \
  --application-arn "<Account_access_manager_ARN>" \
  --entitlement '{
    "principalRole": {
      "principal": {
        "identityCenter": {
          "userId": "<Identity_Center_User_ID>"
        }
      },
      "roleArn": "arn:aws:iam::<ACCOUNT_ID>:role/<ROLE_NAME>"
    }
  }'
```

Il AWS CLI comando seguente può essere utilizzato per creare l'assegnazione di un account per un gruppo:

```
aws account-access create-entitlement \
  --region <Region> \
  --application-arn "<Account_access_manager_ARN>" \
  --entitlement '{
    "principalRole": {
      "principal": {
        "identityCenter": {
          "groupId": "<IDC_GROUP_ID>"
        }
      },
      "roleArn": "arn:aws:iam::<ACCOUNT_ID>:role/<ROLE_NAME>"
    }
  }'
```

Rimuovere l'accesso di un utente o di gruppo a un Account AWS

Puoi trovare l'assegnazione da rimuovere in diversi modi.

Console

Per tutte le procedure in questa sezione, accedi all'account di gestione (o all'account amministratore delegato), seleziona la regione in cui è abilitato l'account access manager e vai a Account access manager nella console IAM.

Accounts tab

Per rimuovere l'accesso di utenti o gruppi dalla scheda Account

1. Scegliere la scheda Accounts (Account).
2. Trova l'account pertinente e scegline il nome.
3. Nella sezione Assegnazioni di accesso all'account, scegli l'utente o il gruppo a cui desideri rimuovere l'accesso e scegli Annulla assegnazione.
4. Nella modalità, scegli Annulla assegnazione per confermare.

Users tab

Per rimuovere l'accesso degli utenti dalla scheda Utenti

1. Scegli la scheda Users (Utenti);
2. Scegli l'utente a cui desideri rimuovere l'accesso.
3. Nella sezione Assegnazioni di accesso all'account, scegli l'account per cui desideri rimuovere l'accesso e scegli Annulla assegnazione.
4. Nella modalità, scegli Annulla assegnazione per confermare.

AWS CLI

Per rimuovere l'accesso tramite la CLI, devi prima trovare l'assegnazione dell'account (diritto) e quindi utilizzarne l'ID nel comando seguente. Più avanti in questo argomento puoi vedere come cercare le assegnazioni di account per un account, utente, gruppo o ruolo specifico. La CLI si riferisce alle assegnazioni come diritti.

```
aws account-access delete-entitlement \
    --region <Region> \
    --application-arn "<Account_access_manager_ARN>" \
    --entitlement-id "<ENTITLEMENT_ID>"
```

Visualizzare o cercare le assegnazioni degli account

È possibile visualizzare le assegnazioni degli account in diversi modi.

Console

Per tutte le procedure in questa sezione, accedi all'account di gestione (o all'account amministratore delegato), seleziona la regione in cui è abilitato l'account access manager e vai a Account access manager nella console IAM.

Accounts tab

Visualizza l'accesso di utenti o gruppi dalla scheda Account

1. Scegliere la scheda Accounts (Account).
2. Per impostazione predefinita, viene visualizzata la vista Gerarchia. Se lo si desidera, è possibile passare alla visualizzazione Elenco. Puoi cercare gli account in base al nome, all'ID, all'email e all'ID dell'unità organizzativa. Quando trovi l'account desiderato, scegline il nome.
3. Nella pagina dell'account, puoi vedere le assegnazioni degli account nella sezione Assegnazioni di accesso all'account. Puoi filtrarli ulteriormente per nome utente, ID utente, nome del gruppo, ID del gruppo e ARN del ruolo IAM.

Users tab

Visualizza l'accesso degli utenti dalla scheda Utenti

1. Scegli la scheda Users (Utenti);
2. Cerca l'utente per nome o ID utente. Quando trovi l'utente desiderato, scegli il nome utente.
3. Nella pagina utente, puoi vedere le assegnazioni degli account dell'utente. Puoi filtrarli ulteriormente in base all'ID dell'account, al nome dell'account e all'ARN del ruolo IAM.

Groups tab

Visualizza l'accesso ai gruppi dalla scheda Gruppi

1. Scegliere la scheda Groups (Gruppi).
2. Cerca il gruppo per nome o ID del gruppo. Quando trovi il gruppo desiderato, scegli il nome del gruppo.

3. Nella pagina del gruppo, puoi vedere le assegnazioni degli account del gruppo. Puoi filtrarli ulteriormente in base all'ID dell'account, al nome dell'account e all'ARN del ruolo IAM.

AWS CLI

I comandi seguenti illustrano come recuperare le assegnazioni degli account utilizzando diversi filtri. La CLI si riferisce alle assegnazioni come diritti.

Recupera le prime 100 assegnazioni (diritti) per un account specifico

```
aws account-access list-entitlements \
    --region <Region> \
    --application-arn "<Account_access_manager_ARN>" \
    --filter '{
      "principalRole": {
        "account": "<ACCOUNT_ID>"
      }
    }' \
    --max-results 100
```

Recupera tutte le assegnazioni (diritti) per un utente specifico

```
aws account-access list-entitlements \
    --region <Region> \
    --application-arn "<Account_access_manager_ARN>" \
    --filter '{
      "principal": {
        "type": "USER",
        "id": "<USER_ID>"
      }
    }'
```

Recupera tutte le assegnazioni (diritti) per un gruppo specifico

```
aws account-access list-entitlements \
    --region <Region> \
    --application-arn "<Account_access_manager_ARN>" \
    --filter '{
      "principal": {
        "type": "GROUP",
```

```
        "id": "<GROUP_ID>"
      }
    }'
```

Sessioni e controllo degli accessi

Comprensione delle sessioni

Gli utenti che accedono Account AWS utilizzando le assegnazioni in Account Access Manager avviano i seguenti tipi di sessione:

1. Sessione di ruolo IAM: il ruolo IAM assunto da un utente è una credenziale temporanea per l'accesso alle AWS risorse con una durata predefinita di 1 ora.
2. Sessione del portale di accesso all'account: gli utenti avviano questa sessione quando utilizzano il portale di accesso all'account. È gestita come sessione applicativa da IAM Identity Center.
3. AWS sessione del portale di accesso (chiamata anche sessione interattiva con l'utente): questa sessione è gestita da IAM Identity Center.
4. Sessione di origine di identità esterna (IdP o Microsoft AD): questa sessione è applicabile quando l'istanza di IAM Identity Center è connessa a una fonte di identità esterna per l'autenticazione.

Per ulteriori informazioni sulle sessioni interattive e applicative con l'utente in IAM Identity Center, inclusa la durata della sessione, consulta Understanding authentication sessions in IAM Identity Center nella Guida per l'Centro identità AWS IAM utente.

> **Important**
>
> Le sessioni dei ruoli IAM assegnati dall'account access manager funzionano in base alla durata predefinita della sessione di 1 ora, indipendentemente dalla durata massima della sessione impostata nel ruolo IAM, dallo stato delle sessioni nel portale di accesso all'account e dal portale di AWS accesso.

Revoca l'accesso degli utenti

Per revocare l'accesso utente assunto tramite le assegnazioni dei ruoli di Account Access Manager, consulta Revoca l'accesso utente nella Guida per l'utente. Centro identità AWS IAM

Attribute-based controllo degli accessi

Puoi configurare il tuo provider di identità (IdP) per inviare attributi come tag di sessione tramite asserzioni SAML impostando il nome dell'attributo su`https://aws.amazon.com/SAML/Attributes/AccessControl:TagKey`, *`TagKey`* dov'è la chiave del tag di sessione che desideri compilare. IAM Identity Center trasmette il nome e il valore dell'attributo dall'IdP per la valutazione delle policy.

Accesso alla forza lavoro

Account Access Manager ti consente di assegnare Account AWS l'accesso agli utenti e ai gruppi della tua forza lavoro in IAM Identity Center mappandoli ai ruoli IAM nel tuo. Account AWS Se non conosci IAM Identity Center, consulta Configurare l'accesso della forza lavoro alle AWS risorse nella Guida per l'Centro identità AWS IAM utente per informazioni sulle sessioni di autenticazione, la gestione degli accessi degli utenti, la gestione delle password e l'autenticazione a più fattori.

Portali utente

Per accedere agli AWS account loro assegnati, gli utenti della forza lavoro possono accedere direttamente al portale di accesso agli account utilizzando un segnalibro o avviarlo dall'interno del AWS portale di accesso.

I tuoi utenti potrebbero interagire con due portali utente:

Portale di accesso all'account

> Si tratta di un portale (URL) che consente agli utenti della forza lavoro di accedere ai dati loro Account AWS assegnati utilizzando l'account access manager. Questo portale (URL) può essere aggiunto ai segnalibri per l'accesso diretto ed è disponibile anche nel portale di AWS accesso come applicazione AWS gestita.

AWS portale di accesso

> Questo è il principale punto di accesso AWS per gli utenti di IAM Identity Center. Questo portale mostra tutte le AWS applicazioni a cui gli utenti hanno accesso, incluso il portale di accesso all'account.

> **Tip**
>
> Consigliamo di aggiungere ai preferiti entrambi i portali per una navigazione più rapida. Se utilizzi un provider di identità esterno, puoi anche creare app di segnalibri per entrambi i portali.

Guida introduttiva al portale di accesso all'account

Individuazione dell'URL del portale di accesso all'account

Per trovare l'URL del portale di accesso all'account, scegli la scheda Impostazioni nella console di gestione dell'accesso all'account. Cerca l'URL dell'applicazione nella sezione Dettagli dell'istanza. L'URL segue lo schema`https://[Tenant-ID].account-access.[Region].app.aws`. Tenant-ID è un identificatore casuale univoco associato all'account access manager.

Se utilizzi il AWS CLI, puoi trovare l'ID del tenant eseguendo il seguente comando CLI nell'account di gestione o di amministratore delegato della tua organizzazione.

```
aws account-access list-applications
```

La risposta contiene informazioni sull'account access manager, inclusi l'ARN dell'applicazione e l'ID del tenant. Utilizza l'ID del tenant e il codice regionale dell'account access manager per creare l'URL dell'applicazione. Segui lo schema fornito in precedenza in questa sezione.

```
{
    "applications": [
        {
            "applicationArn": "arn:aws:account-access:us-
west-2:123456789012:application/1234567890abcdef",
            "tenantId": "aa-gyxmap389",
            "createdAt": "2026-03-27T18:31:19+00:00",
            "updatedAt": "2026-03-27T18:31:19+00:00"
        }
    ]
}
```

Impostazione dell'URL di un vanity portal

Per istruzioni sull'implementazione di vanity domains personalizzati, consulta Regional routing for AWS access portals: Implementazione di vanity domains personalizzati per IAM Identity Center nel Security Blog. AWS Puoi adattare queste istruzioni allo schema URL del portale di accesso all'account.

Comunicare con i tuoi utenti

Comunica quanto segue agli utenti della tua forza lavoro:

- L'URL del portale di accesso all'account.
- Il portale di accesso all'account è disponibile per gli utenti anche come applicazione nella scheda Applicazioni del portale di AWS accesso.
- Se hai abilitato l'avvio del portale di accesso all'account da un portale di provider di identità esterno o da un altro portale esterno AWS, includi queste informazioni nella comunicazione.

Accesso utente a Account AWS

Questo argomento descrive in che modo gli utenti accedono a ciò che è loro Account AWS assegnato con Account Access Manager.

Ottenere l'URL del portale

Puoi accedere ai tuoi dati assegnati Account AWS in diversi modi:

1. L'amministratore potrebbe fornirti un URL del portale di accesso all'account (ad esempio, `https://aa-gyxmap389.account-access.us-east-2.app.aws` o un vanity URL).
2. L'amministratore potrebbe fornirti l'URL del portale di AWS accesso principale che mostra tutte le AWS applicazioni a te assegnate, incluso il portale di accesso all'account.
3. La tua organizzazione potrebbe averti consentito di avviare il portale di accesso all'account da un portale esterno per provider di identità o da un altro portale esterno AWS.

Accesso Account AWS con un browser web

Il portale di accesso agli account mostra a chi Account AWS hai accesso e i ruoli IAM che puoi assumere in ciascun account.

Per accedere a un account in un browser

1. Accedi all'URL del portale di accesso all'account e, quando richiesto, accedi utilizzando le tue credenziali secondo le istruzioni dell'amministratore.

   > **Note**
   >
   > Per avviare invece il portale di accesso all'account dal portale di AWS accesso, accedi alla scheda Applicazioni.

2. La pagina del browser mostra il Account AWS nome a cui sei assegnato. È possibile cercare gli account in base al nome o all'ID dell'account.
3. Espandi l'account di interesse per visualizzare tutti i ruoli che puoi assumere nell'account.
4. Scegli il ruolo IAM con cui accedere (ad esempio, Sviluppatore). Si apre una nuova scheda del browser in cui hai effettuato l'accesso all'account con il ruolo scelto.

Accesso Account AWS con AWS CLI or AWS SDK

Puoi accedere ai AWS servizi in modo programmatico utilizzando i AWS Command Line Interface nostri AWS Software Development Kit (SDK) con credenziali utente di IAM Identity Center.

Il portale di accesso agli account fornisce agli utenti l'accesso Single Sign-on alle assegnazioni degli account gestite in Account Access Account AWS Manager. Dopo aver effettuato l'accesso al portale di accesso all'account, è possibile ottenere credenziali temporanee. Puoi quindi utilizzare le credenziali negli AWS CLI o AWS SDK per accedere alle risorse in un. Account AWS

Se stai utilizzando AWS CLI per accedere ai AWS servizi a livello di programmazione, puoi utilizzare le procedure in questo argomento per avviare l'accesso a. AWS CLI Per informazioni su AWS CLI, consulta la Guida per l'AWS Command Line Interface utente.

Se utilizzi gli AWS SDK per accedere ai AWS servizi in modo programmatico, seguendo le procedure in questo argomento si stabilisce anche direttamente l'autenticazione per gli SDK. AWS Per informazioni sugli SDK, consulta la AWS Guida di riferimento per SDK and Tools AWS .

Account AWS accesso con login aws (consigliato)

Puoi usare il aws login comando per ottenere l'accesso a Account AWS cui sei assegnato in Account Access Manager. Innanzitutto, installa la versione più recente AWS CLI. Quindi, segui questa procedura:

Per accedere a un account con AWS CLI

1. Nel browser predefinito, accedi al portale di accesso all'account e da lì al target Account AWS utilizzando uno dei ruoli IAM assegnati all'account. Segui la procedura descritta in precedenza in questo argomento.
2. Nella riga di comando, esegui il aws login comando.
3. Se non è stata impostata una regione predefinita, AWS CLI viene richiesto di specificare la AWS regione desiderata (ad esempio, us-east-2). AWS CLI Ricorderà la tua scelta per un uso futuro.
4. AWS CLI Apre una pagina Web nel browser predefinito e richiede di continuare con la sessione attiva appena avviata o di accedere a una nuova sessione.
5. Nella pagina web, scegli il riquadro con la sessione avviata in precedenza nel browser.
6. La pagina del browser conferma che le credenziali sono state condivise correttamente e possono essere utilizzate fino alla scadenza della sessione. È ora possibile chiudere la scheda con la pagina Web e continuare a AWS CLI utilizzare questa sessione.

> **Note**
>
> Center-specific AWS CLI I comandi IAM Identity come aws sso configure e aws sso login non funzionano con le assegnazioni degli account in Account Access Manager.

Account AWS accesso con chiavi di accesso

Puoi utilizzare il metodo di aggiornamento manuale delle credenziali per ottenere credenziali temporanee per un ruolo IAM assegnato a te in uno specifico ruolo. Account AWS A tale scopo, è necessario copiare e incollare i comandi richiesti per le credenziali temporanee. Con questo metodo, è necessario aggiornare manualmente le credenziali temporanee.

È possibile eseguire AWS CLI comandi fino alla scadenza delle credenziali temporanee.

Per ottenere le credenziali da aggiornare manualmente

1. Accedi al portale di accesso all'account come spiegato in Accedere Account AWS con un browser web.
2. Individua il nome Account AWS da cui desideri recuperare le credenziali di accesso ed espandilo per mostrare il nome del ruolo IAM (ad esempio Developer).

3. Scegli Genera chiave di accesso accanto al ruolo desiderato.

> **Note**
>
> Se non vedi alcuna Account AWS assegnazione, è probabile che non sei ancora stato assegnato a nessun account. In questo caso, contatta il tuo amministratore e chiedigli di assegnarti l'accesso.

4. Nella finestra di dialogo Ottieni credenziali, scegli macOS e Linux , Windows o la PowerShell scheda a seconda del sistema operativo su cui hai installato. AWS CLI
5. Selezionare una delle seguenti opzioni:

   - Opzione 1: imposta AWS le variabili di ambiente

     Scegli questa opzione per sovrascrivere tutte le impostazioni delle credenziali, incluse le impostazioni nei file delle credenziali e nei file di configurazione. Per ulteriori informazioni, consulta Variabili di ambiente da configurare AWS CLI nella Guida per l'utente. AWS Command Line Interface

     Per utilizzare questa opzione, copiate i comandi negli appunti, incollateli nella finestra del AWS CLI terminale e premete Invio per impostare le variabili di ambiente richieste.
   - Opzione 2: utilizza valori individuali nel tuo client di servizio AWS

     Scegli questa opzione per accedere alle AWS risorse di un client AWS di servizio. Per ulteriori informazioni, consulta Strumenti su cui basarsi AWS.

     Per utilizzare questa opzione, copia i valori negli appunti, incolla i valori nel codice e assegnali alle variabili appropriate per il tuo SDK. Per ulteriori informazioni, consulta la documentazione per la tua API SDK specifica.

Creazione di collegamenti rapidi a AWS Destinazioni della console di gestione

Il portale di accesso all'account offre un'opzione per creare un link diretto condivisibile.

Per creare un collegamento rapido

1. Dopo aver effettuato l'accesso al portale di accesso all'account, scegli il pulsante Crea collegamento rapido.
2. Nella finestra di dialogo:

a. Scegli un Account AWS utilizzando l'ID o il nome dell'account. Mentre digiti, un menu a discesa mostra gli ID e i nomi degli account corrispondenti a cui puoi accedere. Puoi scegliere solo un account a cui hai accesso.

b. (Facoltativo) Scegli un ruolo IAM dall'elenco a discesa. Questi sono i ruoli IAM che ti sono stati assegnati per l'account selezionato. Se si omette di scegliere il ruolo, agli utenti viene richiesto di selezionarne uno assegnato per l'account scelto quando si utilizza il collegamento di scelta rapida.

> **Note**
>
> Non è possibile concedere un nuovo accesso con collegamenti rapidi. I link di scelta rapida funzionano solo con gli account e i ruoli già assegnati all'utente.

Costruire in modo sicuro AWS Collegamenti rapidi della console di gestione con codifica URL

Tutti i valori dei parametri dell'URL, inclusi l'ID dell'account, il nome del ruolo IAM e l'URL di destinazione, devono esserlo. URL-encoded

I link di scelta rapida estendono l'URL del portale di accesso all'account con il seguente percorso:

```
/#/console?account_id=[account_ID]&role_name=[role_name]&destination=[destination_URL]
```

L'URL completo in una AWS regione commerciale segue questo schema:

```
https://[Tenant-ID].account-access.[Region].app.aws/#/console?
account_id=[account_ID]&role_name=[role_name]&destination=[destination_URL]
```

Ecco un esempio di collegamento rapido che consente di accedere all'account 123456789012 con il nome del ruolo S3FullAccess IAM e di reindirizzare l'utente alla home page della console S3:

```
https://aa-gyxmap389.account-access.us-east-2.app.aws/#/console?
account_id=123456789012&role_name=S3FullAccess&destination=https%3A%2F
%2Fconsole.aws.amazon.com%2Fs3%2Fhome
```

Considerazioni

Le seguenti considerazioni si applicano all'uso dei collegamenti rapidi:

- Se condividi collegamenti rapidi con i colleghi, devono avere le stesse assegnazioni di account per poter utilizzare i collegamenti nello stesso modo.
- I link di scelta rapida sono collegamenti regionali. Se la regione di riferimento sta subendo un'interruzione, il collegamento potrebbe non funzionare.
- I collegamenti rapidi contengono l'ID dell'account e il nome del ruolo in testo normale. Valuta se le politiche della tua organizzazione limitano la posizione in cui sono archiviati questi metadati.

Sicurezza nel gestore degli accessi agli account

Questa sezione fornisce informazioni di sicurezza specifiche per Account Access Manager. Integra le informazioni contenute inSicurezza, che riguardano gli argomenti di sicurezza applicabili a IAM nel suo insieme.

Gestione delle identità e degli accessi per l'account access manager

Le seguenti sezioni descrivono le autorizzazioni necessarie per:

- Amministra l'account access manager nell'account di gestione della tua organizzazione
- Amministra l'account access manager da un account amministratore delegato

Amministrazione dell'account access manager dall'account di gestione dell'organizzazione

Per amministrare le impostazioni dell'account access manager nell'account di gestione dell' AWS organizzazione sono necessarie le seguenti autorizzazioni:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AAMCreateApplication",
      "Effect": "Allow",
      "Action": "account-access:CreateApplication",
      "Resource": "*"
    },
    {
      "Sid": "AAMApplicationAndEntitlementManagement",
      "Effect": "Allow",
```

```
      "Action": [
        "account-access:CreateEntitlement",
        "account-access:DeleteApplication",
        "account-access:DeleteEntitlement",
        "account-access:GetApplication",
        "account-access:GetEntitlement",
        "account-access:ListEntitlements",
        "account-access:ListTagsForResource",
        "account-access:TagResource",
        "account-access:UntagResource"
      ],
      "Resource": "arn:aws:account-access:*:111122223333:application/*"
    },
    {
      "Sid": "AAMListApplications",
      "Effect": "Allow",
      "Action": "account-access:ListApplications",
      "Resource": "*"
    },
    {
      "Sid": "DependentIdentityCenterActions",
      "Effect": "Allow",
      "Action": [
        "sso:CreateApplication",
        "sso:DeleteApplication",
        "sso:DescribeApplication",
        "sso:DescribeInstance",
        "sso:DescribeRegion",
        "sso:ListInstances",
        "sso:PutApplicationAccessScope",
        "sso:PutApplicationAssignmentConfiguration",
        "sso:PutApplicationAuthenticationMethod",
        "sso:PutApplicationGrant"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DependentIdentityStoreActions",
      "Effect": "Allow",
      "Action": [
        "identitystore:DescribeGroup",
        "identitystore:DescribeUser",
        "identitystore:ListGroupMemberships",
        "sso-directory:DescribeGroup",
```

```
        "sso-directory:DescribeGroups",
        "sso-directory:DescribeUser",
        "sso-directory:DescribeUsers",
        "sso-directory:ListGroupsForUser",
        "sso-directory:ListMembersInGroup",
        "sso-directory:SearchGroups",
        "sso-directory:SearchUsers"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DependentOrganizationsActions",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:ListAccounts",
        "organizations:ListAccountsForParent",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListRoots"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DependentKMSActions",
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "*"
    },
    {
      "Sid": "CreateServiceLinkedRole",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::111122223333:role/aws-service-role/account-
access.amazonaws.com/AWSServiceRoleForAccountAccessManager",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "account-access.amazonaws.com"
        }
      }
    }
  ]
```

```
}
```

Facoltativo: per vincolare `account-access:CreateApplication` tramite tag, sostituisci la dichiarazione con AAMCreateApplication:

```
{
  "Sid": "AAMCreateApplicationConstrainedByTag",
  "Effect": "Allow",
  "Action": "account-access:CreateApplication",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:RequestTag/Environment": "Production"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "Environment"
      ]
    }
  }
}
```

Facoltativo: per gestire e registrare gli amministratori delegati per l'account access manager, aggiungi:

```
{
  "Sid": "ManageDelegatedAdministrator",
  "Effect": "Allow",
  "Action": [
    "organizations:DeregisterDelegatedAdministrator",
    "organizations:RegisterDelegatedAdministrator"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "organizations:ServicePrincipal": "account-access.amazonaws.com"
    }
  }
}
```

Per assegnare ruoli IAM agli utenti e ai gruppi della forza lavoro dall'account di gestione dell' AWS organizzazione senza amministrare le impostazioni dell'account access manager, sono necessarie le seguenti autorizzazioni:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AAMManageEntitlementsOnly",
      "Effect": "Allow",
      "Action": [
        "account-access:CreateEntitlement",
        "account-access:DeleteEntitlement",
        "account-access:GetApplication",
        "account-access:GetEntitlement",
        "account-access:ListEntitlements"
      ],
      "Resource": "arn:aws:account-access:*:111122223333:application/*"
    },
    {
      "Sid": "AAMListApplications",
      "Effect": "Allow",
      "Action": "account-access:ListApplications",
      "Resource": "*"
    },
    {
      "Sid": "DependentIdentityCenterActions",
      "Effect": "Allow",
      "Action": "sso:ListInstances",
      "Resource": "*"
    },
    {
      "Sid": "DependentIdentityStoreActions",
      "Effect": "Allow",
      "Action": [
        "identitystore:DescribeGroup",
        "identitystore:DescribeUser",
        "identitystore:ListGroupMemberships",
        "sso-directory:DescribeGroup",
        "sso-directory:DescribeGroups",
        "sso-directory:DescribeUser",
        "sso-directory:DescribeUsers",
        "sso-directory:ListGroupsForUser",
```

```
        "sso-directory:ListMembersInGroup",
        "sso-directory:SearchGroups",
        "sso-directory:SearchUsers"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DependentOrganizationsActions",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:ListAccounts",
        "organizations:ListAccountsForParent",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListRoots"
      ],
      "Resource": "*"
    }
  ]
}
```

Amministrazione dell'account access manager in un account di amministrazione delegato

Per amministrare le impostazioni dell'account access manager da un account di amministrazione delegato sono necessarie le seguenti autorizzazioni:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AAMDelegatedAdminManagement",
      "Effect": "Allow",
      "Action": [
        "account-access:CreateEntitlement",
        "account-access:DeleteEntitlement",
        "account-access:GetApplication",
        "account-access:GetEntitlement",
        "account-access:ListEntitlements",
        "account-access:ListTagsForResource",
```

```
        "account-access:TagResource",
        "account-access:UntagResource"
      ],
      "Resource": "arn:aws:account-access:*:111122223333:application/*"
    },
    {
      "Sid": "AAMListApplications",
      "Effect": "Allow",
      "Action": "account-access:ListApplications",
      "Resource": "*"
    },
    {
      "Sid": "DependentIdentityCenterActions",
      "Effect": "Allow",
      "Action": [
        "sso:DescribeApplication",
        "sso:DescribeInstance",
        "sso:DescribeRegion",
        "sso:ListInstances"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DependentIdentityStoreActions",
      "Effect": "Allow",
      "Action": [
        "identitystore:DescribeGroup",
        "identitystore:DescribeUser",
        "identitystore:ListGroupMemberships",
        "sso-directory:DescribeGroup",
        "sso-directory:DescribeGroups",
        "sso-directory:DescribeUser",
        "sso-directory:DescribeUsers",
        "sso-directory:ListGroupsForUser",
        "sso-directory:ListMembersInGroup",
        "sso-directory:SearchGroups",
        "sso-directory:SearchUsers"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DependentOrganizationsActions",
      "Effect": "Allow",
      "Action": [
```

```
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:ListAccounts",
        "organizations:ListAccountsForParent",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListRoots"
      ],
      "Resource": "*"
    }
  ]
}
```

Per assegnare ruoli IAM agli utenti e ai gruppi della forza lavoro da un account di amministrazione delegato senza amministrare le impostazioni dell'account access manager, sono necessarie le seguenti autorizzazioni:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AAMManageEntitlementsOnly",
      "Effect": "Allow",
      "Action": [
        "account-access:CreateEntitlement",
        "account-access:DeleteEntitlement",
        "account-access:GetApplication",
        "account-access:GetEntitlement",
        "account-access:ListEntitlements"
      ],
      "Resource": "arn:aws:account-access:*:111122223333:application/*"
    },
    {
      "Sid": "AAMListApplications",
      "Effect": "Allow",
      "Action": "account-access:ListApplications",
      "Resource": "*"
    },
    {
      "Sid": "DependentIdentityCenterActions",
      "Effect": "Allow",
      "Action": "sso:ListInstances",
```

```
      "Resource": "*"
    },
    {
      "Sid": "DependentIdentityStoreActions",
      "Effect": "Allow",
      "Action": [
        "identitystore:DescribeGroup",
        "identitystore:DescribeUser",
        "identitystore:ListGroupMemberships",
        "sso-directory:DescribeGroup",
        "sso-directory:DescribeGroups",
        "sso-directory:DescribeUser",
        "sso-directory:DescribeUsers",
        "sso-directory:ListGroupsForUser",
        "sso-directory:ListMembersInGroup",
        "sso-directory:SearchGroups",
        "sso-directory:SearchUsers"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DependentOrganizationsActions",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:ListAccounts",
        "organizations:ListAccountsForParent",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListRoots"
      ],
      "Resource": "*"
    }
  ]
}
```

Protezione dei dati in Account Access Manager

Questa sezione integra le informazioni fornite in Protezione dei dati AWS Identity and Access Management con dettagli aggiuntivi specifici per l'account access manager.

Utilizzo dei dati degli utenti e dei gruppi di IAM Identity Center

L'Account Access Manager utilizza i dati di utenti e gruppi provenienti da IAM Identity Center per gestire le assegnazioni degli account. Tuttavia, l'account access manager memorizza e registra solo gli identificatori AWS generati (ID utente e ID di gruppo) e non memorizza né registra nomi utente, indirizzi email o altri attributi personali.

Registrazione e monitoraggio in Account Access Manager

Account Access Manager utilizza il proprio namespace API (`account-access`) e registra le chiamate API tramite. CloudTrail

Per ulteriori informazioni CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

Quote e limiti nell'account access manager

Le tabelle seguenti descrivono le quote all'interno di Account Access Manager. Le richieste di aumento delle quote devono provenire da un account di gestione o amministratore delegato. Per aumentare una quota, vedi Richiedere un aumento della quota.

Quota di assegnazione di gruppo

Risorsa	Quota predefinita	Può essere aumentata
Numero massimo di gruppi assegnati a un ruolo IAM	20	Sì

Limiti di accelerazione

Risorsa	Quota predefinita	Può essere aumentata
API di accesso all'account	L'API di accesso all'account ha un limite di limitazione collettivo di 20 transazioni al secondo (TPS). Per le API di lettura, puoi aprire	Sì

Risorsa	Quota predefinita	Può essere aumentata
	una richiesta di assistenza per richiedere un aumento del limite. Le operazioni API di scrittura hanno un limite di 15 chiamate asincrone in sospeso. Questo limite non può essere aumentato.	

Esempi di codice per l'utilizzo di IAM AWS SDK

I seguenti esempi di codice mostrano come utilizzare IAM con un kit di sviluppo AWS software (SDK).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Esempi di codice per l'utilizzo di IAM AWS SDK
  - Esempi di base per l'utilizzo di IAM AWS SDK
    - Hello IAM
    - Scopri le basi di IAM con un AWS SDK
    - Azioni per l'utilizzo di IAM AWS SDK
      - Utilizzo di AddClientIdToOpenIdConnectProvider con una CLI
      - Utilizzo di AddRoleToInstanceProfile con una CLI
      - Utilizzo di AddUserToGroup con una CLI
      - Utilizzo di AttachGroupPolicy con una CLI
      - Usare AttachRolePolicycon un AWS SDK o CLI
      - Usare AttachUserPolicycon un AWS SDK o CLI
      - Utilizzo di ChangePassword con una CLI
      - Utilizzare CreateAccessKeycon un AWS SDK o CLI
      - Usare CreateAccountAliascon un AWS SDK o CLI
      - Usare CreateGroupcon un AWS SDK o CLI
      - Usare CreateInstanceProfilecon un AWS SDK o CLI
      - Utilizzo di CreateLoginProfile con una CLI
      - Utilizzo di CreateOpenIdConnectProvider con una CLI
      - Utilizzare CreatePolicycon un AWS SDK o CLI
      - Usare CreatePolicyVersioncon un AWS SDK o CLI
      - Usare CreateRolecon un AWS SDK o CLI
      - Usa CreateSAMLProvider con un AWS SDK o CLI
      - Utilizzare CreateServiceLinkedRolecon un AWS SDK o CLI

- Utilizzare CreateUsercon un AWS SDK o CLI
- Utilizzo di CreateVirtualMfaDevice con una CLI
- Utilizzo di DeactivateMfaDevice con una CLI
- Usare DeleteAccessKeycon un AWS SDK o CLI
- Usare DeleteAccountAliascon un AWS SDK o CLI
- Utilizzo di DeleteAccountPasswordPolicy con una CLI
- Usare DeleteGroupcon un AWS SDK o CLI
- Utilizzo di DeleteGroupPolicy con una CLI
- Usare DeleteInstanceProfilecon un AWS SDK o CLI
- Utilizzo di DeleteLoginProfile con una CLI
- Utilizzo di DeleteOpenIdConnectProvider con una CLI
- Usare DeletePolicycon un AWS SDK o CLI
- Usare DeletePolicyVersioncon un AWS SDK o CLI
- Usare DeleteRolecon un AWS SDK o CLI
- Utilizzo di DeleteRolePermissionsBoundary con una CLI
- Usare DeleteRolePolicycon un AWS SDK o CLI
- Usa DeleteSamlProvider con un AWS SDK o CLI
- Usare DeleteServerCertificatecon un AWS SDK o CLI
- Usare DeleteServiceLinkedRolecon un AWS SDK o CLI
- Utilizzo di DeleteSigningCertificate con una CLI
- Utilizzare DeleteUsercon un AWS SDK o CLI
- Utilizzo di DeleteUserPermissionsBoundary con una CLI
- Usare DeleteUserPolicycon un AWS SDK o CLI
- Utilizzo di DeleteVirtualMfaDevice con una CLI
- Utilizzo di DetachGroupPolicy con una CLI
- Utilizzare DetachRolePolicycon un AWS SDK o CLI
- Usare DetachUserPolicycon un AWS SDK o CLI
- Utilizzo di EnableMfaDevice con una CLI
- Usare GenerateCredentialReportcon un AWS SDK o CLI
- Utilizzo di GenerateServiceLastAccessedDetails con una CLI

- Usare GetAccessKeyLastUsedcon un AWS SDK o CLI
- Utilizzare GetAccountAuthorizationDetailscon un AWS SDK o CLI
- Usare GetAccountPasswordPolicycon un AWS SDK o CLI
- Utilizzare GetAccountSummarycon un AWS SDK o CLI
- Utilizzo di GetContextKeysForCustomPolicy con una CLI
- Utilizzo di GetContextKeysForPrincipalPolicy con una CLI
- Usare GetCredentialReportcon un AWS SDK o CLI
- Utilizzo di GetGroup con una CLI
- Utilizzo di GetGroupPolicy con una CLI
- Utilizzo di GetInstanceProfile con una CLI
- Utilizzo di GetLoginProfile con una CLI
- Utilizzo di GetOpenIdConnectProvider con una CLI
- Usare GetPolicycon un AWS SDK o CLI
- Usare GetPolicyVersioncon un AWS SDK o CLI
- Utilizzare GetRolecon un AWS SDK o CLI
- Utilizzo di GetRolePolicy con una CLI
- Utilizzo di GetSamlProvider con una CLI
- Usare GetServerCertificatecon un AWS SDK o CLI
- Utilizzo di GetServiceLastAccessedDetails con una CLI
- Utilizzo di GetServiceLastAccessedDetailsWithEntities con una CLI
- Utilizzare GetServiceLinkedRoleDeletionStatuscon un AWS SDK o CLI
- Usare GetUsercon un AWS SDK o CLI
- Utilizzo di GetUserPolicy con una CLI
- Usare ListAccessKeyscon un AWS SDK o CLI
- Utilizzare ListAccountAliasescon un AWS SDK o CLI
- Utilizzo di ListAttachedGroupPolicies con una CLI
- Utilizzare ListAttachedRolePoliciescon un AWS SDK o CLI
- Utilizzo di ListAttachedUserPolicies con una CLI
- Utilizzo di ListEntitiesForPolicy con una CLI
- Utilizzo di ListGroupPolicies con una CLI

- Utilizzare ListGroupscon un AWS SDK o CLI
- Utilizzo di ListGroupsForUser con una CLI
- Utilizzo di ListInstanceProfiles con una CLI
- Utilizzo di ListInstanceProfilesForRole con una CLI
- Utilizzo di ListMfaDevices con una CLI
- Utilizzo di ListOpenIdConnectProviders con una CLI
- Utilizzare ListPoliciescon un AWS SDK o CLI
- Usare ListPolicyVersionscon un AWS SDK o CLI
- Utilizzare ListRolePoliciescon un AWS SDK o CLI
- Utilizzo di ListRoleTags con una CLI
- Utilizzare ListRolescon un AWS SDK o CLI
- Usa ListSamlProviders con un AWS SDK o CLI
- Utilizzare ListServerCertificatescon un AWS SDK o CLI
- Utilizzo di ListSigningCertificates con una CLI
- Usare ListUserPoliciescon un AWS SDK o CLI
- Utilizzo di ListUserTags con una CLI
- Usare ListUserscon un AWS SDK o CLI
- Utilizzo di ListVirtualMfaDevices con una CLI
- Utilizzo di PutGroupPolicy con una CLI
- Utilizzo di PutRolePermissionsBoundary con una CLI
- Utilizzare PutRolePolicycon un AWS SDK o CLI
- Utilizzo di PutUserPermissionsBoundary con una CLI
- Usare PutUserPolicycon un AWS SDK o CLI
- Utilizzo di RemoveClientIdFromOpenIdConnectProvider con una CLI
- Utilizzo di RemoveRoleFromInstanceProfile con una CLI
- Utilizzo di RemoveUserFromGroup con una CLI
- Utilizzo di ResyncMfaDevice con una CLI
- Usare SetDefaultPolicyVersioncon un AWS SDK o CLI
- Utilizzo di TagRole con una CLI
- Utilizzo di TagUser con una CLI

    - Utilizzo di UntagRole con una CLI
    - Utilizzo di UntagUser con una CLI
    - Utilizzare UpdateAccessKeycon un AWS SDK o CLI
    - Utilizzo di UpdateAccountPasswordPolicy con una CLI
    - Utilizzo di UpdateAssumeRolePolicy con una CLI
    - Utilizzo di UpdateGroup con una CLI
    - Utilizzo di UpdateLoginProfile con una CLI
    - Utilizzo di UpdateOpenIdConnectProviderThumbprint con una CLI
    - Utilizzo di UpdateRole con una CLI
    - Utilizzo di UpdateRoleDescription con una CLI
    - Utilizzo di UpdateSamlProvider con una CLI
    - Usare UpdateServerCertificatecon un AWS SDK o CLI
    - Utilizzo di UpdateSigningCertificate con una CLI
    - Usare UpdateUsercon un AWS SDK o CLI
    - Utilizzare UploadServerCertificatecon un AWS SDK o CLI
    - Utilizzo di UploadSigningCertificate con una CLI
- Scenari per l'utilizzo di IAM AWS SDK
    - Crea e gestisci un servizio resiliente utilizzando un AWS SDK
    - Configurare la connettività dei servizi container
    - Crea un'attività contenitore per il tipo di avvio serverless
    - Crea un'API REST con integrazione proxy di funzioni
    - Crea utenti IAM di sola lettura e lettura-scrittura utilizzando un AWS SDK
    - Creazione di un servizio container per le istanze di macchine virtuali
    - Creazione di un'area di lavoro di monitoraggio gestita
    - Creazione di un dashboard di monitoraggio con il nome della funzione come variabile
    - Creazione della tua prima funzione serverless
    - Inizia a usare i data warehouse serverless utilizzando la CLI
    - Guida introduttiva alla gestione della configurazione
    - Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
    - Guida introduttiva ai feature store di machine learning

- Guida introduttiva ai cluster Kubernetes gestiti
- Guida introduttiva allo streaming gestito
- Guida introduttiva ai cluster di data warehouse con provisioning
- Guida introduttiva all'orchestrazione del flusso di lavoro
- Gestisci le chiavi di accesso IAM utilizzando un AWS SDK
- Gestisci le policy IAM utilizzando un AWS SDK
- Gestisci i ruoli IAM utilizzando un AWS SDK
- Gestisci il tuo account IAM utilizzando un AWS SDK
- Spostamento di segreti codificati per proteggere l'archiviazione segreta
- Permette AWS Compute Optimizer Funzionalità di automazione per applicare le azioni consigliate
- Politica per abilitare l'automazione in tutta l'organizzazione
- Politica per abilitare l'automazione per il tuo account
- Politica per concedere l'accesso completo a Compute Optimizer Automation per un account di gestione di un'organizzazione
- Politica per concedere l'accesso completo a Compute Optimizer Automation per la versione standalone AWS account
- Politica per concedere l'accesso in sola lettura a Compute Optimizer Automation per un account di gestione di un'organizzazione
- Politica per concedere l'accesso in sola lettura a Compute Optimizer Automation per uso autonomo AWS account
- Politica per la concessione di autorizzazioni di ruolo collegate ai servizi per Compute Optimization Automation
- Ripristina una versione della policy IAM utilizzando un AWS SDK
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori
- Configurare il controllo Attribute-Based degli accessi per DynamoDB utilizzando AWS Command Line Interface v2
- Configurazione della gestione dei sistemi
- Utilizzo delle variabili di proprietà nei dashboard di monitoraggio per monitorare più funzioni serverless
- Lavora con DynamoDB Streams e utilizza Time-to-Live AWS Command Line Interface v2
- Lavora con l'API IAM Policy Builder utilizzando un AWS SDK

- Esempi di codice per AWS STS utilizzo di AWS SDK
  - Esempi di base per AWS STS utilizzo di AWS SDK
    - Azioni per AWS STS utilizzo di AWS SDK
      - Usare AssumeRolecon un AWS SDK o CLI
      - Utilizzo di AssumeRoleWithWebIdentity con una CLI
      - Utilizzo di DecodeAuthorizationMessage con una CLI
      - Utilizzo di GetFederationToken con una CLI
      - Usare GetSessionTokencon un AWS SDK o CLI
  - Scenari per AWS STS utilizzo di AWS SDK
    - Assumi un ruolo IAM che richiede un token MFA con AWS STS utilizzando un AWS SDK
    - Configurare la connettività dei servizi container
    - Costruisci un URL con AWS STS per utenti federati che utilizzano un AWS SDK
    - Crea un'attività contenitore per il tipo di avvio serverless
    - Crea un'API REST con integrazione proxy di funzioni
    - Creazione di un servizio container per le istanze di macchine virtuali
    - Creazione di un'area di lavoro di monitoraggio gestita
    - Creazione di un dashboard di monitoraggio con il nome della funzione come variabile
    - Ottieni un token di sessione che richiede un token MFA con AWS STS utilizzando un AWS SDK
    - Guida introduttiva alla gestione della configurazione
    - Guida introduttiva ai registri dei container
    - Guida introduttiva alla messaggistica sull'Internet of Things
    - Guida introduttiva ai feature store di machine learning
    - Guida introduttiva ai cluster Kubernetes gestiti
    - Guida introduttiva allo streaming gestito
    - Guida introduttiva alle notifiche push
    - Guida introduttiva ai motori di ricerca e analisi
    - Guida introduttiva ai firewall per applicazioni Web
    - Spostamento di segreti codificati per proteggere l'archiviazione segreta
    - Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori
    - Configurazione della gestione dei sistemi

Esempi di codice per l'utilizzo di IAM AWS SDK

I seguenti esempi di codice mostrano come utilizzare IAM con un kit di sviluppo AWS software (SDK).

Nozioni di base: esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le azioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica chiamando più funzioni all'interno dello stesso servizio o combinate con altri Servizi AWS.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

IAM

- Esempi di base per l'utilizzo di IAM AWS SDK
  - Hello IAM
  - Scopri le basi di IAM con un AWS SDK
  - Azioni per l'utilizzo di IAM AWS SDK
    - Utilizzo di AddClientIdToOpenIdConnectProvider con una CLI
    - Utilizzo di AddRoleToInstanceProfile con una CLI
    - Utilizzo di AddUserToGroup con una CLI
    - Utilizzo di AttachGroupPolicy con una CLI
    - Usare AttachRolePolicycon un AWS SDK o CLI
    - Usare AttachUserPolicycon un AWS SDK o CLI
    - Utilizzo di ChangePassword con una CLI
    - Utilizzare CreateAccessKeycon un AWS SDK o CLI
    - Usare CreateAccountAliascon un AWS SDK o CLI
    - Usare CreateGroupcon un AWS SDK o CLI
    - Usare CreateInstanceProfilecon un AWS SDK o CLI
    - Utilizzo di CreateLoginProfile con una CLI

- Utilizzo di CreateOpenIdConnectProvider con una CLI
- Utilizzare CreatePolicycon un AWS SDK o CLI
- Usare CreatePolicyVersioncon un AWS SDK o CLI
- Usare CreateRolecon un AWS SDK o CLI
- Usa CreateSAMLProvider con un AWS SDK o CLI
- Utilizzare CreateServiceLinkedRolecon un AWS SDK o CLI
- Utilizzare CreateUsercon un AWS SDK o CLI
- Utilizzo di CreateVirtualMfaDevice con una CLI
- Utilizzo di DeactivateMfaDevice con una CLI
- Usare DeleteAccessKeycon un AWS SDK o CLI
- Usare DeleteAccountAliascon un AWS SDK o CLI
- Utilizzo di DeleteAccountPasswordPolicy con una CLI
- Usare DeleteGroupcon un AWS SDK o CLI
- Utilizzo di DeleteGroupPolicy con una CLI
- Usare DeleteInstanceProfilecon un AWS SDK o CLI
- Utilizzo di DeleteLoginProfile con una CLI
- Utilizzo di DeleteOpenIdConnectProvider con una CLI
- Usare DeletePolicycon un AWS SDK o CLI
- Usare DeletePolicyVersioncon un AWS SDK o CLI
- Usare DeleteRolecon un AWS SDK o CLI
- Utilizzo di DeleteRolePermissionsBoundary con una CLI
- Usare DeleteRolePolicycon un AWS SDK o CLI
- Usa DeleteSamlProvider con un AWS SDK o CLI
- Usare DeleteServerCertificatecon un AWS SDK o CLI
- Usare DeleteServiceLinkedRolecon un AWS SDK o CLI
- Utilizzo di DeleteSigningCertificate con una CLI
- Utilizzare DeleteUsercon un AWS SDK o CLI
- Utilizzo di DeleteUserPermissionsBoundary con una CLI
- Usare DeleteUserPolicycon un AWS SDK o CLI
- Utilizzo di DeleteVirtualMfaDevice con una CLI

- Utilizzo di DetachGroupPolicy con una CLI
- Utilizzare DetachRolePolicycon un AWS SDK o CLI
- Usare DetachUserPolicycon un AWS SDK o CLI
- Utilizzo di EnableMfaDevice con una CLI
- Usare GenerateCredentialReportcon un AWS SDK o CLI
- Utilizzo di GenerateServiceLastAccessedDetails con una CLI
- Usare GetAccessKeyLastUsedcon un AWS SDK o CLI
- Utilizzare GetAccountAuthorizationDetailscon un AWS SDK o CLI
- Usare GetAccountPasswordPolicycon un AWS SDK o CLI
- Utilizzare GetAccountSummarycon un AWS SDK o CLI
- Utilizzo di GetContextKeysForCustomPolicy con una CLI
- Utilizzo di GetContextKeysForPrincipalPolicy con una CLI
- Usare GetCredentialReportcon un AWS SDK o CLI
- Utilizzo di GetGroup con una CLI
- Utilizzo di GetGroupPolicy con una CLI
- Utilizzo di GetInstanceProfile con una CLI
- Utilizzo di GetLoginProfile con una CLI
- Utilizzo di GetOpenIdConnectProvider con una CLI
- Usare GetPolicycon un AWS SDK o CLI
- Usare GetPolicyVersioncon un AWS SDK o CLI
- Utilizzare GetRolecon un AWS SDK o CLI
- Utilizzo di GetRolePolicy con una CLI
- Utilizzo di GetSamlProvider con una CLI
- Usare GetServerCertificatecon un AWS SDK o CLI
- Utilizzo di GetServiceLastAccessedDetails con una CLI
- Utilizzo di GetServiceLastAccessedDetailsWithEntities con una CLI
- Utilizzare GetServiceLinkedRoleDeletionStatuscon un AWS SDK o CLI
- Usare GetUsercon un AWS SDK o CLI
- Utilizzo di GetUserPolicy con una CLI
- Usare ListAccessKeyscon un AWS SDK o CLI

- Utilizzare ListAccountAliasescon un AWS SDK o CLI
- Utilizzo di ListAttachedGroupPolicies con una CLI
- Utilizzare ListAttachedRolePoliciescon un AWS SDK o CLI
- Utilizzo di ListAttachedUserPolicies con una CLI
- Utilizzo di ListEntitiesForPolicy con una CLI
- Utilizzo di ListGroupPolicies con una CLI
- Utilizzare ListGroupscon un AWS SDK o CLI
- Utilizzo di ListGroupsForUser con una CLI
- Utilizzo di ListInstanceProfiles con una CLI
- Utilizzo di ListInstanceProfilesForRole con una CLI
- Utilizzo di ListMfaDevices con una CLI
- Utilizzo di ListOpenIdConnectProviders con una CLI
- Utilizzare ListPoliciescon un AWS SDK o CLI
- Usare ListPolicyVersionscon un AWS SDK o CLI
- Utilizzare ListRolePoliciescon un AWS SDK o CLI
- Utilizzo di ListRoleTags con una CLI
- Utilizzare ListRolescon un AWS SDK o CLI
- Usa ListSamlProviders con un AWS SDK o CLI
- Utilizzare ListServerCertificatescon un AWS SDK o CLI
- Utilizzo di ListSigningCertificates con una CLI
- Usare ListUserPoliciescon un AWS SDK o CLI
- Utilizzo di ListUserTags con una CLI
- Usare ListUserscon un AWS SDK o CLI
- Utilizzo di ListVirtualMfaDevices con una CLI
- Utilizzo di PutGroupPolicy con una CLI
- Utilizzo di PutRolePermissionsBoundary con una CLI
- Utilizzare PutRolePolicycon un AWS SDK o CLI
- Utilizzo di PutUserPermissionsBoundary con una CLI
- Usare PutUserPolicycon un AWS SDK o CLI
- Utilizzo di RemoveClientIdFromOpenIdConnectProvider con una CLI

- Utilizzo di RemoveRoleFromInstanceProfile con una CLI
- Utilizzo di RemoveUserFromGroup con una CLI
- Utilizzo di ResyncMfaDevice con una CLI
- Usare SetDefaultPolicyVersioncon un AWS SDK o CLI
- Utilizzo di TagRole con una CLI
- Utilizzo di TagUser con una CLI
- Utilizzo di UntagRole con una CLI
- Utilizzo di UntagUser con una CLI
- Utilizzare UpdateAccessKeycon un AWS SDK o CLI
- Utilizzo di UpdateAccountPasswordPolicy con una CLI
- Utilizzo di UpdateAssumeRolePolicy con una CLI
- Utilizzo di UpdateGroup con una CLI
- Utilizzo di UpdateLoginProfile con una CLI
- Utilizzo di UpdateOpenIdConnectProviderThumbprint con una CLI
- Utilizzo di UpdateRole con una CLI
- Utilizzo di UpdateRoleDescription con una CLI
- Utilizzo di UpdateSamlProvider con una CLI
- Usare UpdateServerCertificatecon un AWS SDK o CLI
- Utilizzo di UpdateSigningCertificate con una CLI
- Usare UpdateUsercon un AWS SDK o CLI
- Utilizzare UploadServerCertificatecon un AWS SDK o CLI
- Utilizzo di UploadSigningCertificate con una CLI

- Scenari per l'utilizzo di IAM AWS SDK
  - Crea e gestisci un servizio resiliente utilizzando un AWS SDK
  - Configurare la connettività dei servizi container
  - Crea un'attività contenitore per il tipo di avvio serverless
  - Crea un'API REST con integrazione proxy di funzioni
  - Crea utenti IAM di sola lettura e lettura-scrittura utilizzando un AWS SDK
  - Creazione di un servizio container per le istanze di macchine virtuali
  - Creazione di un'area di lavoro di monitoraggio gestita

- Creazione di un dashboard di monitoraggio con il nome della funzione come variabile
- Creazione della tua prima funzione serverless
- Inizia a usare i data warehouse serverless utilizzando la CLI
- Guida introduttiva alla gestione della configurazione
- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
- Guida introduttiva ai feature store di machine learning
- Guida introduttiva ai cluster Kubernetes gestiti
- Guida introduttiva allo streaming gestito
- Guida introduttiva ai cluster di data warehouse con provisioning
- Guida introduttiva all'orchestrazione del flusso di lavoro
- Gestisci le chiavi di accesso IAM utilizzando un AWS SDK
- Gestisci le policy IAM utilizzando un AWS SDK
- Gestisci i ruoli IAM utilizzando un AWS SDK
- Gestisci il tuo account IAM utilizzando un AWS SDK
- Spostamento di segreti codificati per proteggere l'archiviazione segreta
- Permette AWS Compute Optimizer Funzionalità di automazione per applicare le azioni consigliate
- Politica per abilitare l'automazione in tutta l'organizzazione
- Politica per abilitare l'automazione per il tuo account
- Politica per concedere l'accesso completo a Compute Optimizer Automation per un account di gestione di un'organizzazione
- Politica per concedere l'accesso completo a Compute Optimizer Automation per la versione standalone AWS account
- Politica per concedere l'accesso in sola lettura a Compute Optimizer Automation per un account di gestione di un'organizzazione
- Politica per concedere l'accesso in sola lettura a Compute Optimizer Automation per uso autonomo AWS account
- Politica per la concessione di autorizzazioni di ruolo collegate ai servizi per Compute Optimization Automation
- Ripristina una versione della policy IAM utilizzando un AWS SDK
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

- Configurare il controllo Attribute-Based degli accessi per DynamoDB utilizzando AWS Command Line Interface v2
- Configurazione della gestione dei sistemi
- Utilizzo delle variabili di proprietà nei dashboard di monitoraggio per monitorare più funzioni serverless
- Lavora con DynamoDB Streams e utilizza Time-to-Live AWS Command Line Interface v2
- Lavora con l'API IAM Policy Builder utilizzando un AWS SDK

Esempi di base per l'utilizzo di IAM AWS SDK

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di AWS Identity and Access Management with SDK. AWS

Esempi

- Hello IAM
- Scopri le basi di IAM con un AWS SDK
- Azioni per l'utilizzo di IAM AWS SDK
  - Utilizzo di AddClientIdToOpenIdConnectProvider con una CLI
  - Utilizzo di AddRoleToInstanceProfile con una CLI
  - Utilizzo di AddUserToGroup con una CLI
  - Utilizzo di AttachGroupPolicy con una CLI
  - Usare AttachRolePolicycon un AWS SDK o CLI
  - Usare AttachUserPolicycon un AWS SDK o CLI
  - Utilizzo di ChangePassword con una CLI
  - Utilizzare CreateAccessKeycon un AWS SDK o CLI
  - Usare CreateAccountAliascon un AWS SDK o CLI
  - Usare CreateGroupcon un AWS SDK o CLI
  - Usare CreateInstanceProfilecon un AWS SDK o CLI
  - Utilizzo di CreateLoginProfile con una CLI
  - Utilizzo di CreateOpenIdConnectProvider con una CLI
  - Utilizzare CreatePolicycon un AWS SDK o CLI
  - Usare CreatePolicyVersioncon un AWS SDK o CLI

- Usare CreateRolecon un AWS SDK o CLI
- Usa CreateSAMLProvider con un AWS SDK o CLI
- Utilizzare CreateServiceLinkedRolecon un AWS SDK o CLI
- Utilizzare CreateUsercon un AWS SDK o CLI
- Utilizzo di CreateVirtualMfaDevice con una CLI
- Utilizzo di DeactivateMfaDevice con una CLI
- Usare DeleteAccessKeycon un AWS SDK o CLI
- Usare DeleteAccountAliascon un AWS SDK o CLI
- Utilizzo di DeleteAccountPasswordPolicy con una CLI
- Usare DeleteGroupcon un AWS SDK o CLI
- Utilizzo di DeleteGroupPolicy con una CLI
- Usare DeleteInstanceProfilecon un AWS SDK o CLI
- Utilizzo di DeleteLoginProfile con una CLI
- Utilizzo di DeleteOpenIdConnectProvider con una CLI
- Usare DeletePolicycon un AWS SDK o CLI
- Usare DeletePolicyVersioncon un AWS SDK o CLI
- Usare DeleteRolecon un AWS SDK o CLI
- Utilizzo di DeleteRolePermissionsBoundary con una CLI
- Usare DeleteRolePolicycon un AWS SDK o CLI
- Usa DeleteSamlProvider con un AWS SDK o CLI
- Usare DeleteServerCertificatecon un AWS SDK o CLI
- Usare DeleteServiceLinkedRolecon un AWS SDK o CLI
- Utilizzo di DeleteSigningCertificate con una CLI
- Utilizzare DeleteUsercon un AWS SDK o CLI
- Utilizzo di DeleteUserPermissionsBoundary con una CLI
- Usare DeleteUserPolicycon un AWS SDK o CLI
- Utilizzo di DeleteVirtualMfaDevice con una CLI
- Utilizzo di DetachGroupPolicy con una CLI
- Utilizzare DetachRolePolicycon un AWS SDK o CLI
- Usare DetachUserPolicycon un AWS SDK o CLI

- Utilizzo di EnableMfaDevice con una CLI
- Usare GenerateCredentialReportcon un AWS SDK o CLI
- Utilizzo di GenerateServiceLastAccessedDetails con una CLI
- Usare GetAccessKeyLastUsedcon un AWS SDK o CLI
- Utilizzare GetAccountAuthorizationDetailscon un AWS SDK o CLI
- Usare GetAccountPasswordPolicycon un AWS SDK o CLI
- Utilizzare GetAccountSummarycon un AWS SDK o CLI
- Utilizzo di GetContextKeysForCustomPolicy con una CLI
- Utilizzo di GetContextKeysForPrincipalPolicy con una CLI
- Usare GetCredentialReportcon un AWS SDK o CLI
- Utilizzo di GetGroup con una CLI
- Utilizzo di GetGroupPolicy con una CLI
- Utilizzo di GetInstanceProfile con una CLI
- Utilizzo di GetLoginProfile con una CLI
- Utilizzo di GetOpenIdConnectProvider con una CLI
- Usare GetPolicycon un AWS SDK o CLI
- Usare GetPolicyVersioncon un AWS SDK o CLI
- Utilizzare GetRolecon un AWS SDK o CLI
- Utilizzo di GetRolePolicy con una CLI
- Utilizzo di GetSamlProvider con una CLI
- Usare GetServerCertificatecon un AWS SDK o CLI
- Utilizzo di GetServiceLastAccessedDetails con una CLI
- Utilizzo di GetServiceLastAccessedDetailsWithEntities con una CLI
- Utilizzare GetServiceLinkedRoleDeletionStatuscon un AWS SDK o CLI
- Usare GetUsercon un AWS SDK o CLI
- Utilizzo di GetUserPolicy con una CLI
- Usare ListAccessKeyscon un AWS SDK o CLI
- Utilizzare ListAccountAliasescon un AWS SDK o CLI
- Utilizzo di ListAttachedGroupPolicies con una CLI
- Utilizzare ListAttachedRolePoliciescon un AWS SDK o CLI

- Utilizzo di ListAttachedUserPolicies con una CLI
- Utilizzo di ListEntitiesForPolicy con una CLI
- Utilizzo di ListGroupPolicies con una CLI
- Utilizzare ListGroupscon un AWS SDK o CLI
- Utilizzo di ListGroupsForUser con una CLI
- Utilizzo di ListInstanceProfiles con una CLI
- Utilizzo di ListInstanceProfilesForRole con una CLI
- Utilizzo di ListMfaDevices con una CLI
- Utilizzo di ListOpenIdConnectProviders con una CLI
- Utilizzare ListPoliciescon un AWS SDK o CLI
- Usare ListPolicyVersionscon un AWS SDK o CLI
- Utilizzare ListRolePoliciescon un AWS SDK o CLI
- Utilizzo di ListRoleTags con una CLI
- Utilizzare ListRolescon un AWS SDK o CLI
- Usa ListSamlProviders con un AWS SDK o CLI
- Utilizzare ListServerCertificatescon un AWS SDK o CLI
- Utilizzo di ListSigningCertificates con una CLI
- Usare ListUserPoliciescon un AWS SDK o CLI
- Utilizzo di ListUserTags con una CLI
- Usare ListUserscon un AWS SDK o CLI
- Utilizzo di ListVirtualMfaDevices con una CLI
- Utilizzo di PutGroupPolicy con una CLI
- Utilizzo di PutRolePermissionsBoundary con una CLI
- Utilizzare PutRolePolicycon un AWS SDK o CLI
- Utilizzo di PutUserPermissionsBoundary con una CLI
- Usare PutUserPolicycon un AWS SDK o CLI
- Utilizzo di RemoveClientIdFromOpenIdConnectProvider con una CLI
- Utilizzo di RemoveRoleFromInstanceProfile con una CLI
- Utilizzo di RemoveUserFromGroup con una CLI

- Utilizzo di ResyncMfaDevice con una CLI

- Usare SetDefaultPolicyVersioncon un AWS SDK o CLI
- Utilizzo di TagRole con una CLI
- Utilizzo di TagUser con una CLI
- Utilizzo di UntagRole con una CLI
- Utilizzo di UntagUser con una CLI
- Utilizzare UpdateAccessKeycon un AWS SDK o CLI
- Utilizzo di UpdateAccountPasswordPolicy con una CLI
- Utilizzo di UpdateAssumeRolePolicy con una CLI
- Utilizzo di UpdateGroup con una CLI
- Utilizzo di UpdateLoginProfile con una CLI
- Utilizzo di UpdateOpenIdConnectProviderThumbprint con una CLI
- Utilizzo di UpdateRole con una CLI
- Utilizzo di UpdateRoleDescription con una CLI
- Utilizzo di UpdateSamlProvider con una CLI
- Usare UpdateServerCertificatecon un AWS SDK o CLI
- Utilizzo di UpdateSigningCertificate con una CLI
- Usare UpdateUsercon un AWS SDK o CLI
- Utilizzare UploadServerCertificatecon un AWS SDK o CLI
- Utilizzo di UploadSigningCertificate con una CLI

Hello IAM

Gli esempi di codice seguenti mostrano come iniziare a utilizzare IAM.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
namespace IAMActions;

public class HelloIAM
{
    static async Task Main(string[] args)
    {
        // Getting started with AWS Identity and Access Management (IAM). List
        // the policies for the account.
        var iamClient = new AmazonIdentityManagementServiceClient();

        var listPoliciesPaginator = iamClient.Paginators.ListPolicies(new
 ListPoliciesRequest());
        var policies = new List<ManagedPolicy>();

        await foreach (var response in listPoliciesPaginator.Responses)
        {
            policies.AddRange(response.Policies);
        }

        Console.WriteLine("Here are the policies defined for your account:\n");
        policies.ForEach(policy =>
        {
            Console.WriteLine($"Created:
 {policy.CreateDate}\t{policy.PolicyName}\t{policy.Description}");
        });
    }
}
```

- Per i dettagli sull'API, consulta la ListPoliciessezione AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Codice per il file CMakeLists.txt CMake.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS iam)

# Set this project's name.
project("hello_iam")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
    # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.

    # set(BIN_SUB_DIR "/Debug") # if you are building from the command line you
 may need to uncomment this
    # and set the proper subdirectory to the executables' location.

    AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_iam.cpp)

target_link_libraries(${PROJECT_NAME}
```

```
        ${AWSSDK_LINK_LIBRARIES})
```

Codice per il file origine iam.cpp.

```
#include <aws/core/Aws.h>
#include <aws/iam/IAMClient.h>
#include <aws/iam/model/ListPoliciesRequest.h>
#include <iostream>
#include <iomanip>

/*
 *  A "Hello IAM" starter application which initializes an AWS Identity and
 Access Management (IAM) client
 *  and lists the IAM policies.
 *
 *  main function
 *
 *  Usage: 'hello_iam'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        const Aws::String DATE_FORMAT("%Y-%m-%d");
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::IAM::IAMClient iamClient(clientConfig);
        Aws::IAM::Model::ListPoliciesRequest request;

        bool done = false;
        bool header = false;
        while (!done) {
            auto outcome = iamClient.ListPolicies(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Failed to list iam policies: " <<
```

```
                        outcome.GetError().GetMessage() << std::endl;
                result = 1;
                break;
            }

            if (!header) {
                std::cout << std::left << std::setw(55) << "Name" <<
                          std::setw(30) << "ID" << std::setw(80) << "Arn" <<
                          std::setw(64) << "Description" << std::setw(12) <<
                          "CreateDate" << std::endl;
                header = true;
            }

            const auto &policies = outcome.GetResult().GetPolicies();
            for (const auto &policy: policies) {
                std::cout << std::left << std::setw(55) <<
                          policy.GetPolicyName() << std::setw(30) <<
                          policy.GetPolicyId() << std::setw(80) <<
 policy.GetArn() <<
                          std::setw(64) << policy.GetDescription() <<
 std::setw(12) <<
                          policy.GetCreateDate().ToGmtString(DATE_FORMAT.c_str())
 <<
                          std::endl;
            }

            if (outcome.GetResult().GetIsTruncated()) {
                request.SetMarker(outcome.GetResult().GetMarker());
            } else {
                done = true;
            }
        }
    }


    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Per i dettagli sull'API, vedi ListPoliciesin AWS SDK per C++ API Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
package main

import (
 "context"
 "fmt"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/iam"
)

// main uses the AWS SDK for Go (v2) to create an AWS Identity and Access
 Management (IAM)
// client and list up to 10 policies in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 iamClient := iam.NewFromConfig(sdkConfig)
 const maxPols = 10
 fmt.Printf("Let's list up to %v policies for your account.\n", maxPols)
 result, err := iamClient.ListPolicies(ctx, &iam.ListPoliciesInput{
  MaxItems: aws.Int32(maxPols),
 })
```

```
 if err != nil {
  fmt.Printf("Couldn't list policies for your account. Here's why: %v\n", err)
  return
 }
 if len(result.Policies) == 0 {
  fmt.Println("You don't have any policies!")
 } else {
  for _, policy := range result.Policies {
   fmt.Printf("\t%v\n", *policy.PolicyName)
  }
 }
}
```

- Per i dettagli sull'API, consulta la ListPoliciessezione AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.ListPoliciesResponse;
import software.amazon.awssdk.services.iam.model.Policy;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class HelloIAM {
    public static void main(String[] args) {
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        listPolicies(iam);
    }

    public static void listPolicies(IamClient iam) {
        ListPoliciesResponse response = iam.listPolicies();
        List<Policy> polList = response.policies();
        polList.forEach(policy -> {
            System.out.println("Policy Name: " + policy.policyName());
        });
    }
}
```

- Per i dettagli sull'API, consulta la ListPoliciessezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { IAMClient, paginateListPolicies } from "@aws-sdk/client-iam";

const client = new IAMClient({});

export const listLocalPolicies = async () => {
  /**
   * In v3, the clients expose paginateOperationName APIs that are written using
 async generators so that you can use async iterators in a for await..of loop.
   * https://docs.aws.amazon.com/AWSJavaScriptSDK/v3/latest/index.html#paginators
   */
```

```
  const paginator = paginateListPolicies(
    { client, pageSize: 10 },
    // List only customer managed policies.
    { Scope: "Local" },
  );

  console.log("IAM policies defined in your account:");
  let policyCount = 0;
  for await (const page of paginator) {
    if (page.Policies) {
      for (const policy of page.Policies) {
        console.log(`${policy.PolicyName}`);
        policyCount++;
      }
    }
  }
  console.log(`Found ${policyCount} policies.`);
};
```

- Per i dettagli sull'API, consulta la ListPoliciessezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import boto3


def main():
    """
    Lists the managed policies in your AWS account using the AWS SDK for Python
 (Boto3).
```

```
    """
    iam = boto3.client("iam")

    try:
        # Get a paginator for the list_policies operation
        paginator = iam.get_paginator("list_policies")

        # Iterate through the pages of results
        for page in paginator.paginate(Scope="All", OnlyAttached=False):
            for policy in page["Policies"]:
                print(f"Policy name: {policy['PolicyName']}")
                print(f"  Policy ARN: {policy['Arn']}")
    except boto3.exceptions.BotoCoreError as e:
        print(f"Encountered an error while listing policies: {e}")


if __name__ == "__main__":
    main()
```

- Per i dettagli sull'API, consulta ListPolicies AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
require 'aws-sdk-iam'
require 'logger'

# IAMManager is a class responsible for managing IAM operations
# such as listing all IAM policies in the current AWS account.
class IAMManager
  def initialize(client)
    @client = client
```

```
    @logger = Logger.new($stdout)
  end

  # Lists and prints all IAM policies in the current AWS account.
  def list_policies
    @logger.info('Here are the IAM policies in your account:')

    paginator = @client.list_policies
    policies = []

    paginator.each_page do |page|
      policies.concat(page.policies)
    end

    if policies.empty?
      @logger.info("You don't have any IAM policies.")
    else
      policies.each do |policy|
        @logger.info("- #{policy.policy_name}")
      end
    end
  end
end

if $PROGRAM_NAME == __FILE__
  iam_client = Aws::IAM::Client.new
  manager = IAMManager.new(iam_client)
  manager.list_policies
end
```

- Per i dettagli sull'API, consulta la ListPoliciessezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Da src/bin /hello.rs.

```
use aws_sdk_iam::error::SdkError;
use aws_sdk_iam::operation::list_policies::ListPoliciesError;
use clap::Parser;

const PATH_PREFIX_HELP: &str = "The path prefix for filtering the results.";

#[derive(Debug, clap::Parser)]
#[command(about)]
struct HelloScenarioArgs {
    #[arg(long, default_value="/", help=PATH_PREFIX_HELP)]
    pub path_prefix: String,
}

#[tokio::main]
async fn main() -> Result<(), Box<SdkError<ListPoliciesError>>> {
    let sdk_config = aws_config::load_from_env().await;
    let client = aws_sdk_iam::Client::new(&sdk_config);

    let args = HelloScenarioArgs::parse();

    iam_service::list_policies(client, args.path_prefix).await?;

    Ok(())
}
```

Da src/iam -service-lib.rs.

```
pub async fn list_policies(
    client: iamClient,
    path_prefix: String,
) -> Result<Vec<String>, SdkError<ListPoliciesError>> {
    let list_policies = client
        .list_policies()
        .path_prefix(path_prefix)
        .scope(PolicyScopeType::Local)
        .into_paginator()
        .items()
        .send()
        .try_collect()
```

```
        .await?;

    let policy_names = list_policies
        .into_iter()
        .map(|p| {
            let name = p
                .policy_name
                .unwrap_or_else(|| "Missing Policy Name".to_string());
            println!("{}", name);
            name
        })
        .collect();

    Ok(policy_names)
}
```

- Per i dettagli sulle API, consulta ListPolicieslla guida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scopri le basi di IAM con un AWS SDK

Gli esempi di codice seguenti mostrano come creare un utente e assumere un ruolo.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come Centro identità AWS IAM.

- Crea un utente che non disponga di autorizzazioni.
- Crea un ruolo che conceda l'autorizzazione per elencare i bucket Amazon S3 per l'account.
- Aggiungi una policy per consentire all'utente di assumere il ruolo.
- Assumi il ruolo ed elenca i bucket S3 utilizzando le credenziali temporanee, quindi ripulisci le risorse.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
global using Amazon.IdentityManagement;
global using Amazon.S3;
global using Amazon.SecurityToken;
global using IAMActions;
global using IamScenariosCommon;
global using Microsoft.Extensions.DependencyInjection;
global using Microsoft.Extensions.Hosting;
global using Microsoft.Extensions.Logging;
global using Microsoft.Extensions.Logging.Console;
global using Microsoft.Extensions.Logging.Debug;


namespace IAMActions;

public class IAMWrapper
{
    private readonly IAmazonIdentityManagementService _IAMService;

    /// <summary>
    /// Constructor for the IAMWrapper class.
    /// </summary>
    /// <param name="IAMService">An IAM client object.</param>
    public IAMWrapper(IAmazonIdentityManagementService IAMService)
    {
        _IAMService = IAMService;
    }

    /// <summary>
    /// Attach an IAM policy to a role.
    /// </summary>
    /// <param name="policyArn">The policy to attach.</param>
```

```
    /// <param name="roleName">The role that the policy will be attached to.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AttachRolePolicyAsync(string policyArn, string
 roleName)
    {
        var response = await _IAMService.AttachRolePolicyAsync(new
 AttachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Create an IAM access key for a user.
    /// </summary>
    /// <param name="userName">The username for which to create the IAM access
    /// key.</param>
    /// <returns>The AccessKey.</returns>
    public async Task<AccessKey> CreateAccessKeyAsync(string userName)
    {
        var response = await _IAMService.CreateAccessKeyAsync(new
 CreateAccessKeyRequest
        {
            UserName = userName,
        });

        return response.AccessKey;

    }


    /// <summary>
    /// Create an IAM policy.
    /// </summary>
    /// <param name="policyName">The name to give the new IAM policy.</param>
    /// <param name="policyDocument">The policy document for the new policy.</
param>
    /// <returns>The new IAM policy object.</returns>
```

```
    public async Task<ManagedPolicy> CreatePolicyAsync(string policyName, string
 policyDocument)
    {
        var response = await _IAMService.CreatePolicyAsync(new
 CreatePolicyRequest
        {
            PolicyDocument = policyDocument,
            PolicyName = policyName,
        });

        return response.Policy;
    }


    /// <summary>
    /// Create a new IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="rolePolicyDocument">The name of the IAM policy document
    /// for the new role.</param>
    /// <returns>The Amazon Resource Name (ARN) of the role.</returns>
    public async Task<string> CreateRoleAsync(string roleName, string
 rolePolicyDocument)
    {
        var request = new CreateRoleRequest
        {
            RoleName = roleName,
            AssumeRolePolicyDocument = rolePolicyDocument,
        };

        var response = await _IAMService.CreateRoleAsync(request);
        return response.Role.Arn;
    }


    /// <summary>
    /// Create an IAM service-linked role.
    /// </summary>
    /// <param name="serviceName">The name of the AWS Service.</param>
    /// <param name="description">A description of the IAM service-linked role.</
param>
    /// <returns>The IAM role that was created.</returns>
    public async Task<Role> CreateServiceLinkedRoleAsync(string serviceName,
 string description)
```

```
    {
        var request = new CreateServiceLinkedRoleRequest
        {
            AWSServiceName = serviceName,
            Description = description
        };

        var response = await _IAMService.CreateServiceLinkedRoleAsync(request);
        return response.Role;
    }


    /// <summary>
    /// Create an IAM user.
    /// </summary>
    /// <param name="userName">The username for the new IAM user.</param>
    /// <returns>The IAM user that was created.</returns>
    public async Task<User> CreateUserAsync(string userName)
    {
        var response = await _IAMService.CreateUserAsync(new CreateUserRequest
{ UserName = userName });
        return response.User;
    }


    /// <summary>
    /// Delete an IAM user's access key.
    /// </summary>
    /// <param name="accessKeyId">The Id for the IAM access key.</param>
    /// <param name="userName">The username of the user that owns the IAM
    /// access key.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteAccessKeyAsync(string accessKeyId, string
userName)
    {
        var response = await _IAMService.DeleteAccessKeyAsync(new
DeleteAccessKeyRequest
        {
            AccessKeyId = accessKeyId,
            UserName = userName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

```
    /// <summary>
    /// Delete an IAM policy.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the policy to
    /// delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeletePolicyAsync(string policyArn)
    {
        var response = await _IAMService.DeletePolicyAsync(new
 DeletePolicyRequest { PolicyArn = policyArn });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRoleAsync(string roleName)
    {
        var response = await _IAMService.DeleteRoleAsync(new DeleteRoleRequest
 { RoleName = roleName });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM role policy.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="policyName">The name of the IAM role policy to delete.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRolePolicyAsync(string roleName, string
 policyName)
    {
        var response = await _IAMService.DeleteRolePolicyAsync(new
 DeleteRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
```

```
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM user.
    /// </summary>
    /// <param name="userName">The username of the IAM user to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserAsync(string userName)
    {
        var response = await _IAMService.DeleteUserAsync(new DeleteUserRequest
{ UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM user policy.
    /// </summary>
    /// <param name="policyName">The name of the IAM policy to delete.</param>
    /// <param name="userName">The username of the IAM user.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserPolicyAsync(string policyName, string
userName)
    {
        var response = await _IAMService.DeleteUserPolicyAsync(new
DeleteUserPolicyRequest { PolicyName = policyName, UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Detach an IAM policy from an IAM role.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the IAM
policy.</param>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
```

```
    public async Task<bool> DetachRolePolicyAsync(string policyArn, string
 roleName)
    {
        var response = await _IAMService.DetachRolePolicyAsync(new
 DetachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Gets the IAM password policy for an AWS account.
    /// </summary>
    /// <returns>The PasswordPolicy for the AWS account.</returns>
    public async Task<PasswordPolicy> GetAccountPasswordPolicyAsync()
    {
        var response = await _IAMService.GetAccountPasswordPolicyAsync(new
 GetAccountPasswordPolicyRequest());
        return response.PasswordPolicy;
    }


    /// <summary>
    /// Get information about an IAM policy.
    /// </summary>
    /// <param name="policyArn">The IAM policy to retrieve information for.</
param>
    /// <returns>The IAM policy.</returns>
    public async Task<ManagedPolicy> GetPolicyAsync(string policyArn)
    {

        var response = await _IAMService.GetPolicyAsync(new GetPolicyRequest
 { PolicyArn = policyArn });
        return response.Policy;
    }


    /// <summary>
    /// Get information about an IAM role.
    /// </summary>
```

```
    /// <param name="roleName">The name of the IAM role to retrieve information
    /// for.</param>
    /// <returns>The IAM role that was retrieved.</returns>
    public async Task<Role> GetRoleAsync(string roleName)
    {
        var response = await _IAMService.GetRoleAsync(new GetRoleRequest
        {
            RoleName = roleName,
        });

        return response.Role;
    }


    /// <summary>
    /// Get information about an IAM user.
    /// </summary>
    /// <param name="userName">The username of the user.</param>
    /// <returns>An IAM user object.</returns>
    public async Task<User> GetUserAsync(string userName)
    {
        var response = await _IAMService.GetUserAsync(new GetUserRequest
{ UserName = userName });
        return response.User;
    }


    /// <summary>
    /// List the IAM role policies that are attached to an IAM role.
    /// </summary>
    /// <param name="roleName">The IAM role to list IAM policies for.</param>
    /// <returns>A list of the IAM policies attached to the IAM role.</returns>
    public async Task<List<AttachedPolicyType>>
ListAttachedRolePoliciesAsync(string roleName)
    {
        var attachedPolicies = new List<AttachedPolicyType>();
        var attachedRolePoliciesPaginator =
_IAMService.Paginators.ListAttachedRolePolicies(new
ListAttachedRolePoliciesRequest { RoleName = roleName });

        await foreach (var response in attachedRolePoliciesPaginator.Responses)
        {
            attachedPolicies.AddRange(response.AttachedPolicies);
        }
```

```
        return attachedPolicies;
    }


    /// <summary>
    /// List IAM groups.
    /// </summary>
    /// <returns>A list of IAM groups.</returns>
    public async Task<List<Group>> ListGroupsAsync()
    {
        var groupsPaginator = _IAMService.Paginators.ListGroups(new
ListGroupsRequest());
        var groups = new List<Group>();

        await foreach (var response in groupsPaginator.Responses)
        {
            groups.AddRange(response.Groups);
        }

        return groups;
    }


    /// <summary>
    /// List IAM policies.
    /// </summary>
    /// <returns>A list of the IAM policies.</returns>
    public async Task<List<ManagedPolicy>> ListPoliciesAsync()
    {
        var listPoliciesPaginator = _IAMService.Paginators.ListPolicies(new
ListPoliciesRequest());
        var policies = new List<ManagedPolicy>();

        await foreach (var response in listPoliciesPaginator.Responses)
        {
            policies.AddRange(response.Policies);
        }

        return policies;
    }


    /// <summary>
```

```
    /// List IAM role policies.
    /// </summary>
    /// <param name="roleName">The IAM role for which to list IAM policies.</
param>
    /// <returns>A list of IAM policy names.</returns>
    public async Task<List<string>> ListRolePoliciesAsync(string roleName)
    {
        var listRolePoliciesPaginator =
 _IAMService.Paginators.ListRolePolicies(new ListRolePoliciesRequest { RoleName =
 roleName });
        var policyNames = new List<string>();

        await foreach (var response in listRolePoliciesPaginator.Responses)
        {
            policyNames.AddRange(response.PolicyNames);
        }

        return policyNames;
    }


    /// <summary>
    /// List IAM roles.
    /// </summary>
    /// <returns>A list of IAM roles.</returns>
    public async Task<List<Role>> ListRolesAsync()
    {
        var listRolesPaginator = _IAMService.Paginators.ListRoles(new
 ListRolesRequest());
        var roles = new List<Role>();

        await foreach (var response in listRolesPaginator.Responses)
        {
            roles.AddRange(response.Roles);
        }

        return roles;
    }


    /// <summary>
    /// List SAML authentication providers.
    /// </summary>
    /// <returns>A list of SAML providers.</returns>
```

```
    public async Task<List<SAMLProviderListEntry>> ListSAMLProvidersAsync()
    {
        var response = await _IAMService.ListSAMLProvidersAsync(new
 ListSAMLProvidersRequest());
        return response.SAMLProviderList;
    }


    /// <summary>
    /// List IAM users.
    /// </summary>
    /// <returns>A list of IAM users.</returns>
    public async Task<List<User>> ListUsersAsync()
    {
        var listUsersPaginator = _IAMService.Paginators.ListUsers(new
 ListUsersRequest());
        var users = new List<User>();

        await foreach (var response in listUsersPaginator.Responses)
        {
            users.AddRange(response.Users);
        }

        return users;
    }


    /// <summary>
    /// Update the inline policy document embedded in a role.
    /// </summary>
    /// <param name="policyName">The name of the policy to embed.</param>
    /// <param name="roleName">The name of the role to update.</param>
    /// <param name="policyDocument">The policy document that defines the role.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutRolePolicyAsync(string policyName, string
 roleName, string policyDocument)
    {
        var request = new PutRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
            PolicyDocument = policyDocument
        };
```

```
        var response = await _IAMService.PutRolePolicyAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Add or update an inline policy document that is embedded in an IAM user.
    /// </summary>
    /// <param name="userName">The name of the IAM user.</param>
    /// <param name="policyName">The name of the IAM policy.</param>
    /// <param name="policyDocument">The policy document defining the IAM
policy.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutUserPolicyAsync(string userName, string
policyName, string policyDocument)
    {
        var request = new PutUserPolicyRequest
        {
            UserName = userName,
            PolicyName = policyName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutUserPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }

    /// <summary>
    /// Wait for a new access key to be ready to use.
    /// </summary>
    /// <param name="accessKeyId">The Id of the access key.</param>
    /// <returns>A boolean value indicating the success of the action.</returns>
    public async Task<bool> WaitUntilAccessKeyIsReady(string accessKeyId)
    {
        var keyReady = false;

        do
        {
            try
            {
                var response = await _IAMService.GetAccessKeyLastUsedAsync(
                    new GetAccessKeyLastUsedRequest { AccessKeyId =
accessKeyId });
```

```
                if (response.UserName is not null)
                {
                    keyReady = true;
                }
            }
            catch (NoSuchEntityException)
            {
                keyReady = false;
            }
        } while (!keyReady);

        return keyReady;
    }
}
```

```
using Microsoft.Extensions.Configuration;

namespace IAMBasics;

public class IAMBasics
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for the AWS service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonIdentityManagementService>()
            .AddTransient<IAMWrapper>()
            .AddTransient<UIWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<IAMBasics>();
```

```
        IConfiguration configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // Values needed for user, role, and policies.
        string userName = configuration["UserName"]!;
        string s3PolicyName = configuration["S3PolicyName"]!;
        string roleName = configuration["RoleName"]!;


        var iamWrapper = host.Services.GetRequiredService<IAMWrapper>();
        var uiWrapper = host.Services.GetRequiredService<UIWrapper>();

        uiWrapper.DisplayBasicsOverview();
        uiWrapper.PressEnter();

        // First create a user. By default, the new user has
        // no permissions.
        uiWrapper.DisplayTitle("Create User");
        Console.WriteLine($"Creating a new user with user name: {userName}.");
        var user = await iamWrapper.CreateUserAsync(userName);
        var userArn = user.Arn;

        Console.WriteLine($"Successfully created user: {userName} with ARN:
{userArn}.");
        uiWrapper.WaitABit(15, "Now let's wait for the user to be ready for
use.");

        // Define a role policy document that allows the new user
        // to assume the role.
        string assumeRolePolicyDocument = "{" +
          "\"Version\": \"2012-10-17\"," +
          "\"Statement\": [{" +
              "\"Effect\": \"Allow\"," +
              "\"Principal\": {" +
              $" \"AWS\": \"{userArn}\"" +
              "}," +
              "\"Action\": \"sts:AssumeRole\"" +
          "}]" +
```

```
        "}";

        // Permissions to list all buckets.
        string policyDocument = "{" +
            "\"Version\": \"2012-10-17\"," +
            " \"Statement\" : [{" +
                " \"Action\" : [\"s3:ListAllMyBuckets\"]," +
                " \"Effect\" : \"Allow\"," +
                " \"Resource\" : \"*\"" +
            "}]" +
        "}";

        // Create an AccessKey for the user.
        uiWrapper.DisplayTitle("Create access key");
        Console.WriteLine("Now let's create an access key for the new user.");
        var accessKey = await iamWrapper.CreateAccessKeyAsync(userName);

        var accessKeyId = accessKey.AccessKeyId;
        var secretAccessKey = accessKey.SecretAccessKey;

        Console.WriteLine($"We have created the access key with Access key id:
{accessKeyId}.");

        Console.WriteLine("Now let's wait until the IAM access key is ready to
use.");
        var keyReady = await iamWrapper.WaitUntilAccessKeyIsReady(accessKeyId);

        // Now try listing the Amazon Simple Storage Service (Amazon S3)
        // buckets. This should fail at this point because the user doesn't
        // have permissions to perform this task.
        uiWrapper.DisplayTitle("Try to display Amazon S3 buckets");
        Console.WriteLine("Now let's try to display a list of the user's Amazon
S3 buckets.");
        var s3Client1 = new AmazonS3Client(accessKeyId, secretAccessKey);
        var stsClient1 = new AmazonSecurityTokenServiceClient(accessKeyId,
secretAccessKey);

        var s3Wrapper = new S3Wrapper(s3Client1, stsClient1);
        var buckets = await s3Wrapper.ListMyBucketsAsync();

        Console.WriteLine(buckets is null
            ? "As expected, the call to list the buckets has returned a null
list."
            : "Something went wrong. This shouldn't have worked.");
```

```
        uiWrapper.PressEnter();

        uiWrapper.DisplayTitle("Create IAM role");
        Console.WriteLine($"Creating the role: {roleName}");

        // Creating an IAM role to allow listing the S3 buckets. A role name
        // is not case sensitive and must be unique to the account for which it
        // is created.
        var roleArn = await iamWrapper.CreateRoleAsync(roleName,
assumeRolePolicyDocument);

        uiWrapper.PressEnter();

        // Create a policy with permissions to list S3 buckets.
        uiWrapper.DisplayTitle("Create IAM policy");
        Console.WriteLine($"Creating the policy: {s3PolicyName}");
        Console.WriteLine("with permissions to list the Amazon S3 buckets for the
account.");
        var policy = await iamWrapper.CreatePolicyAsync(s3PolicyName,
policyDocument);

        // Wait 15 seconds for the IAM policy to be available.
        uiWrapper.WaitABit(15, "Waiting for the policy to be available.");

        // Attach the policy to the role you created earlier.
        uiWrapper.DisplayTitle("Attach new IAM policy");
        Console.WriteLine("Now let's attach the policy to the role.");
        await iamWrapper.AttachRolePolicyAsync(policy.Arn, roleName);

        // Wait 15 seconds for the role to be updated.
        Console.WriteLine();
        uiWrapper.WaitABit(15, "Waiting for the policy to be attached.");

        // Use the AWS Security Token Service (AWS STS) to have the user
        // assume the role we created.
        var stsClient2 = new AmazonSecurityTokenServiceClient(accessKeyId,
secretAccessKey);

        // Wait for the new credentials to become valid.
        uiWrapper.WaitABit(10, "Waiting for the credentials to be valid.");

        var assumedRoleCredentials = await
s3Wrapper.AssumeS3RoleAsync("temporary-session", roleArn);
```

```
        // Try again to list the buckets using the client created with
        // the new user's credentials. This time, it should work.
        var s3Client2 = new AmazonS3Client(assumedRoleCredentials);

        s3Wrapper.UpdateClients(s3Client2, stsClient2);

        buckets = await s3Wrapper.ListMyBucketsAsync();

        uiWrapper.DisplayTitle("List Amazon S3 buckets");
        Console.WriteLine("This time we should have buckets to list.");
        if (buckets is not null)
        {
            buckets.ForEach(bucket =>
            {
                Console.WriteLine($"{bucket.BucketName} created:
{bucket.CreationDate}");
            });
        }

        uiWrapper.PressEnter();

        // Now clean up all the resources used in the example.
        uiWrapper.DisplayTitle("Clean up resources");
        Console.WriteLine("Thank you for watching. The IAM Basics demo is
complete.");
        Console.WriteLine("Please wait while we clean up the resources we
created.");

        await iamWrapper.DetachRolePolicyAsync(policy.Arn, roleName);

        await iamWrapper.DeletePolicyAsync(policy.Arn);

        await iamWrapper.DeleteRoleAsync(roleName);

        await iamWrapper.DeleteAccessKeyAsync(accessKeyId, userName);

        await iamWrapper.DeleteUserAsync(userName);

        uiWrapper.PressEnter();

        Console.WriteLine("All done cleaning up our resources. Thank you for your
patience.");
    }
```

```
}


namespace IamScenariosCommon;

using System.Net;

/// <summary>
/// A class to perform Amazon Simple Storage Service (Amazon S3) actions for
/// the IAM Basics scenario.
/// </summary>
public class S3Wrapper
{
    private IAmazonS3 _s3Service;
    private IAmazonSecurityTokenService _stsService;

    /// <summary>
    /// Constructor for the S3Wrapper class.
    /// </summary>
    /// <param name="s3Service">An Amazon S3 client object.</param>
    /// <param name="stsService">An AWS Security Token Service (AWS STS)
    /// client object.</param>
    public S3Wrapper(IAmazonS3 s3Service, IAmazonSecurityTokenService stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }

    /// <summary>
    /// Assumes an AWS Identity and Access Management (IAM) role that allows
    /// Amazon S3 access for the current session.
    /// </summary>
    /// <param name="roleSession">A string representing the current session.</
param>
    /// <param name="roleToAssume">The name of the IAM role to assume.</param>
    /// <returns>Credentials for the newly assumed IAM role.</returns>
    public async Task<Credentials> AssumeS3RoleAsync(string roleSession, string
 roleToAssume)
    {
        // Create the request to use with the AssumeRoleAsync call.
        var request = new AssumeRoleRequest()
        {
            RoleSessionName = roleSession,
            RoleArn = roleToAssume,
```

```
        };

        var response = await _stsService.AssumeRoleAsync(request);

        return response.Credentials;
    }


    /// <summary>
    /// Delete an S3 bucket.
    /// </summary>
    /// <param name="bucketName">Name of the S3 bucket to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteBucketAsync(string bucketName)
    {
        var result = await _s3Service.DeleteBucketAsync(new DeleteBucketRequest
{ BucketName = bucketName });
        return result.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// List the buckets that are owned by the user's account.
    /// </summary>
    /// <returns>Async Task.</returns>
    public async Task<List<S3Bucket>?> ListMyBucketsAsync()
    {
        try
        {
            // Get the list of buckets accessible by the new user.
            var response = await _s3Service.ListBucketsAsync();

            return response.Buckets;
        }
        catch (AmazonS3Exception ex)
        {
            // Something else went wrong. Display the error message.
            Console.WriteLine($"Error: {ex.Message}");
            return null;
        }
    }

    /// <summary>
    /// Create a new S3 bucket.
    /// </summary>
```

```
    /// <param name="bucketName">The name for the new bucket.</param>
    /// <returns>A Boolean value indicating whether the action completed
    /// successfully.</returns>
    public async Task<bool> PutBucketAsync(string bucketName)
    {
        var response = await _s3Service.PutBucketAsync(new PutBucketRequest
 { BucketName = bucketName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Update the client objects with new client objects. This is available
    /// because the scenario uses the methods of this class without and then
    /// with the proper permissions to list S3 buckets.
    /// </summary>
    /// <param name="s3Service">The Amazon S3 client object.</param>
    /// <param name="stsService">The AWS STS client object.</param>
    public void UpdateClients(IAmazonS3 s3Service, IAmazonSecurityTokenService
 stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }
}


namespace IamScenariosCommon;

public class UIWrapper
{
    public readonly string SepBar = new('-', Console.WindowWidth);

    /// <summary>
    /// Show information about the IAM Groups scenario.
    /// </summary>
    public void DisplayGroupsOverview()
    {
        Console.Clear();

        DisplayTitle("Welcome to the IAM Groups Demo");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates an Amazon Identity and Access Management
 (IAM) group.");
```

```
        Console.WriteLine("\t2. Adds an IAM policy to the IAM group giving it
full access to Amazon S3.");
        Console.WriteLine("\t3. Creates a new IAM user.");
        Console.WriteLine("\t4. Creates an IAM access key for the user.");
        Console.WriteLine("\t5. Adds the user to the IAM group.");
        Console.WriteLine("\t6. Lists the buckets on the account.");
        Console.WriteLine("\t7. Proves that the user has full Amazon S3 access by
creating a bucket.");
        Console.WriteLine("\t8. List the buckets again to show the new bucket.");
        Console.WriteLine("\t9. Cleans up all the resources created.");
    }

    /// <summary>
    /// Show information about the IAM Basics scenario.
    /// </summary>
    public void DisplayBasicsOverview()
    {
        Console.Clear();

        DisplayTitle("Welcome to IAM Basics");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates a user with no permissions.");
        Console.WriteLine("\t2. Creates a role and policy that grant
s3:ListAllMyBuckets permission.");
        Console.WriteLine("\t3. Grants the user permission to assume the role.");
        Console.WriteLine("\t4. Creates an S3 client object as the user and tries
to list buckets (this will fail).");
        Console.WriteLine("\t5. Gets temporary credentials by assuming the
role.");
        Console.WriteLine("\t6. Creates a new S3 client object with the temporary
credentials and lists the buckets (this will succeed).");
        Console.WriteLine("\t7. Deletes all the resources.");
    }

    /// <summary>
    /// Display a message and wait until the user presses enter.
    /// </summary>
    public void PressEnter()
    {
        Console.Write("\nPress <Enter> to continue. ");
        _ = Console.ReadLine();
        Console.WriteLine();
    }
```

```
    /// <summary>
    /// Pad a string with spaces to center it on the console display.
    /// </summary>
    /// <param name="strToCenter">The string to be centered.</param>
    /// <returns>The padded string.</returns>
    public string CenterString(string strToCenter)
    {
        var padAmount = (Console.WindowWidth - strToCenter.Length) / 2;
        var leftPad = new string(' ', padAmount);
        return $"{leftPad}{strToCenter}";
    }

    /// <summary>
    /// Display a line of hyphens, the centered text of the title, and another
    /// line of hyphens.
    /// </summary>
    /// <param name="strTitle">The string to be displayed.</param>
    public void DisplayTitle(string strTitle)
    {
        Console.WriteLine(SepBar);
        Console.WriteLine(CenterString(strTitle));
        Console.WriteLine(SepBar);
    }

    /// <summary>
    /// Display a countdown and wait for a number of seconds.
    /// </summary>
    /// <param name="numSeconds">The number of seconds to wait.</param>
    public void WaitABit(int numSeconds, string msg)
    {
        Console.WriteLine(msg);

        // Wait for the requested number of seconds.
        for (int i = numSeconds; i > 0; i--)
        {
            System.Threading.Thread.Sleep(1000);
            Console.Write($"{i}...");
        }

        PressEnter();
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per .NET .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
###############################################################################
# function iam_create_user_assume_role
#
# Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
#
#     "IAM access" permissions are needed to run this code.
#     "STS assume role" permissions are needed to run this code. (Note: It might
 be necessary to
```

```
#           create a custom policy).
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function iam_create_user_assume_role() {
  {
    if [ "$IAM_OPERATIONS_SOURCED" != "True" ]; then

      source ./iam_operations.sh
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the IAM create user and assume role demo."
  echo
  echo "This demo will create an IAM user, create an IAM role, and apply the role
 to the user."
  echo_repeat "*" 88
  echo

  echo -n "Enter a name for a new IAM user: "
  get_input
  user_name=$get_input_result

  local user_arn
  user_arn=$(iam_create_user -u "$user_name")

  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created demo IAM user named $user_name"
  else
    errecho "$user_arn"
    errecho "The user failed to create. This demo will exit."
    return 1
  fi

  local access_key_response
  access_key_response=$(iam_create_user_access_key -u "$user_name")
  # shellcheck disable=SC2181
  if [[ ${?} != 0 ]]; then
    errecho "The access key failed to create. This demo will exit."
    clean_up "$user_name"
```

```
    return 1
  fi

  IFS=$'\t ' read -r -a access_key_values <<<"$access_key_response"
  local key_name=${access_key_values[0]}
  local key_secret=${access_key_values[1]}

  echo "Created access key named $key_name"

  echo "Wait 10 seconds for the user to be ready."
  sleep 10
  echo_repeat "*" 88
  echo

  local iam_role_name
  iam_role_name=$(generate_random_name "test-role")
  echo "Creating a role named $iam_role_name with user $user_name as the
principal."

  local assume_role_policy_document="{
    \"Version\": \"2012-10-17\",
    \"Statement\": [{
        \"Effect\": \"Allow\",
        \"Principal\": {\"AWS\": \"$user_arn\"},
        \"Action\": \"sts:AssumeRole\"
        }]
    }"

  local role_arn
  role_arn=$(iam_create_role -n "$iam_role_name" -p
"$assume_role_policy_document")

  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    echo "Created IAM role named $iam_role_name"
  else
    errecho "The role failed to create. This demo will exit."
    clean_up "$user_name" "$key_name"
    return 1
  fi

  local policy_name
  policy_name=$(generate_random_name "test-policy")
  local policy_document="{
```

```
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]}"

  local policy_arn
  policy_arn=$(iam_create_policy -n "$policy_name" -p "$policy_document")
  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created  IAM policy named $policy_name"
  else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name"
    return 1
  fi

  if (iam_attach_role_policy -n "$iam_role_name" -p "$policy_arn"); then
    echo "Attached policy $policy_arn to role $iam_role_name"
  else
    errecho "The policy failed to attach."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
    return 1
  fi

  local assume_role_policy_document="{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"$role_arn\"}]}"

  local assume_role_policy_name
  assume_role_policy_name=$(generate_random_name "test-assume-role-")

  # shellcheck disable=SC2181
  local assume_role_policy_arn
  assume_role_policy_arn=$(iam_create_policy -n "$assume_role_policy_name" -p
"$assume_role_policy_document")
  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    echo "Created  IAM policy named $assume_role_policy_name for sts assume role"
  else
    errecho "The policy failed to create."
```

```
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn"
    return 1
  fi

  echo "Wait 10 seconds to give AWS time to propagate these new resources and
connections."
  sleep 10
  echo_repeat "*" 88
  echo

  echo "Try to list buckets without the new user assuming the role."
  echo_repeat "*" 88
  echo

  # Set the environment variables for the created user.
  # bashsupport disable=BP2001
  export AWS_ACCESS_KEY_ID=$key_name
  # bashsupport disable=BP2001
  export AWS_SECRET_ACCESS_KEY=$key_secret

  local buckets
  buckets=$(s3_list_buckets)

  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. This should not have
happened."
  else
    errecho "Because the role with permissions has not been assumed, listing
buckets failed."
  fi

  echo
  echo_repeat "*" 88
  echo "Now assume the role $iam_role_name and list the buckets."
  echo_repeat "*" 88
  echo

  local credentials

  credentials=$(sts_assume_role -r "$role_arn" -n "AssumeRoleDemoSession")
```

```
 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   echo "Assumed role $iam_role_name"
 else
   errecho "Failed to assume role."
   export AWS_ACCESS_KEY_ID=""
   export AWS_SECRET_ACCESS_KEY=""
   clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn" "$assume_role_policy_arn"
   return 1
 fi

 IFS=$'\t ' read -r -a credentials <<<"$credentials"

 export AWS_ACCESS_KEY_ID=${credentials[0]}
 export AWS_SECRET_ACCESS_KEY=${credentials[1]}
 # bashsupport disable=BP2001
 export AWS_SESSION_TOKEN=${credentials[2]}

 buckets=$(s3_list_buckets)

 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   local bucket_count
   bucket_count=$(echo "$buckets" | wc -w | xargs)
   echo "There are $bucket_count buckets in the account. Listing buckets
succeeded because of "
   echo "the assumed role."
 else
   errecho "Failed to list buckets. This should not happen."
   export AWS_ACCESS_KEY_ID=""
   export AWS_SECRET_ACCESS_KEY=""
   export AWS_SESSION_TOKEN=""
   clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn" "$assume_role_policy_arn"
   return 1
 fi

 local result=0
 export AWS_ACCESS_KEY_ID=""
 export AWS_SECRET_ACCESS_KEY=""

 echo
 echo_repeat "*" 88
```

```
  echo "The created resources will now be deleted."
  echo_repeat "*" 88
  echo

  clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
 "$assume_role_policy_arn"

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    result=1
  fi

  return $result
}
```

Le funzioni IAM utilizzate in questo scenario.

```
###############################################################################
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
# Returns:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
###############################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}
```

```
  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}

########################################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
########################################################################################
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an AWS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
```

```
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    User name:   $user_name"
iecho ""

# If the user already exists, we don't want to try to create it.
if (iam_user_exists "$user_name"); then
  errecho "ERROR: A user with that name already exists in the account."
  return 1
fi

response=$(aws iam create-user --user-name "$user_name" \
  --output text \
  --query 'User.Arn')

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports create-user operation failed.$response"
  return 1
fi

echo "$response"
```

```
  return 0
}

########################################################################################
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
#     And:
#       0 - If successful.
#       1 - If it fails.
########################################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]   Optional file name for the access key output."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:f:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      f) file_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
```

```
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  fi

  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}

###############################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
```

```
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_json -- The assume role policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
```

```
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_policy() {
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.
```

```
# bashsupport disable=BP5008
function usage() {
  echo "function iam_create_policy"
  echo "Creates an AWS Identity and Access Management (IAM) policy."
  echo "  -n policy_name   The name of the IAM policy."
  echo "  -p policy_json -- The policy document."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) policy_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_name" ]]; then
  errecho "ERROR: You must provide a policy name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-policy \
  --policy-name "$policy_name" \
  --policy-document "$policy_document" \
  --output text \
```

```
    --query Policy.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
```

```
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam attach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

########################################################################################
```

```
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1
```

```
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.
```

```
# bashsupport disable=BP5008
function usage() {
  echo "function iam_delete_policy"
  echo "Deletes an AWS Identity and Access Management (IAM) policy"
  echo "  -n policy_arn -- The name of the IAM policy arn."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy arn with the -n parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    Policy arn:  $policy_arn"
iecho ""

response=$(aws iam delete-policy \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
```

```
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an AWS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
```

```
        return 1
        ;;
    esac
  done
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""

  response=$(aws iam delete-role \
    --role-name "$role_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}

########################################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
```

```
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_access_key() {
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_access_key"
    echo "Deletes an AWS Identity and Access Management (IAM) access key for the
 specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:k:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      k) access_key="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
```

```
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Username:   $user_name"
  iecho "    Access key:   $access_key"
  iecho ""

  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
```

```
    echo "function iam_delete_user"
    echo "Deletes an AWS Identity and Access Management (IAM) user. You must
supply a username:"
    echo "  -u user_name    The name of the user."
    echo ""
 }

 # Retrieve the calling parameters.
 while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
 done
 export OPTIND=1

 if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
 fi

 iecho "Parameters:\n"
 iecho "    User name:   $user_name"
 iecho ""

 # If the user does not exist, we don't want to try to delete it.
 if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
 fi

 response=$(aws iam delete-user \
    --user-name "$user_name")

 local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
namespace AwsDoc {
    namespace IAM {

        //! Cleanup by deleting created entities.
        /*!
          \sa DeleteCreatedEntities
          \param client: IAM client.
          \param role: IAM role.
          \param user: IAM user.
          \param policy: IAM policy.
        */
        static bool DeleteCreatedEntities(const Aws::IAM::IAMClient &client,
                                          const Aws::IAM::Model::Role &role,
                                          const Aws::IAM::Model::User &user,
                                          const Aws::IAM::Model::Policy &policy);
    }

    static const int LIST_BUCKETS_WAIT_SEC = 20;

    static const char ALLOCATION_TAG[] = "example_code";
}

//! Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
// "IAM access" permissions are needed to run this code.
// "STS assume role" permissions are needed to run this code. (Note: It might be
 necessary to
//    create a custom policy).
/*!
  \sa iamCreateUserAssumeRoleScenario
  \param clientConfig: Aws client configuration.
  \return bool: Successful completion.
```

```
*/
bool AwsDoc::IAM::iamCreateUserAssumeRoleScenario(
        const Aws::Client::ClientConfiguration &clientConfig) {

    Aws::IAM::IAMClient client(clientConfig);
    Aws::IAM::Model::User user;
    Aws::IAM::Model::Role role;
    Aws::IAM::Model::Policy policy;

    // 1. Create a user.
    {
        Aws::IAM::Model::CreateUserRequest request;
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String userName = "iam-demo-user-" +
                               Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetUserName(userName);

        Aws::IAM::Model::CreateUserOutcome outcome = client.CreateUser(request);
        if (!outcome.IsSuccess()) {
            std::cout << "Error creating IAM user " << userName << ":" <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }
        else {
            std::cout << "Successfully created IAM user " << userName <<
 std::endl;
        }

        user = outcome.GetResult().GetUser();
    }

    // 2. Create a role.
    {
        // Get the IAM user for the current client in order to access its ARN.
        Aws::String iamUserArn;
        {
            Aws::IAM::Model::GetUserRequest request;
            Aws::IAM::Model::GetUserOutcome outcome = client.GetUser(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error getting Iam user. " <<
                          outcome.GetError().GetMessage() << std::endl;

                DeleteCreatedEntities(client, role, user, policy);
                return false;
```

```
            }
            else {
                std::cout << "Successfully retrieved Iam user "
                          << outcome.GetResult().GetUser().GetUserName()
                          << std::endl;
            }

            iamUserArn = outcome.GetResult().GetUser().GetArn();
        }

        Aws::IAM::Model::CreateRoleRequest request;

        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String roleName = "iam-demo-role-" +
                               Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetRoleName(roleName);

        // Build policy document for role.
        Aws::Utils::Document jsonStatement;
        jsonStatement.WithString("Effect", "Allow");

        Aws::Utils::Document jsonPrincipal;
        jsonPrincipal.WithString("AWS", iamUserArn);
        jsonStatement.WithObject("Principal", jsonPrincipal);
        jsonStatement.WithString("Action", "sts:AssumeRole");
        jsonStatement.WithObject("Condition", Aws::Utils::Document());

        Aws::Utils::Document policyDocument;
        policyDocument.WithString("Version", "2012-10-17");

        Aws::Utils::Array<Aws::Utils::Document> statements(1);
        statements[0] = jsonStatement;
        policyDocument.WithArray("Statement", statements);

        std::cout << "Setting policy for role\n   "
                  << policyDocument.View().WriteCompact() << std::endl;

        // Set role policy document as JSON string.

    request.SetAssumeRolePolicyDocument(policyDocument.View().WriteCompact());

        Aws::IAM::Model::CreateRoleOutcome outcome = client.CreateRole(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error creating role. " <<
```

```
                    outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully created a role with name " << roleName
                      << std::endl;
        }

        role = outcome.GetResult().GetRole();
    }

    // 3. Create an IAM policy.
    {
        Aws::IAM::Model::CreatePolicyRequest request;
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String policyName = "iam-demo-policy-" +
                                 Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetPolicyName(policyName);

        // Build IAM policy document.
        Aws::Utils::Document jsonStatement;
        jsonStatement.WithString("Effect", "Allow");
        jsonStatement.WithString("Action", "s3:ListAllMyBuckets");
        jsonStatement.WithString("Resource", "arn:aws:s3:::*");

        Aws::Utils::Document policyDocument;
        policyDocument.WithString("Version", "2012-10-17");

        Aws::Utils::Array<Aws::Utils::Document> statements(1);
        statements[0] = jsonStatement;
        policyDocument.WithArray("Statement", statements);

        std::cout << "Creating a policy.\n   " <<
policyDocument.View().WriteCompact()
                  << std::endl;

        // Set IAM policy document as JSON string.
        request.SetPolicyDocument(policyDocument.View().WriteCompact());

        Aws::IAM::Model::CreatePolicyOutcome outcome =
client.CreatePolicy(request);
        if (!outcome.IsSuccess()) {
```

```
            std::cerr << "Error creating policy. " <<
                      outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully created a policy with name, " <<
policyName <<
                      "." << std::endl;
        }

        policy = outcome.GetResult().GetPolicy();
    }

    // 4. Assume the new role using the AWS Security Token Service (STS).
    Aws::STS::Model::Credentials credentials;
    {
        Aws::STS::STSClient stsClient(clientConfig);

        Aws::STS::Model::AssumeRoleRequest request;
        request.SetRoleArn(role.GetArn());
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String roleSessionName = "iam-demo-role-session-" +

Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetRoleSessionName(roleSessionName);

        Aws::STS::Model::AssumeRoleOutcome assumeRoleOutcome;

        // Repeatedly call AssumeRole, because there is often a delay
        // before the role is available to be assumed.
        // Repeat at most 20 times when access is denied.
        int count = 0;
        while (true) {
            assumeRoleOutcome = stsClient.AssumeRole(request);
            if (!assumeRoleOutcome.IsSuccess()) {
                if (count > 20 ||
                    assumeRoleOutcome.GetError().GetErrorType() !=
                    Aws::STS::STSErrors::ACCESS_DENIED) {
                    std::cerr << "Error assuming role after 20 tries. " <<
                              assumeRoleOutcome.GetError().GetMessage() <<
std::endl;
```

```
                    DeleteCreatedEntities(client, role, user, policy);
                    return false;
                }
                std::this_thread::sleep_for(std::chrono::seconds(1));
            }
            else {
                std::cout << "Successfully assumed the role after " << count
                          << " seconds." << std::endl;
                break;
            }
            count++;
        }

        credentials = assumeRoleOutcome.GetResult().GetCredentials();
    }


    // 5. List objects in the bucket (This should fail).
    {
        Aws::S3::S3Client s3Client(
                Aws::Auth::AWSCredentials(credentials.GetAccessKeyId(),
                                          credentials.GetSecretAccessKey(),
                                          credentials.GetSessionToken()),
                Aws::MakeShared<Aws::S3::S3EndpointProvider>(ALLOCATION_TAG),
                clientConfig);
        Aws::S3::Model::ListBucketsOutcome listBucketsOutcome =
s3Client.ListBuckets();
        if (!listBucketsOutcome.IsSuccess()) {
            if (listBucketsOutcome.GetError().GetErrorType() !=
                Aws::S3::S3Errors::ACCESS_DENIED) {
                std::cerr << "Could not lists buckets. " <<
                          listBucketsOutcome.GetError().GetMessage() <<
std::endl;
            }
            else {
                std::cout
                        << "Access to list buckets denied because privileges have
not been applied."
                        << std::endl;
            }
        }
        else {
            std::cerr
```

```
                    << "Successfully retrieved bucket lists when this should not
happen."
                    << std::endl;
        }
    }

    // 6. Attach the policy to the role.
    {
        Aws::IAM::Model::AttachRolePolicyRequest request;
        request.SetRoleName(role.GetRoleName());
        request.WithPolicyArn(policy.GetArn());

        Aws::IAM::Model::AttachRolePolicyOutcome outcome =
client.AttachRolePolicy(
                request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error creating policy. " <<
                      outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully attached the policy with name, "
                      << policy.GetPolicyName() <<
                      ", to the role, " << role.GetRoleName() << "." <<
std::endl;
        }
    }

    int count = 0;
    // 7. List objects in the bucket (this should succeed).
    // Repeatedly call ListBuckets, because there is often a delay
    // before the policy with ListBucket permissions has been applied to the
role.
    // Repeat at most LIST_BUCKETS_WAIT_SEC times when access is denied.
    while (true) {
        Aws::S3::S3Client s3Client(
                Aws::Auth::AWSCredentials(credentials.GetAccessKeyId(),
                                          credentials.GetSecretAccessKey(),
                                          credentials.GetSessionToken()),
                Aws::MakeShared<Aws::S3::S3EndpointProvider>(ALLOCATION_TAG),
                clientConfig);
```

```
        Aws::S3::Model::ListBucketsOutcome listBucketsOutcome =
 s3Client.ListBuckets();
        if (!listBucketsOutcome.IsSuccess()) {
            if ((count > LIST_BUCKETS_WAIT_SEC) ||
                listBucketsOutcome.GetError().GetErrorType() !=
                Aws::S3::S3Errors::ACCESS_DENIED) {
                std::cerr << "Could not lists buckets after " <<
 LIST_BUCKETS_WAIT_SEC << " seconds. " <<
                          listBucketsOutcome.GetError().GetMessage() <<
 std::endl;
                DeleteCreatedEntities(client, role, user, policy);
                return false;
            }

            std::this_thread::sleep_for(std::chrono::seconds(1));
        }
        else {

            std::cout << "Successfully retrieved bucket lists after " << count
                      << " seconds." << std::endl;
            break;
        }
        count++;
    }

    // 8. Delete all the created resources.
    return DeleteCreatedEntities(client, role, user, policy);
}

bool AwsDoc::IAM::DeleteCreatedEntities(const Aws::IAM::IAMClient &client,
                                        const Aws::IAM::Model::Role &role,
                                        const Aws::IAM::Model::User &user,
                                        const Aws::IAM::Model::Policy &policy) {
    bool result = true;
    if (policy.ArnHasBeenSet()) {
        // Detach the policy from the role.
        {
            Aws::IAM::Model::DetachRolePolicyRequest request;
            request.SetPolicyArn(policy.GetArn());
            request.SetRoleName(role.GetRoleName());

            Aws::IAM::Model::DetachRolePolicyOutcome outcome =
 client.DetachRolePolicy(
                    request);
```

```
            if (!outcome.IsSuccess()) {
                std::cerr << "Error Detaching policy from roles. " <<
                          outcome.GetError().GetMessage() << std::endl;
                result = false;
            }
            else {
                std::cout << "Successfully detached the policy with arn "
                          << policy.GetArn()
                          << " from role " << role.GetRoleName() << "." <<
std::endl;
            }
        }

        // Delete the policy.
        {
            Aws::IAM::Model::DeletePolicyRequest request;
            request.WithPolicyArn(policy.GetArn());

            Aws::IAM::Model::DeletePolicyOutcome outcome =
client.DeletePolicy(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error deleting policy. " <<
                          outcome.GetError().GetMessage() << std::endl;
                result = false;
            }
            else {
                std::cout << "Successfully deleted the policy with arn "
                          << policy.GetArn() << std::endl;
            }
        }

    }

    if (role.RoleIdHasBeenSet()) {
        // Delete the role.
        Aws::IAM::Model::DeleteRoleRequest request;
        request.SetRoleName(role.GetRoleName());

        Aws::IAM::Model::DeleteRoleOutcome outcome = client.DeleteRole(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error deleting role. " <<
                      outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
```

```
        else {
            std::cout << "Successfully deleted the role with name "
                      << role.GetRoleName() << std::endl;
        }
    }

    if (user.ArnHasBeenSet()) {
        // Delete the user.
        Aws::IAM::Model::DeleteUserRequest request;
        request.WithUserName(user.GetUserName());

        Aws::IAM::Model::DeleteUserOutcome outcome = client.DeleteUser(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error deleting user. " <<
                      outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Successfully deleted the user with name "
                      << user.GetUserName() << std::endl;
        }
    }

    return result;
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per C++ .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy

- DetachRolePolicy
- PutUserPolicy

Go

SDK per Go V2

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
import (
 "context"
 "errors"
 "fmt"
 "log"
 "math/rand"
 "strings"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/credentials"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/aws-sdk-go-v2/service/s3"
 "github.com/aws/aws-sdk-go-v2/service/sts"
 "github.com/aws/smithy-go"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/iam/actions"
)

// AssumeRoleScenario shows you how to use the AWS Identity and Access Management
 (IAM)
// service to perform the following actions:
//
```

```
//  1. Create a user who has no permissions.
//  2. Create a role that grants permission to list Amazon Simple Storage Service
//     (Amazon S3) buckets for the account.
//  3. Add a policy to let the user assume the role.
//  4. Try and fail to list buckets without permissions.
//  5. Assume the role and list S3 buckets using temporary credentials.
//  6. Delete the policy, role, and user.
type AssumeRoleScenario struct {
 sdkConfig      aws.Config
 accountWrapper actions.AccountWrapper
 policyWrapper  actions.PolicyWrapper
 roleWrapper    actions.RoleWrapper
 userWrapper    actions.UserWrapper
 questioner     demotools.IQuestioner
 helper         IScenarioHelper
 isTestRun      bool
}

// NewAssumeRoleScenario constructs an AssumeRoleScenario instance from a
 configuration.
// It uses the specified config to get an IAM client and create wrappers for the
 actions
// used in the scenario.
func NewAssumeRoleScenario(sdkConfig aws.Config, questioner
 demotools.IQuestioner,
 helper IScenarioHelper) AssumeRoleScenario {
 iamClient := iam.NewFromConfig(sdkConfig)
 return AssumeRoleScenario{
  sdkConfig:      sdkConfig,
  accountWrapper: actions.AccountWrapper{IamClient: iamClient},
  policyWrapper:  actions.PolicyWrapper{IamClient: iamClient},
  roleWrapper:    actions.RoleWrapper{IamClient: iamClient},
  userWrapper:    actions.UserWrapper{IamClient: iamClient},
  questioner:     questioner,
  helper:         helper,
 }
}

// addTestOptions appends the API options specified in the original configuration
 to
// another configuration. This is used to attach the middleware stubber to
 clients
// that are constructed during the scenario, which is needed for unit testing.
func (scenario AssumeRoleScenario) addTestOptions(scenarioConfig *aws.Config) {
```

```
 if scenario.isTestRun {
  scenarioConfig.APIOptions = append(scenarioConfig.APIOptions,
 scenario.sdkConfig.APIOptions...)
 }
}

// Run runs the interactive scenario.
func (scenario AssumeRoleScenario) Run(ctx context.Context) {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong with the demo.\n")
   log.Println(r)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Println("Welcome to the AWS Identity and Access Management (IAM) assume role
 demo.")
 log.Println(strings.Repeat("-", 88))

 user := scenario.CreateUser(ctx)
 accessKey := scenario.CreateAccessKey(ctx, user)
 role := scenario.CreateRoleAndPolicies(ctx, user)
 noPermsConfig := scenario.ListBucketsWithoutPermissions(ctx, accessKey)
 scenario.ListBucketsWithAssumedRole(ctx, noPermsConfig, role)
 scenario.Cleanup(ctx, user, role)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}

// CreateUser creates a new IAM user. This user has no permissions.
func (scenario AssumeRoleScenario) CreateUser(ctx context.Context) *types.User {
 log.Println("Let's create an example user with no permissions.")
 userName := scenario.questioner.Ask("Enter a name for the example user:",
 demotools.NotEmpty{})
 user, err := scenario.userWrapper.GetUser(ctx, userName)
 if err != nil {
  panic(err)
 }
 if user == nil {
  user, err = scenario.userWrapper.CreateUser(ctx, userName)
  if err != nil {
```

```
   panic(err)
  }
  log.Printf("Created user %v.\n", *user.UserName)
 } else {
  log.Printf("User %v already exists.\n", *user.UserName)
 }
 log.Println(strings.Repeat("-", 88))
 return user
}

// CreateAccessKey creates an access key for the user.
func (scenario AssumeRoleScenario) CreateAccessKey(ctx context.Context, user
 *types.User) *types.AccessKey {
 accessKey, err := scenario.userWrapper.CreateAccessKeyPair(ctx, *user.UserName)
 if err != nil {
  panic(err)
 }
 log.Printf("Created access key %v for your user.", *accessKey.AccessKeyId)
 log.Println("Waiting a few seconds for your user to be ready...")
 scenario.helper.Pause(10)
 log.Println(strings.Repeat("-", 88))
 return accessKey
}

// CreateRoleAndPolicies creates a policy that grants permission to list S3
 buckets for
// the current account and attaches the policy to a newly created role. It also
 adds an
// inline policy to the specified user that grants the user permission to assume
 the role.
func (scenario AssumeRoleScenario) CreateRoleAndPolicies(ctx context.Context,
 user *types.User) *types.Role {
 log.Println("Let's create a role and policy that grant permission to list S3
 buckets.")
 scenario.questioner.Ask("Press Enter when you're ready.")
 listBucketsRole, err := scenario.roleWrapper.CreateRole(ctx,
 scenario.helper.GetName(), *user.Arn)
 if err != nil {
  panic(err)
 }
 log.Printf("Created role %v.\n", *listBucketsRole.RoleName)
 listBucketsPolicy, err := scenario.policyWrapper.CreatePolicy(
  ctx, scenario.helper.GetName(), []string{"s3:ListAllMyBuckets"},
 "arn:aws:s3:::*")
```

```
 if err != nil {
  panic(err)
 }
 log.Printf("Created policy %v.\n", *listBucketsPolicy.PolicyName)
 err = scenario.roleWrapper.AttachRolePolicy(ctx, *listBucketsPolicy.Arn,
 *listBucketsRole.RoleName)
 if err != nil {
  panic(err)
 }
 log.Printf("Attached policy %v to role %v.\n", *listBucketsPolicy.PolicyName,
  *listBucketsRole.RoleName)
 err = scenario.userWrapper.CreateUserPolicy(ctx, *user.UserName,
 scenario.helper.GetName(),
  []string{"sts:AssumeRole"}, *listBucketsRole.Arn)
 if err != nil {
  panic(err)
 }
 log.Printf("Created an inline policy for user %v that lets the user assume the
 role.\n",
  *user.UserName)
 log.Println("Let's give AWS a few seconds to propagate these new resources and
 connections...")
 scenario.helper.Pause(10)
 log.Println(strings.Repeat("-", 88))
 return listBucketsRole
}

// ListBucketsWithoutPermissions creates an Amazon S3 client from the user's
 access key
// credentials and tries to list buckets for the account. Because the user does
 not have
// permission to perform this action, the action fails.
func (scenario AssumeRoleScenario) ListBucketsWithoutPermissions(ctx
 context.Context, accessKey *types.AccessKey) *aws.Config {
 log.Println("Let's try to list buckets without permissions. This should return
 an AccessDenied error.")
 scenario.questioner.Ask("Press Enter when you're ready.")
 noPermsConfig, err := config.LoadDefaultConfig(ctx,
  config.WithCredentialsProvider(credentials.NewStaticCredentialsProvider(
   *accessKey.AccessKeyId, *accessKey.SecretAccessKey, ""),
  ))
 if err != nil {
  panic(err)
 }
```

```
 // Add test options if this is a test run. This is needed only for testing
 purposes.
 scenario.addTestOptions(&noPermsConfig)

 s3Client := s3.NewFromConfig(noPermsConfig)
 _, err = s3Client.ListBuckets(ctx, &s3.ListBucketsInput{})
 if err != nil {
  // The SDK for Go does not model the AccessDenied error, so check ErrorCode
 directly.
  var ae smithy.APIError
  if errors.As(err, &ae) {
   switch ae.ErrorCode() {
   case "AccessDenied":
    log.Println("Got AccessDenied error, which is the expected result because\n"
 +
     "the ListBuckets call was made without permissions.")
   default:
    log.Println("Expected AccessDenied, got something else.")
    panic(err)
   }
  }
 } else {
  log.Println("Expected AccessDenied error when calling ListBuckets without
 permissions,\n" +
   "but the call succeeded. Continuing the example anyway...")
 }
 log.Println(strings.Repeat("-", 88))
 return &noPermsConfig
}

// ListBucketsWithAssumedRole performs the following actions:
//
//  1. Creates an AWS Security Token Service (AWS STS) client from the config
 created from
//     the user's access key credentials.
//  2. Gets temporary credentials by assuming the role that grants permission to
 list the
//     buckets.
//  3. Creates an Amazon S3 client from the temporary credentials.
//  4. Lists buckets for the account. Because the temporary credentials are
 generated by
//     assuming the role that grants permission, the action succeeds.
```

```
func (scenario AssumeRoleScenario) ListBucketsWithAssumedRole(ctx
 context.Context, noPermsConfig *aws.Config, role *types.Role) {
 log.Println("Let's assume the role that grants permission to list buckets and
 try again.")
 scenario.questioner.Ask("Press Enter when you're ready.")
 stsClient := sts.NewFromConfig(*noPermsConfig)
 tempCredentials, err := stsClient.AssumeRole(ctx, &sts.AssumeRoleInput{
  RoleArn:         role.Arn,
  RoleSessionName: aws.String("AssumeRoleExampleSession"),
  DurationSeconds: aws.Int32(900),
 })
 if err != nil {
  log.Printf("Couldn't assume role %v.\n", *role.RoleName)
  panic(err)
 }
 log.Printf("Assumed role %v, got temporary credentials.\n", *role.RoleName)
 assumeRoleConfig, err := config.LoadDefaultConfig(ctx,
  config.WithCredentialsProvider(credentials.NewStaticCredentialsProvider(
   *tempCredentials.Credentials.AccessKeyId,
   *tempCredentials.Credentials.SecretAccessKey,
   *tempCredentials.Credentials.SessionToken),
  ),
 )
 if err != nil {
  panic(err)
 }

 // Add test options if this is a test run. This is needed only for testing
 purposes.
 scenario.addTestOptions(&assumeRoleConfig)

 s3Client := s3.NewFromConfig(assumeRoleConfig)
 result, err := s3Client.ListBuckets(ctx, &s3.ListBucketsInput{})
 if err != nil {
  log.Println("Couldn't list buckets with assumed role credentials.")
  panic(err)
 }
 log.Println("Successfully called ListBuckets with assumed role credentials, \n"
 +
  "here are some of them:")
 for i := 0; i < len(result.Buckets) && i < 5; i++ {
  log.Printf("\t%v\n", *result.Buckets[i].Name)
 }
 log.Println(strings.Repeat("-", 88))
```

```
}

// Cleanup deletes all resources created for the scenario.
func (scenario AssumeRoleScenario) Cleanup(ctx context.Context, user *types.User,
 role *types.Role) {
 if scenario.questioner.AskBool(
  "Do you want to delete the resources created for this example? (y/n)", "y",
 ) {
  policies, err := scenario.roleWrapper.ListAttachedRolePolicies(ctx,
 *role.RoleName)
  if err != nil {
   panic(err)
  }
  for _, policy := range policies {
   err = scenario.roleWrapper.DetachRolePolicy(ctx, *role.RoleName,
 *policy.PolicyArn)
   if err != nil {
    panic(err)
   }
   err = scenario.policyWrapper.DeletePolicy(ctx, *policy.PolicyArn)
   if err != nil {
    panic(err)
   }
   log.Printf("Detached policy %v from role %v and deleted the policy.\n",
    *policy.PolicyName, *role.RoleName)
  }
  err = scenario.roleWrapper.DeleteRole(ctx, *role.RoleName)
  if err != nil {
   panic(err)
  }
  log.Printf("Deleted role %v.\n", *role.RoleName)

  userPols, err := scenario.userWrapper.ListUserPolicies(ctx, *user.UserName)
  if err != nil {
   panic(err)
  }
  for _, userPol := range userPols {
   err = scenario.userWrapper.DeleteUserPolicy(ctx, *user.UserName, userPol)
   if err != nil {
    panic(err)
   }
   log.Printf("Deleted policy %v from user %v.\n", userPol, *user.UserName)
  }
  keys, err := scenario.userWrapper.ListAccessKeys(ctx, *user.UserName)
```

```
  if err != nil {
   panic(err)
  }
  for _, key := range keys {
   err = scenario.userWrapper.DeleteAccessKey(ctx, *user.UserName,
 *key.AccessKeyId)
   if err != nil {
    panic(err)
   }
   log.Printf("Deleted access key %v from user %v.\n", *key.AccessKeyId,
 *user.UserName)
  }
  err = scenario.userWrapper.DeleteUser(ctx, *user.UserName)
  if err != nil {
   panic(err)
  }
  log.Printf("Deleted user %v.\n", *user.UserName)
  log.Println(strings.Repeat("-", 88))
 }

}

// IScenarioHelper abstracts input and wait functions from a scenario so that
 they
// can be mocked for unit testing.
type IScenarioHelper interface {
 GetName() string
 Pause(secs int)
}

const rMax = 100000

type ScenarioHelper struct {
 Prefix string
 Random *rand.Rand
}

// GetName returns a unique name formed of a prefix and a random number.
func (helper *ScenarioHelper) GetName() string {
 return fmt.Sprintf("%v%v", helper.Prefix, helper.Random.Intn(rMax))
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
```

```
 time.Sleep(time.Duration(secs) * time.Second)
}
```

Definisci una struttura che racchiude le azioni dell'account.

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// AccountWrapper encapsulates AWS Identity and Access Management (IAM) account
 actions
// used in the examples.
// It contains an IAM service client that is used to perform account actions.
type AccountWrapper struct {
 IamClient *iam.Client
}



// GetAccountPasswordPolicy gets the account password policy for the current
 account.
// If no policy has been set, a NoSuchEntityException is error is returned.
func (wrapper AccountWrapper) GetAccountPasswordPolicy(ctx context.Context)
 (*types.PasswordPolicy, error) {
 var pwPolicy *types.PasswordPolicy
 result, err := wrapper.IamClient.GetAccountPasswordPolicy(ctx,
  &iam.GetAccountPasswordPolicyInput{})
 if err != nil {
  log.Printf("Couldn't get account password policy. Here's why: %v\n", err)
 } else {
  pwPolicy = result.PasswordPolicy
 }
 return pwPolicy, err
}
```

```
// ListSAMLProviders gets the SAML providers for the account.
func (wrapper AccountWrapper) ListSAMLProviders(ctx context.Context)
 ([]types.SAMLProviderListEntry, error) {
 var providers []types.SAMLProviderListEntry
 result, err := wrapper.IamClient.ListSAMLProviders(ctx,
 &iam.ListSAMLProvidersInput{})
 if err != nil {
  log.Printf("Couldn't list SAML providers. Here's why: %v\n", err)
 } else {
  providers = result.SAMLProviderList
 }
 return providers, err
}
```

Definisci una struttura che racchiude le azioni della policy.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// ListPolicies gets up to maxPolicies policies.
func (wrapper PolicyWrapper) ListPolicies(ctx context.Context, maxPolicies int32)
 ([]types.Policy, error) {
```

```
 var policies []types.Policy
 result, err := wrapper.IamClient.ListPolicies(ctx, &iam.ListPoliciesInput{
  MaxItems: aws.Int32(maxPolicies),
 })
 if err != nil {
  log.Printf("Couldn't list policies. Here's why: %v\n", err)
 } else {
  policies = result.Policies
 }
 return policies, err
}



// PolicyDocument defines a policy document as a Go struct that can be serialized
// to JSON.
type PolicyDocument struct {
 Version   string
 Statement []PolicyStatement
}

// PolicyStatement defines a statement in a policy document.
type PolicyStatement struct {
 Effect    string
 Action    []string
 Principal map[string]string `json:",omitempty"`
 Resource  *string           `json:",omitempty"`
}

// CreatePolicy creates a policy that grants a list of actions to the specified
 resource.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper PolicyWrapper) CreatePolicy(ctx context.Context, policyName string,
 actions []string,
 resourceArn string) (*types.Policy, error) {
 var policy *types.Policy
 policyDoc := PolicyDocument{
  Version: "2012-10-17",
  Statement: []PolicyStatement{{
   Effect:   "Allow",
   Action:   actions,
   Resource: aws.String(resourceArn),
```

```
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n",
 resourceArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreatePolicy(ctx, &iam.CreatePolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
 })
 if err != nil {
  log.Printf("Couldn't create policy %v. Here's why: %v\n", policyName, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}



// GetPolicy gets data about a policy.
func (wrapper PolicyWrapper) GetPolicy(ctx context.Context, policyArn string)
 (*types.Policy, error) {
 var policy *types.Policy
 result, err := wrapper.IamClient.GetPolicy(ctx, &iam.GetPolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't get policy %v. Here's why: %v\n", policyArn, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}



// DeletePolicy deletes a policy.
func (wrapper PolicyWrapper) DeletePolicy(ctx context.Context, policyArn string)
 error {
 _, err := wrapper.IamClient.DeletePolicy(ctx, &iam.DeletePolicyInput{
  PolicyArn: aws.String(policyArn),
```

```
 })
 if err != nil {
  log.Printf("Couldn't delete policy %v. Here's why: %v\n", policyArn, err)
 }
 return err
}
```

Definisci una struttura che racchiude le azioni del ruolo.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// ListRoles gets up to maxRoles roles.
func (wrapper RoleWrapper) ListRoles(ctx context.Context, maxRoles int32)
 ([]types.Role, error) {
 var roles []types.Role
 result, err := wrapper.IamClient.ListRoles(ctx,
  &iam.ListRolesInput{MaxItems: aws.Int32(maxRoles)},
 )
 if err != nil {
  log.Printf("Couldn't list roles. Here's why: %v\n", err)
 } else {
  roles = result.Roles
 }
```

```
 return roles, err
}



// CreateRole creates a role that trusts a specified user. The trusted user can
 assume
// the role to acquire its permissions.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper RoleWrapper) CreateRole(ctx context.Context, roleName string,
 trustedUserArn string) (*types.Role, error) {
 var role *types.Role
 trustPolicy := PolicyDocument{
  Version: "2012-10-17",
  Statement: []PolicyStatement{{
   Effect:    "Allow",
   Principal: map[string]string{"AWS": trustedUserArn},
   Action:    []string{"sts:AssumeRole"},
  }},
 }
 policyBytes, err := json.Marshal(trustPolicy)
 if err != nil {
  log.Printf("Couldn't create trust policy for %v. Here's why: %v\n",
 trustedUserArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreateRole(ctx, &iam.CreateRoleInput{
  AssumeRolePolicyDocument: aws.String(string(policyBytes)),
  RoleName:                 aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't create role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// GetRole gets data about a role.
```

```
func (wrapper RoleWrapper) GetRole(ctx context.Context, roleName string)
 (*types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.GetRole(ctx,
  &iam.GetRoleInput{RoleName: aws.String(roleName)})
 if err != nil {
  log.Printf("Couldn't get role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// CreateServiceLinkedRole creates a service-linked role that is owned by the
 specified service.
func (wrapper RoleWrapper) CreateServiceLinkedRole(ctx context.Context,
 serviceName string, description string) (
 *types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.CreateServiceLinkedRole(ctx,
 &iam.CreateServiceLinkedRoleInput{
  AWSServiceName: aws.String(serviceName),
  Description:    aws.String(description),
 })
 if err != nil {
  log.Printf("Couldn't create service-linked role %v. Here's why: %v\n",
 serviceName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// DeleteServiceLinkedRole deletes a service-linked role.
func (wrapper RoleWrapper) DeleteServiceLinkedRole(ctx context.Context, roleName
 string) error {
 _, err := wrapper.IamClient.DeleteServiceLinkedRole(ctx,
 &iam.DeleteServiceLinkedRoleInput{
  RoleName: aws.String(roleName)},
 )
```

```
 if err != nil {
  log.Printf("Couldn't delete service-linked role %v. Here's why: %v\n",
 roleName, err)
 }
 return err
}



// AttachRolePolicy attaches a policy to a role.
func (wrapper RoleWrapper) AttachRolePolicy(ctx context.Context, policyArn
 string, roleName string) error {
 _, err := wrapper.IamClient.AttachRolePolicy(ctx, &iam.AttachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't attach policy %v to role %v. Here's why: %v\n", policyArn,
 roleName, err)
 }
 return err
}



// ListAttachedRolePolicies lists the policies that are attached to the specified
 role.
func (wrapper RoleWrapper) ListAttachedRolePolicies(ctx context.Context, roleName
 string) ([]types.AttachedPolicy, error) {
 var policies []types.AttachedPolicy
 result, err := wrapper.IamClient.ListAttachedRolePolicies(ctx,
 &iam.ListAttachedRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list attached policies for role %v. Here's why: %v\n",
 roleName, err)
 } else {
  policies = result.AttachedPolicies
 }
 return policies, err
}
```

```
// DetachRolePolicy detaches a policy from a role.
func (wrapper RoleWrapper) DetachRolePolicy(ctx context.Context, roleName string,
 policyArn string) error {
 _, err := wrapper.IamClient.DetachRolePolicy(ctx, &iam.DetachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't detach policy from role %v. Here's why: %v\n", roleName,
 err)
 }
 return err
}



// ListRolePolicies lists the inline policies for a role.
func (wrapper RoleWrapper) ListRolePolicies(ctx context.Context, roleName string)
 ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListRolePolicies(ctx,
 &iam.ListRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list policies for role %v. Here's why: %v\n", roleName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}



// DeleteRole deletes a role. All attached policies must be detached before a
// role can be deleted.
func (wrapper RoleWrapper) DeleteRole(ctx context.Context, roleName string) error
 {
 _, err := wrapper.IamClient.DeleteRole(ctx, &iam.DeleteRoleInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
```

```
  log.Printf("Couldn't delete role %v. Here's why: %v\n", roleName, err)
 }
 return err
}
```

Definisci una struttura che racchiude le azioni dell'utente.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListUsers gets up to maxUsers number of users.
func (wrapper UserWrapper) ListUsers(ctx context.Context, maxUsers int32)
 ([]types.User, error) {
 var users []types.User
 result, err := wrapper.IamClient.ListUsers(ctx, &iam.ListUsersInput{
  MaxItems: aws.Int32(maxUsers),
 })
 if err != nil {
  log.Printf("Couldn't list users. Here's why: %v\n", err)
 } else {
  users = result.Users
 }
```

```
 return users, err
}



// GetUser gets data about a user.
func (wrapper UserWrapper) GetUser(ctx context.Context, userName string)
 (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.GetUser(ctx, &iam.GetUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  var apiError smithy.APIError
  if errors.As(err, &apiError) {
   switch apiError.(type) {
   case *types.NoSuchEntityException:
    log.Printf("User %v does not exist.\n", userName)
    err = nil
   default:
    log.Printf("Couldn't get user %v. Here's why: %v\n", userName, err)
   }
  }
 } else {
  user = result.User
 }
 return user, err
}



// CreateUser creates a new user with the specified name.
func (wrapper UserWrapper) CreateUser(ctx context.Context, userName string)
 (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.CreateUser(ctx, &iam.CreateUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
 } else {
  user = result.User
 }
 return user, err
```

```
}



// CreateUserPolicy adds an inline policy to a user. This example creates a
 policy that
// grants a list of actions on a specified role.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper UserWrapper) CreateUserPolicy(ctx context.Context, userName string,
 policyName string, actions []string,
 roleArn string) error {
 policyDoc := PolicyDocument{
  Version: "2012-10-17",
  Statement: []PolicyStatement{{
   Effect:   "Allow",
   Action:   actions,
   Resource: aws.String(roleArn),
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n", roleArn,
 err)
  return err
 }
 _, err = wrapper.IamClient.PutUserPolicy(ctx, &iam.PutUserPolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
  UserName:       aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create policy for user %v. Here's why: %v\n", userName,
 err)
 }
 return err
}



// ListUserPolicies lists the inline policies for the specified user.
func (wrapper UserWrapper) ListUserPolicies(ctx context.Context, userName string)
 ([]string, error) {
```

```
 var policies []string
 result, err := wrapper.IamClient.ListUserPolicies(ctx,
 &iam.ListUserPoliciesInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list policies for user %v. Here's why: %v\n", userName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}



// DeleteUserPolicy deletes an inline policy from a user.
func (wrapper UserWrapper) DeleteUserPolicy(ctx context.Context, userName string,
 policyName string) error {
 _, err := wrapper.IamClient.DeleteUserPolicy(ctx, &iam.DeleteUserPolicyInput{
  PolicyName: aws.String(policyName),
  UserName:   aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete policy from user %v. Here's why: %v\n", userName,
 err)
 }
 return err
}



// DeleteUser deletes a user.
func (wrapper UserWrapper) DeleteUser(ctx context.Context, userName string) error
 {
 _, err := wrapper.IamClient.DeleteUser(ctx, &iam.DeleteUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete user %v. Here's why: %v\n", userName, err)
 }
 return err
}
```

```
// CreateAccessKeyPair creates an access key for a user. The returned access key
 contains
// the ID and secret credentials needed to use the key.
func (wrapper UserWrapper) CreateAccessKeyPair(ctx context.Context, userName
 string) (*types.AccessKey, error) {
 var key *types.AccessKey
 result, err := wrapper.IamClient.CreateAccessKey(ctx, &iam.CreateAccessKeyInput{
  UserName: aws.String(userName)})
 if err != nil {
  log.Printf("Couldn't create access key pair for user %v. Here's why: %v\n",
 userName, err)
 } else {
  key = result.AccessKey
 }
 return key, err
}



// DeleteAccessKey deletes an access key from a user.
func (wrapper UserWrapper) DeleteAccessKey(ctx context.Context, userName string,
 keyId string) error {
 _, err := wrapper.IamClient.DeleteAccessKey(ctx, &iam.DeleteAccessKeyInput{
  AccessKeyId: aws.String(keyId),
  UserName:    aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete access key %v. Here's why: %v\n", keyId, err)
 }
 return err
}



// ListAccessKeys lists the access keys for the specified user.
func (wrapper UserWrapper) ListAccessKeys(ctx context.Context, userName string)
 ([]types.AccessKeyMetadata, error) {
 var keys []types.AccessKeyMetadata
 result, err := wrapper.IamClient.ListAccessKeys(ctx, &iam.ListAccessKeysInput{
  UserName: aws.String(userName),
 })
 if err != nil {
```

```
  log.Printf("Couldn't list access keys for user %v. Here's why: %v\n", userName,
 err)
 } else {
  keys = result.AccessKeyMetadata
 }
 return keys, err
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Go .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
/*
  To run this Java V2 code example, set up your development environment,
 including your credentials.

  For information, see this documentation topic:

  https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html

  This example performs these operations:

  1. Creates a user that has no permissions.
  2. Creates a role and policy that grants Amazon S3 permissions.
  3. Creates a role.
  4. Grants the user permissions.
  5. Gets temporary credentials by assuming the role.  Creates an Amazon S3
 Service client object with the temporary credentials.
  6. Deletes the resources.
 */

public class IAMScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");
    public static final String PolicyDocument = "{" +
            "  \"Version\": \"2012-10-17\"," +
            "  \"Statement\": [" +
            "    {" +
            "        \"Effect\": \"Allow\"," +
            "        \"Action\": [" +
            "            \"s3:*\"" +
            "       ]," +
            "       \"Resource\": \"*\"" +
            "    }" +
            "   ]" +
            "}";

    public static String userArn;

    public static void main(String[] args) throws Exception {

        final String usage = """

                Usage:
```

```
                    <username> <policyName> <roleName> <roleSessionName>
<bucketName>\s

                Where:
                    username - The name of the IAM user to create.\s
                    policyName - The name of the policy to create.\s
                    roleName - The name of the role to create.\s
                    roleSessionName - The name of the session required for the
assumeRole operation.\s
                    bucketName - The name of the Amazon S3 bucket from which
objects are read.\s
                """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String userName = args[0];
        String policyName = args[1];
        String roleName = args[2];
        String roleSessionName = args[3];
        String bucketName = args[4];

        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the AWS IAM example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 1. Create the IAM user.");
        User createUser = createIAMUser(iam, userName);

        System.out.println(DASHES);
        userArn = createUser.arn();

        AccessKey myKey = createIAMAccessKey(iam, userName);
        String accessKey = myKey.accessKeyId();
        String secretKey = myKey.secretAccessKey();
        String assumeRolePolicyDocument = "{" +
```

```
                "\"Version\": \"2012-10-17\"," +
                "\"Statement\": [{" +
                "\"Effect\": \"Allow\"," +
                "\"Principal\": {" +
                " \"AWS\": \"" + userArn + "\"" +
                "}," +
                "\"Action\": \"sts:AssumeRole\"" +
                "}]" +
                "}";

        System.out.println(assumeRolePolicyDocument);
        System.out.println(userName + " was successfully created.");
        System.out.println(DASHES);
        System.out.println("2. Creates a policy.");
        String polArn = createIAMPolicy(iam, policyName);
        System.out.println("The policy " + polArn + " was successfully
created.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Creates a role.");
        TimeUnit.SECONDS.sleep(30);
        String roleArn = createIAMRole(iam, roleName, assumeRolePolicyDocument);
        System.out.println(roleArn + " was successfully created.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Grants the user permissions.");
        attachIAMRolePolicy(iam, roleName, polArn);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Wait for 30 secs so the resource is available");
        TimeUnit.SECONDS.sleep(30);
        System.out.println("5. Gets temporary credentials by assuming the
role.");
        System.out.println("Perform an Amazon S3 Service operation using the
temporary credentials.");
        assumeRole(roleArn, roleSessionName, bucketName, accessKey, secretKey);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6 Getting ready to delete the AWS resources");
        deleteKey(iam, userName, accessKey);
```

```
        deleteRole(iam, roleName, polArn);
        deleteIAMUser(iam, userName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("This IAM Scenario has successfully completed");
        System.out.println(DASHES);
    }

    public static AccessKey createIAMAccessKey(IamClient iam, String user) {
        try {
            CreateAccessKeyRequest request = CreateAccessKeyRequest.builder()
                    .userName(user)
                    .build();

            CreateAccessKeyResponse response = iam.createAccessKey(request);
            return response.accessKey();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static User createIAMUser(IamClient iam, String username) {
        try {
            // Create an IamWaiter object
            IamWaiter iamWaiter = iam.waiter();
            CreateUserRequest request = CreateUserRequest.builder()
                    .userName(username)
                    .build();

            // Wait until the user is created.
            CreateUserResponse response = iam.createUser(request);
            GetUserRequest userRequest = GetUserRequest.builder()
                    .userName(response.user().userName())
                    .build();

            WaiterResponse<GetUserResponse> waitUntilUserExists =
iamWaiter.waitUntilUserExists(userRequest);

waitUntilUserExists.matched().response().ifPresent(System.out::println);
            return response.user();
```

```
        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static String createIAMRole(IamClient iam, String rolename, String
json) {

        try {
            CreateRoleRequest request = CreateRoleRequest.builder()
                    .roleName(rolename)
                    .assumeRolePolicyDocument(json)
                    .description("Created using the AWS SDK for Java")
                    .build();

            CreateRoleResponse response = iam.createRole(request);
            System.out.println("The ARN of the role is " +
response.role().arn());
            return response.role().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

    public static String createIAMPolicy(IamClient iam, String policyName) {
        try {
            // Create an IamWaiter object.
            IamWaiter iamWaiter = iam.waiter();
            CreatePolicyRequest request = CreatePolicyRequest.builder()
                    .policyName(policyName)
                    .policyDocument(PolicyDocument).build();

            CreatePolicyResponse response = iam.createPolicy(request);
            GetPolicyRequest polRequest = GetPolicyRequest.builder()
                    .policyArn(response.policy().arn())
                    .build();
```

```
            WaiterResponse<GetPolicyResponse> waitUntilPolicyExists =
iamWaiter.waitUntilPolicyExists(polRequest);

waitUntilPolicyExists.matched().response().ifPresent(System.out::println);
            return response.policy().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

    public static void attachIAMRolePolicy(IamClient iam, String roleName, String
policyArn) {
        try {
            ListAttachedRolePoliciesRequest request =
ListAttachedRolePoliciesRequest.builder()
                    .roleName(roleName)
                    .build();

            ListAttachedRolePoliciesResponse response =
iam.listAttachedRolePolicies(request);
            List<AttachedPolicy> attachedPolicies = response.attachedPolicies();
            String polArn;
            for (AttachedPolicy policy : attachedPolicies) {
                polArn = policy.policyArn();
                if (polArn.compareTo(policyArn) == 0) {
                    System.out.println(roleName + " policy is already attached to
this role.");
                    return;
                }
            }

            AttachRolePolicyRequest attachRequest =
AttachRolePolicyRequest.builder()
                    .roleName(roleName)
                    .policyArn(policyArn)
                    .build();

            iam.attachRolePolicy(attachRequest);
            System.out.println("Successfully attached policy " + policyArn + " to
role " + roleName);
```

```
        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Invoke an Amazon S3 operation using the Assumed Role.
    public static void assumeRole(String roleArn, String roleSessionName, String
bucketName, String keyVal,
            String keySecret) {

        // Use the creds of the new IAM user that was created in this code
example.
        AwsBasicCredentials credentials = AwsBasicCredentials.create(keyVal,
keySecret);
        StsClient stsClient = StsClient.builder()
                .region(Region.US_EAST_1)

.credentialsProvider(StaticCredentialsProvider.create(credentials))
                .build();

        try {
            AssumeRoleRequest roleRequest = AssumeRoleRequest.builder()
                    .roleArn(roleArn)
                    .roleSessionName(roleSessionName)
                    .build();

            AssumeRoleResponse roleResponse = stsClient.assumeRole(roleRequest);
            Credentials myCreds = roleResponse.credentials();
            String key = myCreds.accessKeyId();
            String secKey = myCreds.secretAccessKey();
            String secToken = myCreds.sessionToken();

            // List all objects in an Amazon S3 bucket using the temp creds
retrieved by
            // invoking assumeRole.
            Region region = Region.US_EAST_1;
            S3Client s3 = S3Client.builder()
                    .credentialsProvider(

StaticCredentialsProvider.create(AwsSessionCredentials.create(key, secKey,
secToken)))
                    .region(region)
                    .build();
```

```
            System.out.println("Created a S3Client using temp credentials.");
            System.out.println("Listing objects in " + bucketName);
            ListObjectsRequest listObjects = ListObjectsRequest.builder()
                    .bucket(bucketName)
                    .build();

            ListObjectsResponse res = s3.listObjects(listObjects);
            List<S3Object> objects = res.contents();
            for (S3Object myValue : objects) {
                System.out.println("The name of the key is " + myValue.key());
                System.out.println("The owner is " + myValue.owner());
            }

        } catch (StsException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void deleteRole(IamClient iam, String roleName, String polArn)
{

        try {
            // First the policy needs to be detached.
            DetachRolePolicyRequest rolePolicyRequest =
DetachRolePolicyRequest.builder()
                    .policyArn(polArn)
                    .roleName(roleName)
                    .build();

            iam.detachRolePolicy(rolePolicyRequest);

            // Delete the policy.
            DeletePolicyRequest request = DeletePolicyRequest.builder()
                    .policyArn(polArn)
                    .build();

            iam.deletePolicy(request);
            System.out.println("*** Successfully deleted " + polArn);

            // Delete the role.
            DeleteRoleRequest roleRequest = DeleteRoleRequest.builder()
                    .roleName(roleName)
```

```
                    .build();

            iam.deleteRole(roleRequest);
            System.out.println("*** Successfully deleted " + roleName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteKey(IamClient iam, String username, String
 accessKey) {
        try {
            DeleteAccessKeyRequest request = DeleteAccessKeyRequest.builder()
                    .accessKeyId(accessKey)
                    .userName(username)
                    .build();

            iam.deleteAccessKey(request);
            System.out.println("Successfully deleted access key " + accessKey +
                    " from user " + username);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteIAMUser(IamClient iam, String userName) {
        try {
            DeleteUserRequest request = DeleteUserRequest.builder()
                    .userName(userName)
                    .build();

            iam.deleteUser(request);
            System.out.println("*** Successfully deleted " + userName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK for Java 2.x .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
import {
  CreateUserCommand,
  GetUserCommand,
  CreateAccessKeyCommand,
  CreatePolicyCommand,
```

```
  CreateRoleCommand,
  AttachRolePolicyCommand,
  DeleteAccessKeyCommand,
  DeleteUserCommand,
  DeleteRoleCommand,
  DeletePolicyCommand,
  DetachRolePolicyCommand,
  IAMClient,
} from "@aws-sdk/client-iam";
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";
import { AssumeRoleCommand, STSClient } from "@aws-sdk/client-sts";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import { ScenarioInput } from "@aws-doc-sdk-examples/lib/scenario/index.js";

// Set the parameters.
const iamClient = new IAMClient({});
const userName = "iam_basic_test_username";
const policyName = "iam_basic_test_policy";
const roleName = "iam_basic_test_role";

/**
 * Create a new IAM user. If the user already exists, give
 * the option to delete and re-create it.
 * @param {string} name
 */
export const createUser = async (name, confirmAll = false) => {
  try {
    const { User } = await iamClient.send(
      new GetUserCommand({ UserName: name }),
    );
    const input = new ScenarioInput(
      "deleteUser",
      "Do you want to delete and remake this user?",
      { type: "confirm" },
    );
    const deleteUser = await input.handle({}, { confirmAll });
    // If the user exists, and you want to delete it, delete the user
    // and then create it again.
    if (deleteUser) {
      await iamClient.send(new DeleteUserCommand({ UserName: User.UserName }));
      await iamClient.send(new CreateUserCommand({ UserName: name }));
    } else {
      console.warn(
        `${name} already exists. The scenario may not work as expected.`,
```

```
      );
      return User;
    }
  } catch (caught) {
    // If there is no user by that name, create one.
    if (caught instanceof Error && caught.name === "NoSuchEntityException") {
      const { User } = await iamClient.send(
        new CreateUserCommand({ UserName: name }),
      );
      return User;
    }
    throw caught;
  }
};

export const main = async (confirmAll = false) => {
  // Create a user. The user has no permissions by default.
  const User = await createUser(userName, confirmAll);

  if (!User) {
    throw new Error("User not created");
  }

  // Create an access key. This key is used to authenticate the new user to
  // Amazon Simple Storage Service (Amazon S3) and AWS Security Token Service
 (AWS STS).
  // It's not best practice to use access keys. For more information, see
 https://aws.amazon.com/iam/resources/best-practices/.
  const createAccessKeyResponse = await iamClient.send(
    new CreateAccessKeyCommand({ UserName: userName }),
  );

  if (
    !createAccessKeyResponse.AccessKey?.AccessKeyId ||
    !createAccessKeyResponse.AccessKey?.SecretAccessKey
  ) {
    throw new Error("Access key not created");
  }

  const {
    AccessKey: { AccessKeyId, SecretAccessKey },
  } = createAccessKeyResponse;

  let s3Client = new S3Client({
```

```
    credentials: {
      accessKeyId: AccessKeyId,
      secretAccessKey: SecretAccessKey,
    },
  });

  // Retry the list buckets operation until it succeeds. InvalidAccessKeyId is
  // thrown while the user and access keys are still stabilizing.
  await retry({ intervalInMs: 1000, maxRetries: 300 }, async () => {
    try {
      return await listBuckets(s3Client);
    } catch (err) {
      if (err instanceof Error && err.name === "InvalidAccessKeyId") {
        throw err;
      }
    }
  });

  // Retry the create role operation until it succeeds. A MalformedPolicyDocument
error
  // is thrown while the user and access keys are still stabilizing.
  const { Role } = await retry(
    {
      intervalInMs: 2000,
      maxRetries: 60,
    },
    () =>
      iamClient.send(
        new CreateRoleCommand({
          AssumeRolePolicyDocument: JSON.stringify({
            Version: "2012-10-17",
            Statement: [
              {
                Effect: "Allow",
                Principal: {
                  // Allow the previously created user to assume this role.
                  AWS: User.Arn,
                },
                Action: "sts:AssumeRole",
              },
            ],
          }),
          RoleName: roleName,
        }),
```

```
    ),
);

if (!Role) {
  throw new Error("Role not created");
}

// Create a policy that allows the user to list S3 buckets.
const { Policy: listBucketPolicy } = await iamClient.send(
  new CreatePolicyCommand({
    PolicyDocument: JSON.stringify({
      Version: "2012-10-17",
      Statement: [
        {
          Effect: "Allow",
          Action: ["s3:ListAllMyBuckets"],
          Resource: "*",
        },
      ],
    }),
    PolicyName: policyName,
  }),
);

if (!listBucketPolicy) {
  throw new Error("Policy not created");
}

// Attach the policy granting the 's3:ListAllMyBuckets' action to the role.
await iamClient.send(
  new AttachRolePolicyCommand({
    PolicyArn: listBucketPolicy.Arn,
    RoleName: Role.RoleName,
  }),
);

// Assume the role.
const stsClient = new STSClient({
  credentials: {
    accessKeyId: AccessKeyId,
    secretAccessKey: SecretAccessKey,
  },
});
```

```
// Retry the assume role operation until it succeeds.
const { Credentials } = await retry(
  { intervalInMs: 2000, maxRetries: 60 },
  () =>
    stsClient.send(
      new AssumeRoleCommand({
        RoleArn: Role.Arn,
        RoleSessionName: `iamBasicScenarioSession-${Math.floor(
          Math.random() * 1000000,
        )}`,
        DurationSeconds: 900,
      }),
    ),
);

if (!Credentials?.AccessKeyId || !Credentials?.SecretAccessKey) {
  throw new Error("Credentials not created");
}

s3Client = new S3Client({
  credentials: {
    accessKeyId: Credentials.AccessKeyId,
    secretAccessKey: Credentials.SecretAccessKey,
    sessionToken: Credentials.SessionToken,
  },
});

// List the S3 buckets again.
// Retry the list buckets operation until it succeeds. AccessDenied might
// be thrown while the role policy is still stabilizing.
await retry({ intervalInMs: 2000, maxRetries: 120 }, () =>
  listBuckets(s3Client),
);

// Clean up.
await iamClient.send(
  new DetachRolePolicyCommand({
    PolicyArn: listBucketPolicy.Arn,
    RoleName: Role.RoleName,
  }),
);

await iamClient.send(
  new DeletePolicyCommand({
```

```
      PolicyArn: listBucketPolicy.Arn,
    }),
  );

  await iamClient.send(
    new DeleteRoleCommand({
      RoleName: Role.RoleName,
    }),
  );

  await iamClient.send(
    new DeleteAccessKeyCommand({
      UserName: userName,
      AccessKeyId,
    }),
  );

  await iamClient.send(
    new DeleteUserCommand({
      UserName: userName,
    }),
  );
};

/**
 *
 * @param {S3Client} s3Client
 */
const listBuckets = async (s3Client) => {
  const { Buckets } = await s3Client.send(new ListBucketsCommand({}));

  if (!Buckets) {
    throw new Error("Buckets not listed");
  }

  console.log(Buckets.map((bucket) => bucket.Name).join("\n"));
};
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per JavaScript .
  - AttachRolePolicy

- CreateAccessKey
- CreatePolicy
- CreateRole
- CreateUser
- DeleteAccessKey
- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

Kotlin

SDK per Kotlin

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <username> <policyName> <roleName> <roleSessionName> <fileLocation>
 <bucketName>

    Where:
        username - The name of the IAM user to create.
        policyName - The name of the policy to create.
        roleName - The name of the role to create.
        roleSessionName - The name of the session required for the assumeRole
 operation.
```

```
        fileLocation - The file location to the JSON required to create the role
 (see Readme).
        bucketName - The name of the Amazon S3 bucket from which objects are
 read.
    """

    if (args.size != 6) {
        println(usage)
        exitProcess(1)
    }

    val userName = args[0]
    val policyName = args[1]
    val roleName = args[2]
    val roleSessionName = args[3]
    val fileLocation = args[4]
    val bucketName = args[5]

    createUser(userName)
    println("$userName was successfully created.")

    val polArn = createPolicy(policyName)
    println("The policy $polArn was successfully created.")

    val roleArn = createRole(roleName, fileLocation)
    println("$roleArn was successfully created.")
    attachRolePolicy(roleName, polArn)

    println("*** Wait for 1 MIN so the resource is available.")
    delay(60000)
    assumeGivenRole(roleArn, roleSessionName, bucketName)

    println("*** Getting ready to delete the AWS resources.")
    deleteRole(roleName, polArn)
    deleteUser(userName)
    println("This IAM Scenario has successfully completed.")
}

suspend fun createUser(usernameVal: String?): String? {
    val request =
        CreateUserRequest {
            userName = usernameVal
        }
```

```
    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createUser(request)
        return response.user?.userName
    }
}

suspend fun createPolicy(policyNameVal: String?): String {
    val policyDocumentValue = """
    {
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "s3:*"
                ],
                "Resource": "*"
            }
        ]
    }
    """.trimIndent()

    val request =
        CreatePolicyRequest {
            policyName = policyNameVal
            policyDocument = policyDocumentValue
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createPolicy(request)
        return response.policy?.arn.toString()
    }
}

suspend fun createRole(
    rolenameVal: String?,
    fileLocation: String?,
): String? {
    val jsonObject = fileLocation?.let { readJsonSimpleDemo(it) } as JSONObject

    val request =
        CreateRoleRequest {
            roleName = rolenameVal
            assumeRolePolicyDocument = jsonObject.toJSONString()
```

```
            description = "Created using the AWS SDK for Kotlin"
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createRole(request)
        return response.role?.arn
    }
}

suspend fun attachRolePolicy(
    roleNameVal: String,
    policyArnVal: String,
) {
    val request =
        ListAttachedRolePoliciesRequest {
            roleName = roleNameVal
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listAttachedRolePolicies(request)
        val attachedPolicies = response.attachedPolicies

        // Ensure that the policy is not attached to this role.
        val checkStatus: Int
        if (attachedPolicies != null) {
            checkStatus = checkMyList(attachedPolicies, policyArnVal)
            if (checkStatus == -1) {
                return
            }
        }

        val policyRequest =
            AttachRolePolicyRequest {
                roleName = roleNameVal
                policyArn = policyArnVal
            }
        iamClient.attachRolePolicy(policyRequest)
        println("Successfully attached policy $policyArnVal to role
 $roleNameVal")
    }
}

fun checkMyList(
    attachedPolicies: List<AttachedPolicy>,
```

```
    policyArnVal: String,
): Int {
    for (policy in attachedPolicies) {
        val polArn = policy.policyArn.toString()

        if (polArn.compareTo(policyArnVal) == 0) {
            println("The policy is already attached to this role.")
            return -1
        }
    }
    return 0
}

suspend fun assumeGivenRole(
    roleArnVal: String?,
    roleSessionNameVal: String?,
    bucketName: String,
) {
    val stsClient = StsClient.fromEnvironment { region = "us-east-1" }
    val roleRequest =
        AssumeRoleRequest {
            roleArn = roleArnVal
            roleSessionName = roleSessionNameVal
        }

    val roleResponse = stsClient.assumeRole(roleRequest)
    val myCreds = roleResponse.credentials
    val key = myCreds?.accessKeyId
    val secKey = myCreds?.secretAccessKey
    val secToken = myCreds?.sessionToken

    val staticCredentials = StaticCredentialsProvider {
        accessKeyId = key
        secretAccessKey = secKey
        sessionToken = secToken
    }

    // List all objects in an Amazon S3 bucket using the temp creds.
    val s3 = S3Client.fromEnvironment {
        region = "us-east-1"
        credentialsProvider = staticCredentials
    }

    println("Created a S3Client using temp credentials.")
```

```
    println("Listing objects in $bucketName")

    val listObjects =
        ListObjectsRequest {
            bucket = bucketName
        }

    val response = s3.listObjects(listObjects)
    response.contents?.forEach { myObject ->
        println("The name of the key is ${myObject.key}")
        println("The owner is ${myObject.owner}")
    }
}

suspend fun deleteRole(
    roleNameVal: String,
    polArn: String,
) {
    val iam = IamClient.fromEnvironment { region = "AWS_GLOBAL" }

    // First the policy needs to be detached.
    val rolePolicyRequest =
        DetachRolePolicyRequest {
            policyArn = polArn
            roleName = roleNameVal
        }

    iam.detachRolePolicy(rolePolicyRequest)

    // Delete the policy.
    val request =
        DeletePolicyRequest {
            policyArn = polArn
        }

    iam.deletePolicy(request)
    println("*** Successfully deleted $polArn")

    // Delete the role.
    val roleRequest =
        DeleteRoleRequest {
            roleName = roleNameVal
        }
```

```
    iam.deleteRole(roleRequest)
    println("*** Successfully deleted $roleNameVal")
}

suspend fun deleteUser(userNameVal: String) {
    val iam = IamClient.fromEnvironment { region = "AWS_GLOBAL" }
    val request =
        DeleteUserRequest {
            userName = userNameVal
        }

    iam.deleteUser(request)
    println("*** Successfully deleted $userNameVal")
}

@Throws(java.lang.Exception::class)
fun readJsonSimpleDemo(filename: String): Any? {
    val reader = FileReader(filename)
    val jsonParser = JSONParser()
    return jsonParser.parse(reader)
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Kotlin.
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

PHP

SDK per PHP

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
namespace Iam\Basics;

require 'vendor/autoload.php';

use Aws\Credentials\Credentials;
use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;
use Aws\Sts\StsClient;
use Iam\IAMService;

echo("\n");
echo("---------------------------------------\n");
print("Welcome to the IAM getting started demo using PHP!\n");
echo("---------------------------------------\n");

$uuid = uniqid();
$service = new IAMService();

$user = $service->createUser("iam_demo_user_$uuid");
echo "Created user with the arn: {$user['Arn']}\n";

$key = $service->createAccessKey($user['UserName']);
$assumeRolePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{$user['Arn']}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }";
$assumeRoleRole = $service->createRole("iam_demo_role_$uuid",
 $assumeRolePolicyDocument);
```

```
echo "Created role: {$assumeRoleRole['RoleName']}\n";

$listAllBucketsPolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
}";
$listAllBucketsPolicy = $service->createPolicy("iam_demo_policy_$uuid",
 $listAllBucketsPolicyDocument);
echo "Created policy: {$listAllBucketsPolicy['PolicyName']}\n";

$service->attachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);

$inlinePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"{$assumeRoleRole['Arn']}\"}]
}";
$inlinePolicy = $service->createUserPolicy("iam_demo_inline_policy_$uuid",
 $inlinePolicyDocument, $user['UserName']);
//First, fail to list the buckets with the user
$credentials = new Credentials($key['AccessKeyId'], $key['SecretAccessKey']);
$s3Client = new S3Client(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $credentials]);
try {
    $s3Client->listBuckets([
    ]);
    echo "this should not run";
} catch (S3Exception $exception) {
    echo "successfully failed!\n";
}

$stsClient = new StsClient(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $credentials]);
sleep(10);
$assumedRole = $stsClient->assumeRole([
    'RoleArn' => $assumeRoleRole['Arn'],
    'RoleSessionName' => "DemoAssumeRoleSession_$uuid",
]);
```

```
$assumedCredentials = [
    'key' => $assumedRole['Credentials']['AccessKeyId'],
    'secret' => $assumedRole['Credentials']['SecretAccessKey'],
    'token' => $assumedRole['Credentials']['SessionToken'],
];
$s3Client = new S3Client(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $assumedCredentials]);
try {
    $s3Client->listBuckets([]);
    echo "this should now run!\n";
} catch (S3Exception $exception) {
    echo "this should now not fail\n";
}

$service->detachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);
$deletePolicy = $service->deletePolicy($listAllBucketsPolicy['Arn']);
echo "Delete policy: {$listAllBucketsPolicy['PolicyName']}\n";
$deletedRole = $service->deleteRole($assumeRoleRole['Arn']);
echo "Deleted role: {$assumeRoleRole['RoleName']}\n";
$deletedKey = $service->deleteAccessKey($key['AccessKeyId'], $user['UserName']);
$deletedUser = $service->deleteUser($user['UserName']);
echo "Delete user: {$user['UserName']}\n";
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per PHP .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy

- PutUserPolicy

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
import json
import sys
import time
from uuid import uuid4

import boto3
from botocore.exceptions import ClientError


def progress_bar(seconds):
    """Shows a simple progress bar in the command window."""
    for _ in range(seconds):
        time.sleep(1)
        print(".", end="")
        sys.stdout.flush()
    print()


def setup(iam_resource):
    """
    Creates a new user with no permissions.
    Creates an access key pair for the user.
    Creates a role with a policy that lets the user assume the role.
    Creates a policy that allows listing Amazon S3 buckets.
    Attaches the policy to the role.
```

```
    Creates an inline policy for the user that lets the user assume the role.

    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
resource
                         that has permissions to create users, roles, and
policies
                         in the account.
    :return: The newly created user, user key, and role.
    """
    try:
        user = iam_resource.create_user(UserName=f"demo-user-{uuid4()}")
        print(f"Created user {user.name}.")
    except ClientError as error:
        print(
            f"Couldn't create a user for the demo. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        user_key = user.create_access_key_pair()
        print(f"Created access key pair for user.")
    except ClientError as error:
        print(
            f"Couldn't create access keys for user {user.name}. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    print(f"Wait for user to be ready.", end="")
    progress_bar(10)

    try:
        role = iam_resource.create_role(
            RoleName=f"demo-role-{uuid4()}",
            AssumeRolePolicyDocument=json.dumps(
                {
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {"AWS": user.arn},
                            "Action": "sts:AssumeRole",
                        }
```

```
                    ],
                }
            ),
        )
        print(f"Created role {role.name}.")
    except ClientError as error:
        print(
            f"Couldn't create a role for the demo. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        policy = iam_resource.create_policy(
            PolicyName=f"demo-policy-{uuid4()}",
            PolicyDocument=json.dumps(
                {
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "s3:ListAllMyBuckets",
                            "Resource": "arn:aws:s3:::*",
                        }
                    ],
                }
            ),
        )
        role.attach_policy(PolicyArn=policy.arn)
        print(f"Created policy {policy.policy_name} and attached it to the
role.")
    except ClientError as error:
        print(
            f"Couldn't create a policy and attach it to role {role.name}. Here's
why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        user.create_policy(
            PolicyName=f"demo-user-policy-{uuid4()}",
            PolicyDocument=json.dumps(
                {
```

```
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "sts:AssumeRole",
                            "Resource": role.arn,
                        }
                    ],
                }
            ),
        )
        print(
            f"Created an inline policy for {user.name} that lets the user assume
 "
            f"the role."
        )
    except ClientError as error:
        print(
            f"Couldn't create an inline policy for user {user.name}. Here's why:
 "
            f"{error.response['Error']['Message']}"
        )
        raise

    print("Give AWS time to propagate these new resources and connections.",
 end="")
    progress_bar(10)

    return user, user_key, role


def show_access_denied_without_role(user_key):
    """
    Shows that listing buckets without first assuming the role is not allowed.

    :param user_key: The key of the user created during setup. This user does not
                     have permission to list buckets in the account.
    """
    print(f"Try to list buckets without first assuming the role.")
    s3_denied_resource = boto3.resource(
        "s3", aws_access_key_id=user_key.id,
 aws_secret_access_key=user_key.secret
    )
    try:
```

```
        for bucket in s3_denied_resource.buckets.all():
            print(bucket.name)
        raise RuntimeError("Expected to get AccessDenied error when listing
 buckets!")
    except ClientError as error:
        if error.response["Error"]["Code"] == "AccessDenied":
            print("Attempt to list buckets with no permissions: AccessDenied.")
        else:
            raise


def list_buckets_from_assumed_role(user_key, assume_role_arn, session_name):
    """
    Assumes a role that grants permission to list the Amazon S3 buckets in the
 account.
    Uses the temporary credentials from the role to list the buckets that are
 owned
    by the assumed role's account.

    :param user_key: The access key of a user that has permission to assume the
 role.
    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    """
    sts_client = boto3.client(
        "sts", aws_access_key_id=user_key.id,
 aws_secret_access_key=user_key.secret
    )
    try:
        response = sts_client.assume_role(
            RoleArn=assume_role_arn, RoleSessionName=session_name
        )
        temp_credentials = response["Credentials"]
        print(f"Assumed role {assume_role_arn} and got temporary credentials.")
    except ClientError as error:
        print(
            f"Couldn't assume role {assume_role_arn}. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    # Create an S3 resource that can access the account with the temporary
 credentials.
```

```
    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )
    print(f"Listing buckets for the assumed role's account:")
    try:
        for bucket in s3_resource.buckets.all():
            print(bucket.name)
    except ClientError as error:
        print(
            f"Couldn't list buckets for the account. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise




def teardown(user, role):
    """
    Removes all resources created during setup.

    :param user: The demo user.
    :param role: The demo role.
    """
    try:
        for attached in role.attached_policies.all():
            policy_name = attached.policy_name
            role.detach_policy(PolicyArn=attached.arn)
            attached.delete()
            print(f"Detached and deleted {policy_name}.")
        role.delete()
        print(f"Deleted {role.name}.")
    except ClientError as error:
        print(
            "Couldn't detach policy, delete policy, or delete role. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        for user_pol in user.policies.all():
```

```
            user_pol.delete()
            print("Deleted inline user policy.")
        for key in user.access_keys.all():
            key.delete()
            print("Deleted user's access key.")
        user.delete()
        print(f"Deleted {user.name}.")
    except ClientError as error:
        print(
            "Couldn't delete user policy or delete user. Here's why: "
            f"{error.response['Error']['Message']}"
        )


def usage_demo():
    """Drives the demonstration."""
    print("-" * 88)
    print(f"Welcome to the IAM create user and assume role demo.")
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    user = None
    role = None
    try:
        user, user_key, role = setup(iam_resource)
        print(f"Created {user.name} and {role.name}.")
        show_access_denied_without_role(user_key)
        list_buckets_from_assumed_role(user_key, role.arn,
 "AssumeRoleDemoSession")
    except Exception:
        print("Something went wrong!")
    finally:
        if user is not None and role is not None:
            teardown(user, role)
        print("Thanks for watching!")


if __name__ == "__main__":
    usage_demo()
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - AttachRolePolicy

- CreateAccessKey
- CreatePolicy
- CreateRole
- CreateUser
- DeleteAccessKey
- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

Ruby

SDK per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
# Wraps the scenario actions.
class ScenarioCreateUserAssumeRole
  attr_reader :iam_client

  # @param [Aws::IAM::Client] iam_client: The AWS IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Waits for the specified number of seconds.
```

```
  #
  # @param duration [Integer] The number of seconds to wait.
  def wait(duration)
    puts('Give AWS time to propagate resources...')
    sleep(duration)
  end

  # Creates a user.
  #
  # @param user_name [String] The name to give the user.
  # @return [Aws::IAM::User] The newly created user.
  def create_user(user_name)
    user = @iam_client.create_user(user_name: user_name).user
    @logger.info("Created demo user named #{user.user_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info('Tried and failed to create demo user.')
    @logger.info("\t#{e.code}: #{e.message}")
    @logger.info("\nCan't continue the demo without a user!")
    raise
  else
    user
  end

  # Creates an access key for a user.
  #
  # @param user [Aws::IAM::User] The user that owns the key.
  # @return [Aws::IAM::AccessKeyPair] The newly created access key.
  def create_access_key_pair(user)
    user_key = @iam_client.create_access_key(user_name:
user.user_name).access_key
    @logger.info("Created accesskey pair for user #{user.user_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create access keys for user #{user.user_name}.")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  else
    user_key
  end

  # Creates a role that can be assumed by a user.
  #
  # @param role_name [String] The name to give the role.
  # @param user [Aws::IAM::User] The user who is granted permission to assume the
role.
```

```
  # @return [Aws::IAM::Role] The newly created role.
  def create_role(role_name, user)
    trust_policy = {
      Version: '2012-10-17',
      Statement: [{
        Effect: 'Allow',
        Principal: { 'AWS': user.arn },
        Action: 'sts:AssumeRole'
      }]
    }.to_json
    role = @iam_client.create_role(
      role_name: role_name,
      assume_role_policy_document: trust_policy
    ).role
    @logger.info("Created role #{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create a role for the demo. Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  else
    role
  end

  # Creates a policy that grants permission to list S3 buckets in the account,
and
  # then attaches the policy to a role.
  #
  # @param policy_name [String] The name to give the policy.
  # @param role [Aws::IAM::Role] The role that the policy is attached to.
  # @return [Aws::IAM::Policy] The newly created policy.
  def create_and_attach_role_policy(policy_name, role)
    policy_document = {
      Version: '2012-10-17',
      Statement: [{
        Effect: 'Allow',
        Action: 's3:ListAllMyBuckets',
        Resource: 'arn:aws:s3:::*'
      }]
    }.to_json
    policy = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document
    ).policy
    @iam_client.attach_role_policy(
```

```
      role_name: role.role_name,
      policy_arn: policy.arn
    )
    @logger.info("Created policy #{policy.policy_name} and attached it to role
#{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create a policy and attach it to role
#{role.role_name}. Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  end

  # Creates an inline policy for a user that lets the user assume a role.
  #
  # @param policy_name [String] The name to give the policy.
  # @param user [Aws::IAM::User] The user that owns the policy.
  # @param role [Aws::IAM::Role] The role that can be assumed.
  # @return [Aws::IAM::UserPolicy] The newly created policy.
  def create_user_policy(policy_name, user, role)
    policy_document = {
      Version: '2012-10-17',
      Statement: [{
        Effect: 'Allow',
        Action: 'sts:AssumeRole',
        Resource: role.arn
      }]
    }.to_json
    @iam_client.put_user_policy(
      user_name: user.user_name,
      policy_name: policy_name,
      policy_document: policy_document
    )
    puts("Created an inline policy for #{user.user_name} that lets the user
assume role #{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create an inline policy for user #{user.user_name}.
Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  end

  # Creates an Amazon S3 resource with specified credentials. This is separated
into a
  # factory function so that it can be mocked for unit testing.
```

```
 #
 # @param credentials [Aws::Credentials] The credentials used by the Amazon S3
resource.
 def create_s3_resource(credentials)
   Aws::S3::Resource.new(client: Aws::S3::Client.new(credentials: credentials))
 end

 # Lists the S3 buckets for the account, using the specified Amazon S3 resource.
 # Because the resource uses credentials with limited access, it may not be able
to
 # list the S3 buckets.
 #
 # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
 def list_buckets(s3_resource)
   count = 10
   s3_resource.buckets.each do |bucket|
     @logger.info "\t#{bucket.name}"
     count -= 1
     break if count.zero?
   end
 rescue Aws::Errors::ServiceError => e
   if e.code == 'AccessDenied'
     puts('Attempt to list buckets with no permissions: AccessDenied.')
   else
     @logger.info("Couldn't list buckets for the account. Here's why: ")
     @logger.info("\t#{e.code}: #{e.message}")
     raise
   end
 end

 # Creates an AWS Security Token Service (AWS STS) client with specified
credentials.
 # This is separated into a factory function so that it can be mocked for unit
testing.
 #
 # @param key_id [String] The ID of the access key used by the STS client.
 # @param key_secret [String] The secret part of the access key used by the STS
client.
 def create_sts_client(key_id, key_secret)
   Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
 end

 # Gets temporary credentials that can be used to assume a role.
 #
```

```
 # @param role_arn [String] The ARN of the role that is assumed when these
credentials
 #                          are used.
 # @param sts_client [AWS::STS::Client] An AWS STS client.
 # @return [Aws::AssumeRoleCredentials] The credentials that can be used to
assume the role.
 def assume_role(role_arn, sts_client)
   credentials = Aws::AssumeRoleCredentials.new(
     client: sts_client,
     role_arn: role_arn,
     role_session_name: 'create-use-assume-role-scenario'
   )
   @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
   credentials
 end

 # Deletes a role. If the role has policies attached, they are detached and
 # deleted before the role is deleted.
 #
 # @param role_name [String] The name of the role to delete.
 def delete_role(role_name)
   @iam_client.list_attached_role_policies(role_name:
role_name).attached_policies.each do |policy|
     @iam_client.detach_role_policy(role_name: role_name, policy_arn:
policy.policy_arn)
     @iam_client.delete_policy(policy_arn: policy.policy_arn)
     @logger.info("Detached and deleted policy #{policy.policy_name}.")
   end
   @iam_client.delete_role({ role_name: role_name })
   @logger.info("Role deleted: #{role_name}.")
 rescue Aws::Errors::ServiceError => e
   @logger.info("Couldn't detach policies and delete role #{role.name}. Here's
why:")
   @logger.info("\t#{e.code}: #{e.message}")
   raise
 end

 # Deletes a user. If the user has inline policies or access keys, they are
deleted
 # before the user is deleted.
 #
 # @param user [Aws::IAM::User] The user to delete.
 def delete_user(user_name)
   user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
```

```
    user.each do |key|
      @iam_client.delete_access_key({ access_key_id: key.access_key_id,
 user_name: user_name })
      @logger.info("Deleted access key #{key.access_key_id} for user
 '#{user_name}'.")
    end

    @iam_client.delete_user(user_name: user_name)
    @logger.info("Deleted user '#{user_name}'.")
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting user '#{user_name}': #{e.message}")
  end
end

# Runs the IAM create a user and assume a role scenario.
def run_scenario(scenario)
  puts('-' * 88)
  puts('Welcome to the IAM create a user and assume a role demo!')
  puts('-' * 88)
  user = scenario.create_user("doc-example-user-#{Random.uuid}")
  user_key = scenario.create_access_key_pair(user)
  scenario.wait(10)
  role = scenario.create_role("doc-example-role-#{Random.uuid}", user)
  scenario.create_and_attach_role_policy("doc-example-role-policy-
#{Random.uuid}", role)
  scenario.create_user_policy("doc-example-user-policy-#{Random.uuid}", user,
 role)
  scenario.wait(10)
  puts('Try to list buckets with credentials for a user who has no permissions.')
  puts('Expect AccessDenied from this call.')
  scenario.list_buckets(
    scenario.create_s3_resource(Aws::Credentials.new(user_key.access_key_id,
 user_key.secret_access_key))
  )
  puts('Now, assume the role that grants permission.')
  temp_credentials = scenario.assume_role(
    role.arn, scenario.create_sts_client(user_key.access_key_id,
 user_key.secret_access_key)
  )
  puts('Here are your buckets:')
  scenario.list_buckets(scenario.create_s3_resource(temp_credentials))
  puts("Deleting role '#{role.role_name}' and attached policies.")
  scenario.delete_role(role.role_name)
  puts("Deleting user '#{user.user_name}', policies, and keys.")
```

```
  scenario.delete_user(user.user_name)
  puts('Thanks for watching!')
  puts('-' * 88)
rescue Aws::Errors::ServiceError => e
  puts('Something went wrong with the demo.')
  puts("\t#{e.code}: #{e.message}")
end

run_scenario(ScenarioCreateUserAssumeRole.new(Aws::IAM::Client.new)) if
 $PROGRAM_NAME == __FILE__
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Ruby .
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
use aws_config::meta::region::RegionProviderChain;
use aws_sdk_iam::Error as iamError;
use aws_sdk_iam::{config::Credentials as iamCredentials, config::Region, Client
 as iamClient};
use aws_sdk_s3::Client as s3Client;
use aws_sdk_sts::Client as stsClient;
use tokio::time::{sleep, Duration};
use uuid::Uuid;

#[tokio::main]
async fn main() -> Result<(), iamError> {
    let (client, uuid, list_all_buckets_policy_document, inline_policy_document)
 =
        initialize_variables().await;

    if let Err(e) = run_iam_operations(
        client,
        uuid,
        list_all_buckets_policy_document,
        inline_policy_document,
    )
    .await
    {
        println!("{:?}", e);
    };

    Ok(())
}

async fn initialize_variables() -> (iamClient, String, String, String) {
    let region_provider = RegionProviderChain::first_try(Region::new("us-
west-2"));

    let shared_config =
 aws_config::from_env().region(region_provider).load().await;
    let client = iamClient::new(&shared_config);
    let uuid = Uuid::new_v4().to_string();

    let list_all_buckets_policy_document = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
```

```
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
    }"
    .to_string();
    let inline_policy_document = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"{}\"}]
    }"
    .to_string();

    (
        client,
        uuid,
        list_all_buckets_policy_document,
        inline_policy_document,
    )
}

async fn run_iam_operations(
    client: iamClient,
    uuid: String,
    list_all_buckets_policy_document: String,
    inline_policy_document: String,
) -> Result<(), iamError> {
    let user = iam_service::create_user(&client, &format!("{}{}",
 "iam_demo_user_", uuid)).await?;
    println!("Created the user with the name: {}", user.user_name());
    let key = iam_service::create_access_key(&client, user.user_name()).await?;

    let assume_role_policy_document = "{
        \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }"
    .to_string()
    .replace("{}", user.arn());

    let assume_role_role = iam_service::create_role(
```

```
        &client,
        &format!("{}{}", "iam_demo_role_", uuid),
        &assume_role_policy_document,
    )
    .await?;
    println!("Created the role with the ARN: {}", assume_role_role.arn());

    let list_all_buckets_policy = iam_service::create_policy(
        &client,
        &format!("{}{}", "iam_demo_policy_", uuid),
        &list_all_buckets_policy_document,
    )
    .await?;
    println!(
        "Created policy: {}",
        list_all_buckets_policy.policy_name.as_ref().unwrap()
    );

    let attach_role_policy_result =
        iam_service::attach_role_policy(&client, &assume_role_role,
&list_all_buckets_policy)
            .await?;
    println!(
        "Attached the policy to the role: {:?}",
        attach_role_policy_result
    );

    let inline_policy_name = format!("{}{}", "iam_demo_inline_policy_", uuid);
    let inline_policy_document = inline_policy_document.replace("{}",
assume_role_role.arn());
    iam_service::create_user_policy(&client, &user, &inline_policy_name,
&inline_policy_document)
        .await?;
    println!("Created inline policy.");

    //First, fail to list the buckets with the user.
    let creds = iamCredentials::from_keys(key.access_key_id(),
key.secret_access_key(), None);
    let fail_config = aws_config::from_env()
        .credentials_provider(creds.clone())
        .load()
        .await;
    println!("Fail config: {:?}", fail_config);
    let fail_client: s3Client = s3Client::new(&fail_config);
```

```
    match fail_client.list_buckets().send().await {
        Ok(e) => {
            println!("This should not run. {:?}", e);
        }
        Err(e) => {
            println!("Successfully failed with error: {:?}", e)
        }
    }

    let sts_config = aws_config::from_env()
        .credentials_provider(creds.clone())
        .load()
        .await;
    let sts_client: stsClient = stsClient::new(&sts_config);
    sleep(Duration::from_secs(10)).await;
    let assumed_role = sts_client
        .assume_role()
        .role_arn(assume_role_role.arn())
        .role_session_name(format!("iam_demo_assumerole_session_{uuid}"))
        .send()
        .await;
    println!("Assumed role: {:?}", assumed_role);
    sleep(Duration::from_secs(10)).await;

    let assumed_credentials = iamCredentials::from_keys(
        assumed_role
            .as_ref()
            .unwrap()
            .credentials
            .as_ref()
            .unwrap()
            .access_key_id(),
        assumed_role
            .as_ref()
            .unwrap()
            .credentials
            .as_ref()
            .unwrap()
            .secret_access_key(),
        Some(
            assumed_role
                .as_ref()
                .unwrap()
                .credentials
```

```
                .as_ref()
                .unwrap()
                .session_token
                .clone(),
        ),
    );

    let succeed_config = aws_config::from_env()
        .credentials_provider(assumed_credentials)
        .load()
        .await;
    println!("succeed config: {:?}", succeed_config);
    let succeed_client: s3Client = s3Client::new(&succeed_config);
    sleep(Duration::from_secs(10)).await;
    match succeed_client.list_buckets().send().await {
        Ok(_) => {
            println!("This should now run successfully.")
        }
        Err(e) => {
            println!("This should not run. {:?}", e);
            panic!()
        }
    }

    //Clean up.
    iam_service::detach_role_policy(
        &client,
        assume_role_role.role_name(),
        list_all_buckets_policy.arn().unwrap_or_default(),
    )
    .await?;
    iam_service::delete_policy(&client, list_all_buckets_policy).await?;
    iam_service::delete_role(&client, &assume_role_role).await?;
    println!("Deleted role {}", assume_role_role.role_name());
    iam_service::delete_access_key(&client, &user, &key).await?;
    println!("Deleted key for {}", key.user_name());
    iam_service::delete_user_policy(&client, &user, &inline_policy_name).await?;
    println!("Deleted inline user policy: {}", inline_policy_name);
    iam_service::delete_user(&client, &user).await?;
    println!("Deleted user {}", user.user_name());

    Ok(())
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API SDK AWS per Rust.
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Azioni per l'utilizzo di IAM AWS SDK

I seguenti esempi di codice mostrano come eseguire singole azioni IAM con gli AWS SDK. Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Questi estratti chiamano l'API IAM e sono estratti di codice da programmi più grandi che devono essere eseguiti in modo contestuale. È possibile visualizzare le azioni nel contesto in Scenari per l'utilizzo di IAM AWS SDK.

Gli esempi seguenti includono solo le azioni più comunemente utilizzate. Per un elenco completo, consulta la documentazione di riferimento dell'API AWS Identity and Access Management.

Esempi

- Utilizzo di AddClientIdToOpenIdConnectProvider con una CLI

- Utilizzo di AddRoleToInstanceProfile con una CLI
- Utilizzo di AddUserToGroup con una CLI
- Utilizzo di AttachGroupPolicy con una CLI
- Usare AttachRolePolicycon un AWS SDK o CLI
- Usare AttachUserPolicycon un AWS SDK o CLI
- Utilizzo di ChangePassword con una CLI
- Utilizzare CreateAccessKeycon un AWS SDK o CLI
- Usare CreateAccountAliascon un AWS SDK o CLI
- Usare CreateGroupcon un AWS SDK o CLI
- Usare CreateInstanceProfilecon un AWS SDK o CLI
- Utilizzo di CreateLoginProfile con una CLI
- Utilizzo di CreateOpenIdConnectProvider con una CLI
- Utilizzare CreatePolicycon un AWS SDK o CLI
- Usare CreatePolicyVersioncon un AWS SDK o CLI
- Usare CreateRolecon un AWS SDK o CLI
- Usa CreateSAMLProvider con un AWS SDK o CLI
- Utilizzare CreateServiceLinkedRolecon un AWS SDK o CLI
- Utilizzare CreateUsercon un AWS SDK o CLI
- Utilizzo di CreateVirtualMfaDevice con una CLI
- Utilizzo di DeactivateMfaDevice con una CLI
- Usare DeleteAccessKeycon un AWS SDK o CLI
- Usare DeleteAccountAliascon un AWS SDK o CLI
- Utilizzo di DeleteAccountPasswordPolicy con una CLI
- Usare DeleteGroupcon un AWS SDK o CLI
- Utilizzo di DeleteGroupPolicy con una CLI
- Usare DeleteInstanceProfilecon un AWS SDK o CLI
- Utilizzo di DeleteLoginProfile con una CLI
- Utilizzo di DeleteOpenIdConnectProvider con una CLI

- Usare DeletePolicycon un AWS SDK o CLI
- Usare DeletePolicyVersioncon un AWS SDK o CLI
- Usare DeleteRolecon un AWS SDK o CLI
- Utilizzo di DeleteRolePermissionsBoundary con una CLI
- Usare DeleteRolePolicycon un AWS SDK o CLI
- Usa DeleteSamlProvider con un AWS SDK o CLI
- Usare DeleteServerCertificatecon un AWS SDK o CLI
- Usare DeleteServiceLinkedRolecon un AWS SDK o CLI
- Utilizzo di DeleteSigningCertificate con una CLI
- Utilizzare DeleteUsercon un AWS SDK o CLI
- Utilizzo di DeleteUserPermissionsBoundary con una CLI
- Usare DeleteUserPolicycon un AWS SDK o CLI
- Utilizzo di DeleteVirtualMfaDevice con una CLI
- Utilizzo di DetachGroupPolicy con una CLI
- Utilizzare DetachRolePolicycon un AWS SDK o CLI
- Usare DetachUserPolicycon un AWS SDK o CLI
- Utilizzo di EnableMfaDevice con una CLI
- Usare GenerateCredentialReportcon un AWS SDK o CLI
- Utilizzo di GenerateServiceLastAccessedDetails con una CLI
- Usare GetAccessKeyLastUsedcon un AWS SDK o CLI
- Utilizzare GetAccountAuthorizationDetailscon un AWS SDK o CLI
- Usare GetAccountPasswordPolicycon un AWS SDK o CLI
- Utilizzare GetAccountSummarycon un AWS SDK o CLI
- Utilizzo di GetContextKeysForCustomPolicy con una CLI
- Utilizzo di GetContextKeysForPrincipalPolicy con una CLI
- Usare GetCredentialReportcon un AWS SDK o CLI
- Utilizzo di GetGroup con una CLI
- Utilizzo di GetGroupPolicy con una CLI

- Utilizzo di GetInstanceProfile con una CLI
- Utilizzo di GetLoginProfile con una CLI
- Utilizzo di GetOpenIdConnectProvider con una CLI
- Usare GetPolicycon un AWS SDK o CLI
- Usare GetPolicyVersioncon un AWS SDK o CLI
- Utilizzare GetRolecon un AWS SDK o CLI
- Utilizzo di GetRolePolicy con una CLI
- Utilizzo di GetSamlProvider con una CLI
- Usare GetServerCertificatecon un AWS SDK o CLI
- Utilizzo di GetServiceLastAccessedDetails con una CLI
- Utilizzo di GetServiceLastAccessedDetailsWithEntities con una CLI
- Utilizzare GetServiceLinkedRoleDeletionStatuscon un AWS SDK o CLI
- Usare GetUsercon un AWS SDK o CLI
- Utilizzo di GetUserPolicy con una CLI
- Usare ListAccessKeyscon un AWS SDK o CLI
- Utilizzare ListAccountAliasescon un AWS SDK o CLI
- Utilizzo di ListAttachedGroupPolicies con una CLI
- Utilizzare ListAttachedRolePoliciescon un AWS SDK o CLI
- Utilizzo di ListAttachedUserPolicies con una CLI
- Utilizzo di ListEntitiesForPolicy con una CLI
- Utilizzo di ListGroupPolicies con una CLI
- Utilizzare ListGroupscon un AWS SDK o CLI
- Utilizzo di ListGroupsForUser con una CLI
- Utilizzo di ListInstanceProfiles con una CLI
- Utilizzo di ListInstanceProfilesForRole con una CLI
- Utilizzo di ListMfaDevices con una CLI
- Utilizzo di ListOpenIdConnectProviders con una CLI
- Utilizzare ListPoliciescon un AWS SDK o CLI

- Usare ListPolicyVersionscon un AWS SDK o CLI
- Utilizzare ListRolePoliciescon un AWS SDK o CLI
- Utilizzo di ListRoleTags con una CLI
- Utilizzare ListRolescon un AWS SDK o CLI
- Usa ListSamlProviders con un AWS SDK o CLI
- Utilizzare ListServerCertificatescon un AWS SDK o CLI
- Utilizzo di ListSigningCertificates con una CLI
- Usare ListUserPoliciescon un AWS SDK o CLI
- Utilizzo di ListUserTags con una CLI
- Usare ListUserscon un AWS SDK o CLI
- Utilizzo di ListVirtualMfaDevices con una CLI
- Utilizzo di PutGroupPolicy con una CLI
- Utilizzo di PutRolePermissionsBoundary con una CLI
- Utilizzare PutRolePolicycon un AWS SDK o CLI
- Utilizzo di PutUserPermissionsBoundary con una CLI
- Usare PutUserPolicycon un AWS SDK o CLI
- Utilizzo di RemoveClientIdFromOpenIdConnectProvider con una CLI
- Utilizzo di RemoveRoleFromInstanceProfile con una CLI
- Utilizzo di RemoveUserFromGroup con una CLI
- Utilizzo di ResyncMfaDevice con una CLI
- Usare SetDefaultPolicyVersioncon un AWS SDK o CLI
- Utilizzo di TagRole con una CLI
- Utilizzo di TagUser con una CLI
- Utilizzo di UntagRole con una CLI
- Utilizzo di UntagUser con una CLI
- Utilizzare UpdateAccessKeycon un AWS SDK o CLI
- Utilizzo di UpdateAccountPasswordPolicy con una CLI
- Utilizzo di UpdateAssumeRolePolicy con una CLI
- Utilizzo di UpdateGroup con una CLI

- Utilizzo di UpdateLoginProfile con una CLI
- Utilizzo di UpdateOpenIdConnectProviderThumbprint con una CLI
- Utilizzo di UpdateRole con una CLI
- Utilizzo di UpdateRoleDescription con una CLI
- Utilizzo di UpdateSamlProvider con una CLI
- Usare UpdateServerCertificatecon un AWS SDK o CLI
- Utilizzo di UpdateSigningCertificate con una CLI
- Usare UpdateUsercon un AWS SDK o CLI
- Utilizzare UploadServerCertificatecon un AWS SDK o CLI
- Utilizzo di UploadSigningCertificate con una CLI

Utilizzo di **`AddClientIdToOpenIdConnectProvider`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`AddClientIdToOpenIdConnectProvider`.

CLI

AWS CLI

Per aggiungere un ID client (audience) a un provider Open-ID Connect (OIDC)

Il comando `add-client-id-to-open-id-connect-provider` seguente aggiunge l'ID client `my-application-ID` al provider OIDC denominato `server.example.com`.

```
aws iam add-client-id-to-open-id-connect-provider \
    --client-id my-application-ID \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

Questo comando non produce alcun output.

Come creare un provider OIDC, utilizzare il comando `create-open-id-connect-provider`.

Per ulteriori informazioni, consulta Creazione di provider di identità OpenID Connect (OIDC) nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta AWS CLI Command AddClientIdToOpenIdConnectProviderReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo comando aggiunge l'ID client (o il pubblico) **`my-application-ID`** al provider OIDC esistente denominato **`server.example.com`**.

```
Add-IAMClientIDToOpenIDConnectProvider -ClientID "my-application-ID"
 -OpenIDConnectProviderARN "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
```

- Per i dettagli sull'API, vedere AddClientIdToOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando aggiunge l'ID client (o il pubblico) **`my-application-ID`** al provider OIDC esistente denominato **`server.example.com`**.

```
Add-IAMClientIDToOpenIDConnectProvider -ClientID "my-application-ID"
 -OpenIDConnectProviderARN "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
```

- Per i dettagli sull'API, vedere AddClientIdToOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`AddRoleToInstanceProfile`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`AddRoleToInstanceProfile`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un servizio contenitore per le istanze di macchine virtuali
- Guida introduttiva allo streaming gestito
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

CLI

AWS CLI

Come aggiungere un ruolo a un profilo dell'istanza

Il comando `add-role-to-instance-profile` seguente aggiunge il ruolo denominato `S3Access` al profilo dell'istanza denominato `Webserver`.

```
aws iam add-role-to-instance-profile \
    --role-name S3Access \
    --instance-profile-name Webserver
```

Questo comando non produce alcun output.

Per creare un profilo dell'istanza, utilizza il comando `create-instance-profile`.

Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AddRoleToInstanceProfile AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo comando aggiunge il ruolo denominato **`S3Access`** a un profilo dell'istanza esistente denominato **`webserver`**. Per creare il profilo dell'istanza, utilizza il comando **`New-IAMInstanceProfile`**. Dopo aver creato il profilo dell'istanza e averlo associato a un ruolo utilizzando questo comando, potrai collegarlo a un'istanza EC2. A tale scopo, utilizza il cmdlet **`New-EC2Instance`** con il parametro **`InstanceProfile_Arn`** o con il parametro **`InstanceProfile-Name`** per avviare la nuova istanza.

```
Add-IAMRoleToInstanceProfile -RoleName "S3Access" -InstanceProfileName
 "webserver"
```

- Per i dettagli sull'API, vedere AddRoleToInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando aggiunge il ruolo denominato **`S3Access`** a un profilo dell'istanza esistente denominato **`webserver`**. Per creare il profilo dell'istanza, utilizza il comando **`New-IAMInstanceProfile`**. Dopo aver creato il profilo dell'istanza e averlo associato a un ruolo utilizzando questo comando, potrai collegarlo a un'istanza EC2. A tale scopo, utilizza il cmdlet **`New-EC2Instance`** con il parametro **`InstanceProfile_Arn`** o con il parametro **`InstanceProfile-Name`** per avviare la nuova istanza.

```
Add-IAMRoleToInstanceProfile -RoleName "S3Access" -InstanceProfileName
 "webserver"
```

- Per i dettagli sull'API, vedere AddRoleToInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`AddUserToGroup`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`AddUserToGroup`.

CLI

AWS CLI

Come aggiungere un utente a un gruppo IAM

Il comando `add-user-to-group` seguente aggiunte l'utente IAM denominato `Bob` al gruppo IAM denominato `Admins`.

```
aws iam add-user-to-group \
    --user-name Bob \
    --group-name Admins
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Aggiunta e rimozione di utenti in un gruppo di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AddUserToGroup AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo comando aggiunge l'utente denominato **Bob** al gruppo denominato **Admins**.

```
Add-IAMUserToGroup -UserName "Bob" -GroupName "Admins"
```

- Per i dettagli sull'API, vedere AddUserToGroupin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando aggiunge l'utente denominato **Bob** al gruppo denominato **Admins**.

```
Add-IAMUserToGroup -UserName "Bob" -GroupName "Admins"
```

- Per i dettagli sull'API, vedere AddUserToGroupin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`AttachGroupPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`AttachGroupPolicy`.

CLI

AWS CLI

Per collegare una policy gestita a un gruppo IAM

Il `attach-group-policy` comando seguente collega la policy AWS gestita denominata `ReadOnlyAccess` al gruppo IAM denominato. `Finance`

```
aws iam attach-group-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --group-name Finance
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy gestite e policy inline nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AttachGroupPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio collega la policy gestita dal cliente denominata **`TesterPolicy`** al gruppo IAM **`Testers`**. Gli utenti di quel gruppo sono immediatamente interessati dalle autorizzazioni definite nella versione predefinita di tale policy.

```
Register-IAMGroupPolicy -GroupName Testers -PolicyArn
 arn:aws:iam::123456789012:policy/TesterPolicy
```

Esempio 2: questo esempio allega la policy AWS gestita denominata **`AdministratorAccess`** al gruppo IAM. **`Admins`** Gli utenti di quel gruppo sono immediatamente interessati dalle autorizzazioni definite nella versione predefinita di tale policy.

```
Register-IAMGroupPolicy -GroupName Admins -PolicyArn arn:aws:iam::aws:policy/
AdministratorAccess
```

- Per i dettagli sull'API, vedere AttachGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio collega la policy gestita dal cliente denominata **`TesterPolicy`** al gruppo IAM **`Testers`**. Gli utenti di quel gruppo sono immediatamente interessati dalle autorizzazioni definite nella versione predefinita di tale policy.

```
Register-IAMGroupPolicy -GroupName Testers -PolicyArn
 arn:aws:iam::123456789012:policy/TesterPolicy
```

Esempio 2: questo esempio allega la policy AWS gestita denominata **`AdministratorAccess`** al gruppo IAM. **`Admins`** Gli utenti di quel gruppo sono immediatamente interessati dalle autorizzazioni definite nella versione predefinita di tale policy.

```
Register-IAMGroupPolicy -GroupName Admins -PolicyArn arn:aws:iam::aws:policy/
AdministratorAccess
```

- Per i dettagli sull'API, vedere AttachGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`AttachRolePolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`AttachRolePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Configura la connettività dei servizi container
- Crea un'attività contenitore per il tipo di avvio serverless
- Crea un'API REST con integrazione con proxy di funzioni
- Creazione di un servizio contenitore per le istanze di macchine virtuali
- Creazione di un'area di lavoro di monitoraggio gestita
- Creazione di una dashboard di monitoraggio con il nome della funzione come variabile
- Creazione della tua prima funzione serverless
- Guida introduttiva alla gestione della configurazione
- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
- Guida introduttiva agli store di funzionalità di apprendimento automatico
- Guida introduttiva ai cluster Kubernetes gestiti

- Guida introduttiva allo streaming gestito
- Guida introduttiva all'orchestrazione del flusso di lavoro
- Gestione dei ruoli
- Spostamento di segreti codificati in uno spazio di archiviazione segreto sicuro
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori
- Configurazione della gestione dei sistemi
- Utilizzo delle variabili di proprietà nei dashboard di monitoraggio per monitorare più funzioni serverless
- Lavora con Streams e Time-to-Live

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Attach an IAM policy to a role.
    /// </summary>
    /// <param name="policyArn">The policy to attach.</param>
    /// <param name="roleName">The role that the policy will be attached to.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AttachRolePolicyAsync(string policyArn, string
 roleName)
    {
        var response = await _IAMService.AttachRolePolicyAsync(new
 AttachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
```

```
    }
```

- Per i dettagli sull'API, consulta la AttachRolePolicysezione AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
  local role_name policy_arn response
```

```
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam attach-role-policy \
    --role-name "$role_name" \
```

```
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta AttachRolePolicy AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::attachRolePolicy(const Aws::String &roleName,
                                   const Aws::String &policyArn,
                                   const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::ListAttachedRolePoliciesRequest list_request;
    list_request.SetRoleName(roleName);

    bool done = false;
    while (!done) {
        auto list_outcome = iam.ListAttachedRolePolicies(list_request);
        if (!list_outcome.IsSuccess()) {
            std::cerr << "Failed to list attached policies of role " <<
```

```
                    roleName << ": " << list_outcome.GetError().GetMessage() <<
                    std::endl;
            return false;
        }

        const auto &policies = list_outcome.GetResult().GetAttachedPolicies();
        if (std::any_of(policies.cbegin(), policies.cend(),
                        [=](const Aws::IAM::Model::AttachedPolicy &policy) {
                                return policy.GetPolicyArn() == policyArn;
                        })) {
            std::cout << "Policy " << policyArn <<
                      " is already attached to role " << roleName << std::endl;
            return true;
        }

        done = !list_outcome.GetResult().GetIsTruncated();
        list_request.SetMarker(list_outcome.GetResult().GetMarker());
    }

    Aws::IAM::Model::AttachRolePolicyRequest request;
    request.SetRoleName(roleName);
    request.SetPolicyArn(policyArn);

    Aws::IAM::Model::AttachRolePolicyOutcome outcome =
 iam.AttachRolePolicy(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Failed to attach policy " << policyArn << " to role " <<
                  roleName << ": " << outcome.GetError().GetMessage() <<
 std::endl;
    }
    else {
        std::cout << "Successfully attached policy " << policyArn << " to role "
 <<
                  roleName << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la AttachRolePolicysezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come collegare una policy gestita a un ruolo IAM

Il `attach-role-policy` comando seguente collega la policy AWS gestita denominata `ReadOnlyAccess` al ruolo IAM denominato`ReadOnlyRole`.

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --role-name ReadOnlyRole
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy gestite e policy inline nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AttachRolePolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
```

```
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// AttachRolePolicy attaches a policy to a role.
func (wrapper RoleWrapper) AttachRolePolicy(ctx context.Context, policyArn
 string, roleName string) error {
 _, err := wrapper.IamClient.AttachRolePolicy(ctx, &iam.AttachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't attach policy %v to role %v. Here's why: %v\n", policyArn,
 roleName, err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta la AttachRolePolicysezione AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.AttachRolePolicyRequest;
```

```
import software.amazon.awssdk.services.iam.model.AttachedPolicy;
import software.amazon.awssdk.services.iam.model.ListAttachedRolePoliciesRequest;
import
 software.amazon.awssdk.services.iam.model.ListAttachedRolePoliciesResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class AttachRolePolicy {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <roleName> <policyArn>\s

                Where:
                    roleName - A role name that you can obtain from the AWS
 Management Console.\s
                    policyArn - A policy ARN that you can obtain from the AWS
 Management Console.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String roleName = args[0];
        String policyArn = args[1];

        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        attachIAMRolePolicy(iam, roleName, policyArn);
        iam.close();
```

```
    }

    public static void attachIAMRolePolicy(IamClient iam, String roleName, String
 policyArn) {
        try {
            ListAttachedRolePoliciesRequest request =
 ListAttachedRolePoliciesRequest.builder()
                    .roleName(roleName)
                    .build();

            ListAttachedRolePoliciesResponse response =
 iam.listAttachedRolePolicies(request);
            List<AttachedPolicy> attachedPolicies = response.attachedPolicies();

            // Ensure that the policy is not attached to this role
            String polArn = "";
            for (AttachedPolicy policy : attachedPolicies) {
                polArn = policy.policyArn();
                if (polArn.compareTo(policyArn) == 0) {
                    System.out.println(roleName + " policy is already attached to
 this role.");
                    return;
                }
            }

            AttachRolePolicyRequest attachRequest =
 AttachRolePolicyRequest.builder()
                    .roleName(roleName)
                    .policyArn(policyArn)
                    .build();

            iam.attachRolePolicy(attachRequest);

            System.out.println("Successfully attached policy " + policyArn +
                    " to role " + roleName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

- Per i dettagli sull'API, consulta la AttachRolePolicysezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Collega la policy.

```
import { AttachRolePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} policyArn
 * @param {string} roleName
 */
export const attachRolePolicy = (policyArn, roleName) => {
  const command = new AttachRolePolicyCommand({
    PolicyArn: policyArn,
    RoleName: roleName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la AttachRolePolicysezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var paramsRoleList = {
  RoleName: process.argv[2],
};

iam.listAttachedRolePolicies(paramsRoleList, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    var myRolePolicies = data.AttachedPolicies;
    myRolePolicies.forEach(function (val, index, array) {
      if (myRolePolicies[index].PolicyName === "AmazonDynamoDBFullAccess") {
        console.log(
          "AmazonDynamoDBFullAccess is already attached to this role."
        );
        process.exit();
      }
    });
    var params = {
      PolicyArn: "arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess",
      RoleName: process.argv[2],
    };
    iam.attachRolePolicy(params, function (err, data) {
      if (err) {
        console.log("Unable to attach policy to role", err);
      } else {
        console.log("Role attached successfully");
```

```
      }
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la AttachRolePolicysezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun attachIAMRolePolicy(
    roleNameVal: String,
    policyArnVal: String,
) {
    val request =
        ListAttachedRolePoliciesRequest {
            roleName = roleNameVal
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listAttachedRolePolicies(request)
        val attachedPolicies = response.attachedPolicies

        // Ensure that the policy is not attached to this role.
        val checkStatus: Int
        if (attachedPolicies != null) {
            checkStatus = checkList(attachedPolicies, policyArnVal)
            if (checkStatus == -1) {
                return
            }
        }
```

```
        val policyRequest =
            AttachRolePolicyRequest {
                roleName = roleNameVal
                policyArn = policyArnVal
            }
        iamClient.attachRolePolicy(policyRequest)
        println("Successfully attached policy $policyArnVal to role
 $roleNameVal")
    }
}

fun checkList(
    attachedPolicies: List<AttachedPolicy>,
    policyArnVal: String,
): Int {
    for (policy in attachedPolicies) {
        val polArn = policy.policyArn.toString()

        if (polArn.compareTo(policyArnVal) == 0) {
            println("The policy is already attached to this role.")
            return -1
        }
    }
    return 0
}
```

- Per i dettagli sull'API, AttachRolePolicyconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();
```

```
$assumeRolePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{$user['Arn']}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }";
$assumeRoleRole = $service->createRole("iam_demo_role_$uuid",
 $assumeRolePolicyDocument);
echo "Created role: {$assumeRoleRole['RoleName']}\n";

$listAllBucketsPolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
}";
$listAllBucketsPolicy = $service->createPolicy("iam_demo_policy_$uuid",
 $listAllBucketsPolicyDocument);
echo "Created policy: {$listAllBucketsPolicy['PolicyName']}\n";

$service->attachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);

    public function attachRolePolicy($roleName, $policyArn)
    {
        return $this->customWaiter(function () use ($roleName, $policyArn) {
            $this->iamClient->attachRolePolicy([
                'PolicyArn' => $policyArn,
                'RoleName' => $roleName,
            ]);
        });
    }
```

- Per i dettagli sull'API, consulta la AttachRolePolicysezione AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio allega la policy AWS gestita denominata **`SecurityAudit`** al ruolo IAM. **`CoSecurityAuditors`** Gli utenti che assumo quel ruolo sono immediatamente interessati dalle autorizzazioni definite nella versione predefinita di tale policy.

```
Register-IAMRolePolicy -RoleName CoSecurityAuditors -PolicyArn
 arn:aws:iam::aws:policy/SecurityAudit
```

- Per i dettagli sull'API, vedere AttachRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio allega la policy AWS gestita denominata **`SecurityAudit`** al ruolo IAM. **`CoSecurityAuditors`** Gli utenti che assumo quel ruolo sono immediatamente interessati dalle autorizzazioni definite nella versione predefinita di tale policy.

```
Register-IAMRolePolicy -RoleName CoSecurityAuditors -PolicyArn
 arn:aws:iam::aws:policy/SecurityAudit
```

- Per i dettagli sull'API, vedere AttachRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Collega una policy a un ruolo utilizzando l'oggetto Policy Boto3.

```
def attach_to_role(role_name, policy_arn):
```

```
    """
    Attaches a policy to a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Policy(policy_arn).attach_role(RoleName=role_name)
        logger.info("Attached policy %s to role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to role %s.", policy_arn,
 role_name)
        raise
```

Collega una policy a un ruolo utilizzando l'oggetto Role Boto3.

```
def attach_policy(role_name, policy_arn):
    """
    Attaches a policy to a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Role(role_name).attach_policy(PolicyArn=policy_arn)
        logger.info("Attached policy %s to role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to role %s.", policy_arn,
 role_name)
        raise
```

- Per i dettagli sull'API, consulta AttachRolePolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, collega e scollega le policy relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'PolicyManager'
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end

  # Fetches an IAM policy by its ARN
  # @param policy_arn [String] the ARN of the IAM policy to retrieve
  # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
  def get_policy(policy_arn)
```

```
    response = @iam_client.get_policy(policy_arn: policy_arn)
    policy = response.policy
    @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
    policy
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
    raise
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
    raise
  end

  # Attaches a policy to a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_role(role_name, policy_arn)
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
```

```
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta la AttachRolePolicysezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn attach_role_policy(
    client: &iamClient,
    role: &Role,
    policy: &Policy,
) -> Result<AttachRolePolicyOutput, SdkError<AttachRolePolicyError>> {
    client
        .attach_role_policy()
        .role_name(role.role_name())
        .policy_arn(policy.arn().unwrap_or_default())
        .send()
        .await
```

```
}
```

- Per i dettagli sulle API, consulta la AttachRolePolicyguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    lo_iam->attachrolepolicy(
      iv_rolename = iv_role_name
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy attached to role successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role or policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, AttachRolePolicyconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func attachRolePolicy(role: String, policyArn: String) async throws {
        let input = AttachRolePolicyInput(
            policyArn: policyArn,
            roleName: role
        )
        do {
            _ = try await client.attachRolePolicy(input: input)
        } catch {
            print("ERROR: Attaching a role policy:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la AttachRolePolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`AttachUserPolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`AttachUserPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di utenti di sola lettura e di lettura e scrittura

CLI

AWS CLI

Come collegare una policy gestita a un utente IAM

Il `attach-user-policy` comando seguente collega la policy AWS gestita denominata `AdministratorAccess` all'utente IAM denominato. `Alice`

```
aws iam attach-user-policy \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \
    --user-name Alice
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy gestite e policy inline nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AttachUserPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio allega la policy AWS gestita denominata **`AmazonCognitoPowerUser`** all'utente IAM. **`Bob`** L'utente è immediatamente interessato dalle autorizzazioni definite nella versione più recente di tale policy.

```
Register-IAMUserPolicy -UserName Bob -PolicyArn arn:aws:iam::aws:policy/
AmazonCognitoPowerUser
```

- Per i dettagli sull'API, vedere AttachUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio allega la policy AWS gestita denominata **`AmazonCognitoPowerUser`** all'utente IAM. **`Bob`** L'utente è immediatamente interessato dalle autorizzazioni definite nella versione più recente di tale policy.

```
Register-IAMUserPolicy -UserName Bob -PolicyArn arn:aws:iam::aws:policy/
AmazonCognitoPowerUser
```

- Per i dettagli sull'API, vedere AttachUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def attach_policy(user_name, policy_arn):
    """
    Attaches a policy to a user.

    :param user_name: The name of the user.
    :param policy_arn: The Amazon Resource Name (ARN) of the policy.
    """
    try:
        iam.User(user_name).attach_policy(PolicyArn=policy_arn)
        logger.info("Attached policy %s to user %s.", policy_arn, user_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to user %s.", policy_arn,
 user_name)
        raise
```

- Per i dettagli sull'API, consulta AttachUserPolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
  # Attaches a policy to a user
  #
  # @param user_name [String] The name of the user
  # @param policy_arn [String] The Amazon Resource Name (ARN) of the policy
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_user(user_name, policy_arn)
    @iam_client.attach_user_policy(
      user_name: user_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to user: #{e.message}")
    false
  end
```

- Per i dettagli sull'API, consulta la AttachUserPolicysezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn attach_user_policy(
    client: &iamClient,
    user_name: &str,
    policy_arn: &str,
) -> Result<(), iamError> {
    client
        .attach_user_policy()
        .user_name(user_name)
        .policy_arn(policy_arn)
        .send()
        .await?;
```

```
    Ok(())
}
```

- Per i dettagli sulle API, consulta la AttachUserPolicyguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        lo_iam->attachuserpolicy(
          iv_username = iv_user_name
          iv_policyarn = iv_policy_arn ).
        MESSAGE 'Policy attached to user successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'User or policy does not exist.' TYPE 'E'.
      CATCH /aws1/cx_iamlimitexceededex.
        MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, AttachUserPolicyconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ChangePassword`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ChangePassword`.

CLI

AWS CLI

Per modificare la password per l'utente IAM

Per modificare la password per il tuo utente IAM, ti consigliamo di utilizzare il parametro `--cli-input-json` per passare un file JSON che contenga la vecchia e la nuova password. Con questo metodo, potrai utilizzare password complesse con caratteri non alfanumerici. Quando le password vengono passate come parametri della riga di comando può essere difficile utilizzare password con caratteri non alfanumerici. Per utilizzare il parametro `--cli-input-json`, inizia a utilizzare il comando `change-password` con il parametro `--generate-cli-skeleton`, come nell'esempio seguente.

```
aws iam change-password \
    --generate-cli-skeleton > change-password.json
```

Il comando precedente crea un file JSON chiamato change-password.json che può essere utilizzato per inserire la vecchia e la nuova password. Ad esempio, il file potrebbe avere il seguente aspetto.

```
{
    "OldPassword": "3s0K_;xh4~8XXI",
    "NewPassword": "]35d/{pB9Fo9wJ"
}
```

Quindi, per modificare la password, usa nuovamente il comando `change-password`, questa volta passando il parametro per specificare il file JSON `--cli-input-json`. Il comando `change-password` seguente utilizza il parametro `--cli-input-json` con un file JSON chiamato change-password.json.

```
aws iam change-password \
    --cli-input-json file://change-password.json
```

Questo comando non produce alcun output.

Questo comando può essere chiamato solo dagli utenti IAM. Se questo comando viene chiamato utilizzando le credenziali AWS dell'account (root), restituisce un `InvalidUserType` errore.

Per ulteriori informazioni, consulta Come permettere a un utente IAM di cambiare la propria password nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, vedere ChangePasswordin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo comando modifica la password dell'utente che esegue il comando. Questo comando può essere chiamato solo dagli utenti IAM. Se questo comando viene chiamato quando si accede con le credenziali AWS dell'account (root), restituisce un errore. **`InvalidUserType`**

```
Edit-IAMPassword -OldPassword "MyOldP@ssw0rd" -NewPassword "MyNewP@ssw0rd"
```

- Per i dettagli sull'API, vedere ChangePasswordin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando modifica la password dell'utente che esegue il comando. Questo comando può essere chiamato solo dagli utenti IAM. Se questo comando viene chiamato quando si accede con le credenziali AWS dell'account (root), restituisce un errore. **`InvalidUserType`**

```
Edit-IAMPassword -OldPassword "MyOldP@ssw0rd" -NewPassword "MyNewP@ssw0rd"
```

- Per i dettagli sull'API, vedere ChangePasswordin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateAccessKey`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateAccessKey`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Creazione di utenti di sola lettura e di lettura e scrittura
- Gestione delle chiavi di accesso

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an IAM access key for a user.
    /// </summary>
    /// <param name="userName">The username for which to create the IAM access
    /// key.</param>
    /// <returns>The AccessKey.</returns>
    public async Task<AccessKey> CreateAccessKeyAsync(string userName)
    {
        var response = await _IAMService.CreateAccessKeyAsync(new
CreateAccessKeyRequest
        {
            UserName = userName,
        });

        return response.AccessKey;

    }
```

- Per i dettagli sull'API, CreateAccessKeyconsulta AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
#     And:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
```

```
  echo "  -u user_name   The name of the IAM user."
  echo "  [-f file_name]   Optional file name for the access key output."
  echo ""
}

# Retrieve the calling parameters.
while getopts "u:f:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    f) file_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

response=$(aws iam create-access-key \
  --user-name "$user_name" \
  --output text)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports create-access-key operation failed.$response"
  return 1
fi

if [[ -n "$file_name" ]]; then
  echo "$response" >"$file_name"
fi
```

```
  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}
```

- Per i dettagli sull'API, consulta CreateAccessKey AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
Aws::String AwsDoc::IAM::createAccessKey(const Aws::String &userName,
                                         const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::CreateAccessKeyRequest request;
    request.SetUserName(userName);

    Aws::String result;
    Aws::IAM::Model::CreateAccessKeyOutcome outcome =
 iam.CreateAccessKey(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error creating access key for IAM user " << userName
                  << ":" << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        const auto &accessKey = outcome.GetResult().GetAccessKey();
```

```
        std::cout << "Successfully created access key for IAM user " <<
                  userName << std::endl << "  aws_access_key_id = " <<
                  accessKey.GetAccessKeyId() << std::endl <<
                  " aws_secret_access_key = " << accessKey.GetSecretAccessKey()
 <<
                  std::endl;
        result = accessKey.GetAccessKeyId();
    }

    return result;
}
```

- Per i dettagli sull'API, CreateAccessKeyconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come creare una chiave di accesso per un utente IAM

Il comando `create-access-key` seguente crea una chiave di accesso (ID chiave di accesso e chiave di accesso segreta) per l'utente IAM denominato Bob.

```
aws iam create-access-key \
    --user-name Bob
```

Output:

```
{
    "AccessKey": {
        "UserName": "Bob",
        "Status": "Active",
        "CreateDate": "2015-03-09T18:39:23.411Z",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    }
}
```

Conserva la chiave di accesso segreta in un luogo sicuro. Se viene persa, non può essere recuperata e dovrai creare una nuova chiave di accesso.

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta CreateAccessKey AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// CreateAccessKeyPair creates an access key for a user. The returned access key
 contains
// the ID and secret credentials needed to use the key.
func (wrapper UserWrapper) CreateAccessKeyPair(ctx context.Context, userName
 string) (*types.AccessKey, error) {
 var key *types.AccessKey
```

```
 result, err := wrapper.IamClient.CreateAccessKey(ctx, &iam.CreateAccessKeyInput{
  UserName: aws.String(userName)})
 if err != nil {
  log.Printf("Couldn't create access key pair for user %v. Here's why: %v\n",
 userName, err)
 } else {
  key = result.AccessKey
 }
 return key, err
}
```

- Per i dettagli sull'API, CreateAccessKeyconsulta AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.CreateAccessKeyRequest;
import software.amazon.awssdk.services.iam.model.CreateAccessKeyResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateAccessKey {
    public static void main(String[] args) {
```

```
        final String usage = """

                Usage:
                   <user>\s

                Where:
                   user - An AWS IAM user that you can obtain from the AWS
 Management Console.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String user = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        String keyId = createIAMAccessKey(iam, user);
        System.out.println("The Key Id is " + keyId);
        iam.close();
    }

    public static String createIAMAccessKey(IamClient iam, String user) {
        try {
            CreateAccessKeyRequest request = CreateAccessKeyRequest.builder()
                    .userName(user)
                    .build();

            CreateAccessKeyResponse response = iam.createAccessKey(request);
            return response.accessKey().accessKeyId();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Per i dettagli sull'API, CreateAccessKeyconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea la chiave di accesso.

```
import { CreateAccessKeyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} userName
 */
export const createAccessKey = (userName) => {
  const command = new CreateAccessKeyCommand({ UserName: userName });
  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, CreateAccessKeyconsulta AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
```

```
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.createAccessKey({ UserName: "IAM_USER_NAME" }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.AccessKey);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, CreateAccessKeyconsulta AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun createIAMAccessKey(user: String?): String {
    val request =
        CreateAccessKeyRequest {
            userName = user
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createAccessKey(request)
        return response.accessKey?.accessKeyId.toString()
    }
}
```

- Per i dettagli sull'API, CreateAccessKeyconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: questo esempio crea una nuova chiave di accesso e una coppia di chiavi di accesso segrete e le assegna all'utente **David**. Assicurati di salvare i valori **AccessKeyId** e **SecretAccessKey** in un file perché questa è l'unica volta in cui puoi ottenere il **SecretAccessKey**. Non puoi recuperarla in un secondo momento. Se perdi la chiave segreta, dovrai creare una nuova coppia di chiavi di accesso.

```
New-IAMAccessKey -UserName David
```

Output:

```
AccessKeyId     : AKIAIOSFODNN7EXAMPLE
CreateDate      : 4/13/2015 1:00:42 PM
SecretAccessKey : wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Status          : Active
UserName        : David
```

- Per i dettagli sull'API, vedere CreateAccessKeyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea una nuova chiave di accesso e una coppia di chiavi di accesso segrete e le assegna all'utente **David**. Assicurati di salvare i valori **AccessKeyId** e **SecretAccessKey** in un file perché questa è l'unica volta in cui puoi ottenere il **SecretAccessKey**. Non puoi recuperarla in un secondo momento. Se perdi la chiave segreta, dovrai creare una nuova coppia di chiavi di accesso.

```
New-IAMAccessKey -UserName David
```

Output:

```
AccessKeyId     : AKIAIOSFODNN7EXAMPLE
CreateDate      : 4/13/2015 1:00:42 PM
```

```
SecretAccessKey : wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Status          : Active
UserName        : David
```

- Per i dettagli sull'API, vedere CreateAccessKeyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def create_key(user_name):
    """
    Creates an access key for the specified user. Each user can have a
    maximum of two keys.

    :param user_name: The name of the user.
    :return: The created access key.
    """
    try:
        key_pair = iam.User(user_name).create_access_key_pair()
        logger.info(
            "Created access key pair for %s. Key ID is %s.",
            key_pair.user_name,
            key_pair.id,
        )
    except ClientError:
        logger.exception("Couldn't create access key pair for %s.", user_name)
        raise
    else:
        return key_pair
```

- Per i dettagli sull'API, consulta CreateAccessKey AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, disattiva ed elimina le chiavi di accesso.

```
# Manages access keys for IAM users
class AccessKeyManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'AccessKeyManager'
  end

  # Lists access keys for a user
  #
  # @param user_name [String] The name of the user.
  def list_access_keys(user_name)
    response = @iam_client.list_access_keys(user_name: user_name)
    if response.access_key_metadata.empty?
      @logger.info("No access keys found for user '#{user_name}'.")
    else
      response.access_key_metadata.map(&:access_key_id)
    end
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Error listing access keys: cannot find user '#{user_name}'.")
    []
  rescue StandardError => e
    @logger.error("Error listing access keys: #{e.message}")
    []
  end

  # Creates an access key for a user
```

```
  #
  # @param user_name [String] The name of the user.
  # @return [Boolean]
  def create_access_key(user_name)
    response = @iam_client.create_access_key(user_name: user_name)
    access_key = response.access_key
    @logger.info("Access key created for user '#{user_name}':
#{access_key.access_key_id}")
    access_key
  rescue Aws::IAM::Errors::LimitExceeded
    @logger.error('Error creating access key: limit exceeded. Cannot create
more.')
    nil
  rescue StandardError => e
    @logger.error("Error creating access key: #{e.message}")
    nil
  end

  # Deactivates an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def deactivate_access_key(user_name, access_key_id)
    @iam_client.update_access_key(
      user_name: user_name,
      access_key_id: access_key_id,
      status: 'Inactive'
    )
    true
  rescue StandardError => e
    @logger.error("Error deactivating access key: #{e.message}")
    false
  end

  # Deletes an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def delete_access_key(user_name, access_key_id)
    @iam_client.delete_access_key(
      user_name: user_name,
      access_key_id: access_key_id
```

```
    )
    true
  rescue StandardError => e
    @logger.error("Error deleting access key: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, CreateAccessKeyconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn create_access_key(client: &iamClient, user_name: &str) ->
 Result<AccessKey, iamError> {
    let mut tries: i32 = 0;
    let max_tries: i32 = 10;

    let response: Result<CreateAccessKeyOutput, SdkError<CreateAccessKeyError>> =
 loop {
        match client.create_access_key().user_name(user_name).send().await {
            Ok(inner_response) => {
                break Ok(inner_response);
            }
            Err(e) => {
                tries += 1;
                if tries > max_tries {
                    break Err(e);
                }
                sleep(Duration::from_secs(2)).await;
            }
        }
    };
```

```
    Ok(response.unwrap().access_key.unwrap())
}
```

- Per i dettagli sulle API, consulta il riferimento CreateAccessKeyall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    oo_result = lo_iam->createaccesskey(
      iv_username = iv_user_name ).
    MESSAGE 'Access key created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Maximum number of access keys reached.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, CreateAccessKeyconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func createAccessKey(userName: String) async throws ->
 IAMClientTypes.AccessKey {
        let input = CreateAccessKeyInput(
            userName: userName
        )
        do {
            let output = try await iamClient.createAccessKey(input: input)
            guard let accessKey = output.accessKey else {
                throw ServiceHandlerError.keyError
            }
            return accessKey
        } catch {
            print("ERROR: createAccessKey:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la CreateAccessKeyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`CreateAccountAlias`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateAccountAlias`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::createAccountAlias(const Aws::String &aliasName,
                                     const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::CreateAccountAliasRequest request;
    request.SetAccountAlias(aliasName);

    Aws::IAM::Model::CreateAccountAliasOutcome outcome = iam.CreateAccountAlias(
            request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error creating account alias " << aliasName << ": "
                  << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully created account alias " << aliasName <<
                  std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la CreateAccountAliassezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come Creare l'alias di un account

Il `create-account-alias` comando seguente crea l'alias `examplecorp` per il tuo AWS account.

```
aws iam create-account-alias \
    --account-alias examplecorp
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Your AWS account ID and its alias nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta CreateAccountAlias AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.CreateAccountAliasRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateAccountAlias {
    public static void main(String[] args) {
        final String usage = """
                Usage:
```

```
                    <alias>\s

                Where:
                    alias - The account alias to create (for example,
 myawsaccount).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alias = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        createIAMAccountAlias(iam, alias);
        iam.close();
        System.out.println("Done");
    }

    public static void createIAMAccountAlias(IamClient iam, String alias) {
        try {
            CreateAccountAliasRequest request =
 CreateAccountAliasRequest.builder()
                    .accountAlias(alias)
                    .build();

            iam.createAccountAlias(request);
            System.out.println("Successfully created account alias: " + alias);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la CreateAccountAliassezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea l'alias dell'account.

```
import { CreateAccountAliasCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} alias - A unique name for the account alias.
 * @returns
 */
export const createAccountAlias = (alias) => {
  const command = new CreateAccountAliasCommand({
    AccountAlias: alias,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la CreateAccountAliassezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.createAccountAlias({ AccountAlias: process.argv[2] }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la CreateAccountAliassezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun createIAMAccountAlias(alias: String) {
    val request =
        CreateAccountAliasRequest {
            accountAlias = alias
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.createAccountAlias(request)
        println("Successfully created account alias named $alias")
```

```
    }
}
```

- Per i dettagli sull'API, CreateAccountAliasconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: Questo esempio modifica l'alias dell' AWS account in. **mycompanyaws** <accountidnumber>L'indirizzo della pagina di accesso dell'utente passa all'URL originale utilizzando https://mycompanyaws.signin.aws.amazon.com/console. il numero ID dell'account anziché l'alias (https://.signin.aws.amazon). com/console) continua a funzionare. Tuttavia, tutti gli URL basati su alias precedentemente definiti smettono di funzionare.

```
New-IAMAccountAlias -AccountAlias mycompanyaws
```

- Per i dettagli sull'API, vedere CreateAccountAliasin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio modifica l'alias dell' AWS account in. **mycompanyaws** <accountidnumber>L'indirizzo della pagina di accesso dell'utente passa all'URL originale utilizzando https://mycompanyaws.signin.aws.amazon.com/console. il numero ID dell'account anziché l'alias (https://.signin.aws.amazon). com/console) continua a funzionare. Tuttavia, tutti gli URL basati su alias precedentemente definiti smettono di funzionare.

```
New-IAMAccountAlias -AccountAlias mycompanyaws
```

- Per i dettagli sull'API, vedere CreateAccountAliasin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def create_alias(alias):
    """
    Creates an alias for the current account. The alias can be used in place of
 the
    account ID in the sign-in URL. An account can have only one alias. When a new
    alias is created, it replaces any existing alias.

    :param alias: The alias to assign to the account.
    """

    try:
        iam.create_account_alias(AccountAlias=alias)
        logger.info("Created an alias '%s' for your account.", alias)
    except ClientError:
        logger.exception("Couldn't create alias '%s' for your account.", alias)
        raise
```

- Per i dettagli sull'API, consulta CreateAccountAlias AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca, crea ed elimina gli alias degli account.

```
class IAMAliasManager
  # Initializes the IAM client and logger
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists available AWS account aliases.
  def list_aliases
    response = @iam_client.list_account_aliases

    if response.account_aliases.count.positive?
      @logger.info('Account aliases are:')
      response.account_aliases.each { |account_alias| @logger.info("
 #{account_alias}") }
    else
      @logger.info('No account aliases found.')
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing account aliases: #{e.message}")
  end

  # Creates an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to create.
  # @return [Boolean] true if the account alias was created; otherwise, false.
  def create_account_alias(account_alias)
    @iam_client.create_account_alias(account_alias: account_alias)
    true
```

```
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating account alias: #{e.message}")
    false
  end

  # Deletes an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to delete.
  # @return [Boolean] true if the account alias was deleted; otherwise, false.
  def delete_account_alias(account_alias)
    @iam_client.delete_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting account alias: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta la CreateAccountAliassezione AWS SDK per Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        lo_iam->createaccountalias(
          iv_accountalias = iv_account_alias ).
        MESSAGE 'Account alias created successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamentityalrdyexex.
        MESSAGE 'Account alias already exists.' TYPE 'E'.
      CATCH /aws1/cx_iamlimitexceededex.
        MESSAGE 'Account alias limit exceeded.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, CreateAccountAliasconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`CreateGroup`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateGroup`.

CLI

AWS CLI

Come creare un gruppo IAM

Il comando `create-group` seguente crea un gruppo IAM denominato `Admins`.

```
aws iam create-group \
    --group-name Admins
```

Output:

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-03-09T20:30:24.940Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    }
}
```

Per ulteriori informazioni, consulta Creazione di gruppi di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta CreateGroup AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { CreateGroupCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} groupName
 */
export const createGroup = async (groupName) => {
  const command = new CreateGroupCommand({ GroupName: groupName });

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per i dettagli sull'API, consulta la CreateGroupsezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea un nuovo gruppo IAM denominato **`Developers`**.

```
New-IAMGroup -GroupName Developers
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Developers
```

```
CreateDate : 4/14/2015 11:21:31 AM
GroupId    : QNEJ5PM4NFSQCEXAMPLE1
GroupName  : Developers
Path       : /
```

- Per i dettagli sull'API, vedere CreateGroupin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea un nuovo gruppo IAM denominato **`Developers`**.

```
New-IAMGroup -GroupName Developers
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Developers
CreateDate : 4/14/2015 11:21:31 AM
GroupId    : QNEJ5PM4NFSQCEXAMPLE1
GroupName  : Developers
Path       : /
```

- Per i dettagli sull'API, vedere CreateGroupin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`CreateInstanceProfile`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateInstanceProfile`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione e gestione di un servizio resiliente
- Creazione di un servizio contenitore per le istanze di macchine virtuali
- Guida introduttiva allo streaming gestito
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
   /// <summary>
   /// Create a policy, role, and profile that is associated with instances with
a specified name.
   /// An instance's associated profile defines a role that is assumed by the
   /// instance.The role has attached policies that specify the AWS permissions
granted to
   /// clients that run on the instance.
   /// </summary>
   /// <param name="policyName">Name to use for the policy.</param>
   /// <param name="roleName">Name to use for the role.</param>
   /// <param name="profileName">Name to use for the profile.</param>
   /// <param name="ssmOnlyPolicyFile">Path to a policy file for SSM.</param>
   /// <param name="awsManagedPolicies">AWS Managed policies to be attached to
the role.</param>
   /// <returns>The Arn of the profile.</returns>
   public async Task<string> CreateInstanceProfileWithName(
       string policyName,
       string roleName,
       string profileName,
       string ssmOnlyPolicyFile,
       List<string>? awsManagedPolicies = null)
   {

       var assumeRoleDoc = "{" +
                                   "\"Version\": \"2012-10-17\"," +
                                   "\"Statement\": [{" +
                                        "\"Effect\": \"Allow\"," +
                                        "\"Principal\": {" +
                                        "\"Service\": [" +
                                            "\"ec2.amazonaws.com\"" +
                                        "]" +
                                        "}," +
```

```
                                "\"Action\": \"sts:AssumeRole\"" +
                                "}]" +
                            "}";

        var policyDocument = await File.ReadAllTextAsync(ssmOnlyPolicyFile);

        var policyArn = "";

        try
        {
            var createPolicyResult = await _amazonIam.CreatePolicyAsync(
                new CreatePolicyRequest
                {
                    PolicyName = policyName,
                    PolicyDocument = policyDocument
                });
            policyArn = createPolicyResult.Policy.Arn;
        }
        catch (EntityAlreadyExistsException)
        {
            // The policy already exists, so we look it up to get the Arn.
            var policiesPaginator = _amazonIam.Paginators.ListPolicies(
                new ListPoliciesRequest()
                {
                    Scope = PolicyScopeType.Local
                });
            // Get the entire list using the paginator.
            await foreach (var policy in policiesPaginator.Policies)
            {
                if (policy.PolicyName.Equals(policyName))
                {
                    policyArn = policy.Arn;
                }
            }

            if (policyArn == null)
            {
                throw new InvalidOperationException("Policy not found");
            }
        }

        try
        {
            await _amazonIam.CreateRoleAsync(new CreateRoleRequest()
```

```
            {
                RoleName = roleName,
                AssumeRolePolicyDocument = assumeRoleDoc,
            });
            await _amazonIam.AttachRolePolicyAsync(new AttachRolePolicyRequest()
            {
                RoleName = roleName,
                PolicyArn = policyArn
            });
            if (awsManagedPolicies != null)
            {
                foreach (var awsPolicy in awsManagedPolicies)
                {
                    await _amazonIam.AttachRolePolicyAsync(new
AttachRolePolicyRequest()
                    {
                        PolicyArn = $"arn:aws:iam::aws:policy/{awsPolicy}",
                        RoleName = roleName
                    });
                }
            }
        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine("Role already exists.");
        }

        string profileArn = "";
        try
        {
            var profileCreateResponse = await
_amazonIam.CreateInstanceProfileAsync(
                new CreateInstanceProfileRequest()
                {
                    InstanceProfileName = profileName
                });
            // Allow time for the profile to be ready.
            profileArn = profileCreateResponse.InstanceProfile.Arn;
            Thread.Sleep(10000);
            await _amazonIam.AddRoleToInstanceProfileAsync(
                new AddRoleToInstanceProfileRequest()
                {
                    InstanceProfileName = profileName,
                    RoleName = roleName
```

```
                });

        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine("Policy already exists.");
            var profileGetResponse = await _amazonIam.GetInstanceProfileAsync(
                new GetInstanceProfileRequest()
                {
                    InstanceProfileName = profileName
                });
            profileArn = profileGetResponse.InstanceProfile.Arn;
        }
        return profileArn;
    }
```

- Per i dettagli sull'API, consulta la CreateInstanceProfilesezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come creare un profilo dell'istanza

Il comando `create-instance-profile` seguente crea un profilo dell'istanza denominato `Webserver`.

```
aws iam create-instance-profile \
    --instance-profile-name Webserver
```

Output:

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJMBYC7DLSPEXAMPLE",
        "Roles": [],
        "CreateDate": "2015-03-09T20:33:19.626Z",
        "InstanceProfileName": "Webserver",
        "Path": "/",
```

```
        "Arn": "arn:aws:iam::123456789012:instance-profile/Webserver"
    }
}
```

Per aggiungere un ruolo a un profilo dell'istanza, utilizza il comando `add-role-to-instance-profile`.

Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta CreateInstanceProfile AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
const { InstanceProfile } = await iamClient.send(
  new CreateInstanceProfileCommand({
    InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
  }),
);
await waitUntilInstanceProfileExists(
  { client: iamClient },
  { InstanceProfileName: NAMES.ssmOnlyInstanceProfileName },
);
```

- Per i dettagli sull'API, consulta la CreateInstanceProfilesezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea un nuovo profilo dell'istanza IAM denominato **`ProfileForDevEC2Instance`**. È necessario eseguire il comando **`Add-IAMRoleToInstanceProfile`** separatamente per associare il profilo dell'istanza a un ruolo IAM esistente che fornisce le autorizzazioni all'istanza. Infine, collega il profilo dell'istanza a un'istanza EC2 all'avvio. A tale scopo, utilizza il cmdlet **`New-EC2Instance`** con il parametro **`InstanceProfile_Arn`** o **`InstanceProfile_Name`**.

```
New-IAMInstanceProfile -InstanceProfileName ProfileForDevEC2Instance
```

Output:

```
Arn                 : arn:aws:iam::123456789012:instance-profile/
ProfileForDevEC2Instance
CreateDate          : 4/14/2015 11:31:39 AM
InstanceProfileId   : DYMFXL556EY46EXAMPLE1
InstanceProfileName : ProfileForDevEC2Instance
Path                : /
Roles               : {}
```

- Per i dettagli sull'API, vedere CreateInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea un nuovo profilo dell'istanza IAM denominato **`ProfileForDevEC2Instance`**. È necessario eseguire il comando **`Add-IAMRoleToInstanceProfile`** separatamente per associare il profilo dell'istanza a un ruolo IAM esistente che fornisce le autorizzazioni all'istanza. Infine, collega il profilo dell'istanza a un'istanza EC2 all'avvio. A tale scopo, utilizza il cmdlet **`New-EC2Instance`** con il parametro **`InstanceProfile_Arn`** o **`InstanceProfile_Name`**.

```
New-IAMInstanceProfile -InstanceProfileName ProfileForDevEC2Instance
```

Output:

```
Arn                 : arn:aws:iam::123456789012:instance-profile/
ProfileForDevEC2Instance
```

```
CreateDate          : 4/14/2015 11:31:39 AM
InstanceProfileId   : DYMFXL556EY46EXAMPLE1
InstanceProfileName : ProfileForDevEC2Instance
Path                : /
Roles               : {}
```

- Per i dettagli sull'API, vedere CreateInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo esempio crea una policy, un ruolo e un profilo dell'istanza e li collega tutti insieme.

```
class AutoScalingWrapper:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix: str,
        inst_type: str,
        ami_param: str,
        autoscaling_client: boto3.client,
        ec2_client: boto3.client,
        ssm_client: boto3.client,
        iam_client: boto3.client,
    ):
        """
        Initializes the AutoScaler class with the necessary parameters.

        :param resource_prefix: The prefix for naming AWS resources that are
 created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
```

```
        :param ami_param: The Systems Manager parameter used to look up the AMI
that is created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
        self.ami_param = ami_param
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        sts_client = boto3.client("sts")
        self.account_id = sts_client.get_caller_identity()["Account"]

        self.key_pair_name = f"{resource_prefix}-key-pair"
        self.launch_template_name = f"{resource_prefix}-template-"
        self.group_name = f"{resource_prefix}-group"

        # Happy path
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
        self.instance_profile_name = f"{resource_prefix}-prof"

        # Failure mode
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
        self.bad_creds_role_name = f"{resource_prefix}-bc-role"
        self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"


    def create_instance_profile(
        self,
        policy_file: str,
        policy_name: str,
        role_name: str,
        profile_name: str,
        aws_managed_policies: Tuple[str, ...] = (),
    ) -> str:
        """
        Creates a policy, role, and profile that is associated with instances
created by
        this class. An instance's associated profile defines a role that is
assumed by the
```

```
        instance. The role has attached policies that specify the AWS permissions
granted to
        clients that run on the instance.

        :param policy_file: The name of a JSON file that contains the policy
definition to
                            create and attach to the role.
        :param policy_name: The name to give the created policy.
        :param role_name: The name to give the created role.
        :param profile_name: The name to the created profile.
        :param aws_managed_policies: Additional AWS-managed policies that are
attached to
                                     the role, such as
AmazonSSMManagedInstanceCore to grant
                                     use of Systems Manager to send commands to
the instance.
        :return: The ARN of the profile that is created.
        """
        assume_role_doc = {
            "Version":"2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {"Service": "ec2.amazonaws.com"},
                    "Action": "sts:AssumeRole",
                }
            ],
        }
        policy_arn = self.create_policy(policy_file, policy_name)
        self.create_role(role_name, assume_role_doc)
        self.attach_policy(role_name, policy_arn, aws_managed_policies)

        try:
            profile_response = self.iam_client.create_instance_profile(
                InstanceProfileName=profile_name
            )
            waiter = self.iam_client.get_waiter("instance_profile_exists")
            waiter.wait(InstanceProfileName=profile_name)
            time.sleep(10)  # wait a little longer
            profile_arn = profile_response["InstanceProfile"]["Arn"]
            self.iam_client.add_role_to_instance_profile(
                InstanceProfileName=profile_name, RoleName=role_name
            )
```

```
            log.info("Created profile %s and added role %s.", profile_name,
role_name)
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                prof_response = self.iam_client.get_instance_profile(
                    InstanceProfileName=profile_name
                )
                profile_arn = prof_response["InstanceProfile"]["Arn"]
                log.info(
                    "Instance profile %s already exists, nothing to do.",
profile_name
                )
            log.error(f"Full error:\n\t{err}")
        return profile_arn
```

- Per i dettagli sull'API, consulta CreateInstanceProfile AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`CreateLoginProfile`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateLoginProfile`.

CLI

AWS CLI

Per creare una password per un utente IAM

Per creare una password per un utente IAM, ti consigliamo di utilizzare il parametro `--cli-input-json` per passare un file JSON che contenga la password. Con questo metodo, potrai creare una password complessa con caratteri non alfanumerici. Quando le password vengono passate come parametri della riga di comando può essere difficile utilizzare password con caratteri non alfanumerici.

Per utilizzare il parametro `--cli-input-json`, inizia a utilizzare il comando `create-login-profile` con il parametro `--generate-cli-skeleton`, come nell'esempio seguente.

```
aws iam create-login-profile \
    --generate-cli-skeleton > create-login-profile.json
```

Il comando precedente crea un file JSON chiamato create-login-profile.json che può essere utilizzato per inserire le informazioni per un comando `create-login-profile` successivo. Ad esempio:

```
{
    "UserName": "Bob",
    "Password": "&1-3a6u:RA0djs",
    "PasswordResetRequired": true
}
```

Quindi, per modificare la password per un utente IAM, usa nuovamente il comando `create-login-profile`, questa volta passando il parametro `--cli-input-json` per specificare il file JSON. Il comando `create-login-profile` seguente utilizza il parametro `--cli-input-json` con un file JSON chiamato create-login-profile.json.

```
aws iam create-login-profile \
    --cli-input-json file://create-login-profile.json
```

Output:

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2015-03-10T20:55:40.274Z",
        "PasswordResetRequired": true
    }
}
```

Se la nuova password viola la policy delle password dell'account, il comando restituisce un errore `PasswordPolicyViolation`.

Per modificare la password di un utente che ne ha già una, usa `update-login-profile`. Per impostare una policy delle password per l'account, utilizza il comando `update-account-password-policy`.

Se la policy delle password dell'account lo consente, gli utenti IAM possono modificare le proprie password utilizzando il comando `change-password`.

Per ulteriori informazioni, consulta Gestione delle password per gli utenti IAM nella Guida per l'utente di AWS .

- Per i dettagli sull'API, consulta CreateLoginProfile AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea una password (temporanea) per l'utente IAM Bob e imposta l'indicatore che richiede all'utente di modificare la password al successivo accesso di **Bob**.

```
New-IAMLoginProfile -UserName Bob -Password P@ssw0rd -PasswordResetRequired $true
```

Output:

```
CreateDate                  PasswordResetRequired             UserName
----------                  ---------------------             --------
4/14/2015 12:26:30 PM       True                              Bob
```

- Per i dettagli sull'API, vedere CreateLoginProfilein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea una password (temporanea) per l'utente IAM Bob e imposta l'indicatore che richiede all'utente di modificare la password al successivo accesso di **Bob**.

```
New-IAMLoginProfile -UserName Bob -Password P@ssw0rd -PasswordResetRequired $true
```

Output:

```
CreateDate                  PasswordResetRequired             UserName
----------                  ---------------------             --------
```

```
4/14/2015 12:26:30 PM         True                                  Bob
```

- Per i dettagli sull'API, vedere CreateLoginProfilein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`CreateOpenIdConnectProvider`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateOpenIdConnectProvider`.

CLI

AWS CLI

Per creare un provider IAM OpenID Connect (OIDC)

Per creare un provider OpenID Connect (OIDC), consigliamo di utilizzare il parametro `--cli-input-json` per passare un file JSON che contenga i parametri richiesti. Quando si crea un provider OIDC, è necessario passare l'URL del provider e l'URL deve iniziare con `https://`. Può essere difficile passare l'URL come parametro della riga di comando, perché i caratteri due punti (:) e barra (/) hanno un significato speciale in alcuni ambienti della riga di comando. L'utilizzo del parametro `--cli-input-json` consente di aggirare questa limitazione.

Per utilizzare il parametro `--cli-input-json`, inizia a utilizzare il comando `create-open-id-connect-provider` con il parametro `--generate-cli-skeleton`, come nell'esempio seguente.

```
aws iam create-open-id-connect-provider \
    --generate-cli-skeleton > create-open-id-connect-provider.json
```

Il comando precedente crea un file JSON chiamato create-open-id-connect-provider.json che può essere utilizzato per inserire le informazioni per un comando `create-open-id-connect-provider` successivo. Ad esempio:

```
{
    "Url": "https://server.example.com",
    "ClientIDList": [
        "example-application-ID"
```

```
    ],
    "ThumbprintList": [
        "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"
    ]
}
```

Se gli endpoint discovery e JWKS del tuo provider OIDC utilizzano certificati diversi, includi le impronte digitali per entrambi gli endpoint nel. `ThumbprintList`

Successivamente, per creare il provider OpenID Connect (OIDC), utilizza nuovamente il comando `create-open-id-connect-provider`, questa volta passando il parametro `--cli-input-json` per specificare il file JSON. Il comando `create-open-id-connect-provider` seguente utilizza il parametro `--cli-input-json` con un file JSON chiamato create-open-id-connect-provider.json.

```
aws iam create-open-id-connect-provider \
    --cli-input-json file://create-open-id-connect-provider.json
```

Output:

```
{
    "OpenIDConnectProviderArn": "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
}
```

Per ulteriori informazioni sui provider OIDC, consulta Creazione di provider di identità OpenID Connect (OIDC) nella Guida per l'utente di AWS IAM.

Per ulteriori informazioni su come ottenere impronte digitali per un provider OIDC, consulta Ottenere l'impronta personale per un provider di identità OpenID Connect nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta Command Reference. CreateOpenIdConnectProviderAWS CLI

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea un provider OIDC IAM associato al servizio del provider compatibile con OIDC che si trova nell'URL **`https://example.oidcprovider.com`** e

nell'ID client **my-testapp-1**. Il provider OIDC fornisce l'impronta digitale. Per autenticare l'impronta digitale, segui i passaggi riportati in http://docs.aws.amazon.com/IAM/latest/UserGuide/identity-providers-oidc-obtain-thumbprint.html.

```
New-IAMOpenIDConnectProvider -Url https://example.oidcprovider.com -ClientIDList
 my-testapp-1 -ThumbprintList 990F419EXAMPLEECF12DDEDA5EXAMPLE52F20D9E
```

Output:

```
arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com
```

- Per i dettagli sull'API, vedere CreateOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea un provider OIDC IAM associato al servizio del provider compatibile con OIDC che si trova nell'URL **https://example.oidcprovider.com** e nell'ID client **my-testapp-1**. Il provider OIDC fornisce l'impronta digitale. Per autenticare l'impronta digitale, segui i passaggi riportati in http://docs.aws.amazon.com/IAM/latest/UserGuide/identity-providers-oidc-obtain-thumbprint.html.

```
New-IAMOpenIDConnectProvider -Url https://example.oidcprovider.com -ClientIDList
 my-testapp-1 -ThumbprintList 990F419EXAMPLEECF12DDEDA5EXAMPLE52F20D9E
```

Output:

```
arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com
```

- Per i dettagli sull'API, vedere CreateOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **CreatePolicy**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreatePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Creazione di utenti di sola lettura e di lettura e scrittura
- Creazione di un'area di lavoro di monitoraggio gestita
- Guida introduttiva allo streaming gestito
- Guida introduttiva all'orchestrazione del flusso di lavoro
- Gestione delle policy
- Configura il controllo degli accessi Attribute-Based
- Configurazione della gestione dei sistemi
- Lavora con l'API IAM Policy Builder

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an IAM policy.
    /// </summary>
    /// <param name="policyName">The name to give the new IAM policy.</param>
    /// <param name="policyDocument">The policy document for the new policy.</
param>
    /// <returns>The new IAM policy object.</returns>
    public async Task<ManagedPolicy> CreatePolicyAsync(string policyName, string
 policyDocument)
    {
        var response = await _IAMService.CreatePolicyAsync(new
 CreatePolicyRequest
        {
            PolicyDocument = policyDocument,
            PolicyName = policyName,
```

```
        });

        return response.Policy;
    }
```

- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_policy() {
```

```
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name   The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) policy_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-policy \
    --policy-name "$policy_name" \
```

```
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}
```

- Per i dettagli sull'API, consulta CreatePolicy AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
Aws::String AwsDoc::IAM::createPolicy(const Aws::String &policyName,
                                      const Aws::String &rsrcArn,
                                      const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::CreatePolicyRequest request;
    request.SetPolicyName(policyName);
    request.SetPolicyDocument(BuildSamplePolicyDocument(rsrcArn));

    Aws::IAM::Model::CreatePolicyOutcome outcome = iam.CreatePolicy(request);
    Aws::String result;
    if (!outcome.IsSuccess()) {
        std::cerr << "Error creating policy " << policyName << ": " <<
```

```
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        result = outcome.GetResult().GetPolicy().GetArn();
        std::cout << "Successfully created policy " << policyName <<
                  std::endl;
    }

    return result;
}

Aws::String AwsDoc::IAM::BuildSamplePolicyDocument(const Aws::String &rsrc_arn) {
    std::stringstream stringStream;
    stringStream << "{"
                 << "  \"Version\": \"2012-10-17\","
                 << "  \"Statement\": ["
                 << "    {"
                 << "        \"Effect\": \"Allow\","
                 << "        \"Action\": \"logs:CreateLogGroup\","
                 << "        \"Resource\": \""
                 << rsrc_arn
                 << "\""
                 << "    },"
                 << "    {"
                 << "        \"Effect\": \"Allow\","
                 << "        \"Action\": ["
                 << "            \"dynamodb:DeleteItem\","
                 << "            \"dynamodb:GetItem\","
                 << "            \"dynamodb:PutItem\","
                 << "            \"dynamodb:Scan\","
                 << "            \"dynamodb:UpdateItem\""
                 << "       ],"
                 << "       \"Resource\": \""
                 << rsrc_arn
                 << "\""
                 << "    }"
                 << "   ]"
                 << "}";

    return stringStream.str();
}
```

- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Esempio 1: Come creare una policy gestita dal cliente

Il comando seguente crea una policy gestita dal cliente denominata `my-policy`. Il file `policy.json` è un documento JSON nella cartella corrente che consente l'accesso in sola lettura alla cartella `shared` in un bucket Amazon S3 denominato `amzn-s3-demo-bucket`.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json
```

Contenuto di policy.json:

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/shared/*"
            ]
        }
    ]
}
```

Output:

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "CreateDate": "2015-06-01T19:31:18.620Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
        "DefaultVersionId": "v1",
```

```
        "Path": "/",
        "Arn": "arn:aws:iam::0123456789012:policy/my-policy",
        "UpdateDate": "2015-06-01T19:31:18.620Z"
    }
}
```

Per ulteriori informazioni sull'utilizzo dei file come input per i parametri di stringa, consultate Specificare i valori dei parametri per la AWS CLI nella Guida per l'utente della AWS CLI.

Esempio 2: Come creare una policy gestita dal cliente con una descrizione

Il comando seguente crea una policy gestita dal cliente denominata `my-policy` con una descrizione non modificabile.

Il file `policy.json` è un documento JSON nella cartella corrente che consente l'accesso in sola lettura a tutte le operazioni Put, List e Get per un bucket Amazon S3 denominato `amzn-s3-demo-bucket`.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json \
    --description "This policy grants access to all Put, Get, and List actions
 for amzn-s3-demo-bucket"
```

Contenuto di policy.json:

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
                "s3:ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

Output:

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "ANPAWGSUGIDPEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2023-05-24T22:38:47+00:00",
        "UpdateDate": "2023-05-24T22:38:47+00:00"
    }
}
```

Per ulteriori informazioni sulle Idenity-based politiche, consulta le politiche e Identity-based le politiche basate sulle risorse nella Guida per l'utente IAM.AWS

Esempio 3: creare una policy gestita dal cliente con i tag

Il comando seguente crea una policy gestita dal cliente denominata `my-policy` con tag. Questo esempio utilizza il `--tags` parametro con i seguenti tag: JSON-formatted . `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'` In alternativa, il parametro `--tags` può essere utilizzato con i tag in formato abbreviato: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

Il file `policy.json` è un documento JSON nella cartella corrente che consente l'accesso in sola lettura a tutte le operazioni Put, List e Get per un bucket Amazon S3 denominato `amzn-s3-demo-bucket`.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

Contenuto di policy.json:

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

Output:

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "ANPAWGSUGIDPEXAMPLE",
        "Arn": "arn:aws:iam::12345678012:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2023-05-24T23:16:39+00:00",
        "UpdateDate": "2023-05-24T23:16:39+00:00",
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
                "Key": "Location",
                "Value": "Seattle"
            {
        ]
    }
}
```

Per ulteriori informazioni sulle policy di applicazione di tag, consulta Applicazione di tag a policy gestite dal cliente nella Guida per l'utente IAM AWS .

- Per i dettagli sulle API, vedere CreatePolicyin AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// PolicyDocument defines a policy document as a Go struct that can be serialized
// to JSON.
type PolicyDocument struct {
 Version   string
 Statement []PolicyStatement
}

// PolicyStatement defines a statement in a policy document.
type PolicyStatement struct {
```

```
 Effect    string
 Action    []string
 Principal map[string]string `json:",omitempty"`
 Resource  *string           `json:",omitempty"`
}

// CreatePolicy creates a policy that grants a list of actions to the specified
 resource.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper PolicyWrapper) CreatePolicy(ctx context.Context, policyName string,
 actions []string,
 resourceArn string) (*types.Policy, error) {
 var policy *types.Policy
 policyDoc := PolicyDocument{
  Version: "2012-10-17",
  Statement: []PolicyStatement{{
   Effect:   "Allow",
   Action:   actions,
   Resource: aws.String(resourceArn),
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n",
 resourceArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreatePolicy(ctx, &iam.CreatePolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
 })
 if err != nil {
  log.Printf("Couldn't create policy %v. Here's why: %v\n", policyName, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}
```

- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.core.waiters.WaiterResponse;
import software.amazon.awssdk.services.iam.model.CreatePolicyRequest;
import software.amazon.awssdk.services.iam.model.CreatePolicyResponse;
import software.amazon.awssdk.services.iam.model.GetPolicyRequest;
import software.amazon.awssdk.services.iam.model.GetPolicyResponse;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.waiters.IamWaiter;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreatePolicy {

    public static final String PolicyDocument = "{" +
            "  \"Version\": \"2012-10-17\"," +
            "  \"Statement\": [" +
            "    {" +
            "        \"Effect\": \"Allow\"," +
            "        \"Action\": [" +
            "            \"dynamodb:DeleteItem\"," +
            "            \"dynamodb:GetItem\"," +
            "            \"dynamodb:PutItem\"," +
            "            \"dynamodb:Scan\"," +
            "            \"dynamodb:UpdateItem\"" +
```

```
            "       ]," +
            "       \"Resource\": \"*\"" +
            "    }" +
            "   ]" +
            "}";

    public static void main(String[] args) {

        final String usage = """
                Usage:
                    CreatePolicy <policyName>\s

                Where:
                    policyName - A unique policy name.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String policyName = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        String result = createIAMPolicy(iam, policyName);
        System.out.println("Successfully created a policy with this ARN value: "
+ result);
        iam.close();
    }

    public static String createIAMPolicy(IamClient iam, String policyName) {
        try {
            // Create an IamWaiter object.
            IamWaiter iamWaiter = iam.waiter();

            CreatePolicyRequest request = CreatePolicyRequest.builder()
                    .policyName(policyName)
                    .policyDocument(PolicyDocument)
                    .build();

            CreatePolicyResponse response = iam.createPolicy(request);
```

```
            // Wait until the policy is created.
            GetPolicyRequest polRequest = GetPolicyRequest.builder()
                    .policyArn(response.policy().arn())
                    .build();

            WaiterResponse<GetPolicyResponse> waitUntilPolicyExists =
 iamWaiter.waitUntilPolicyExists(polRequest);

 waitUntilPolicyExists.matched().response().ifPresent(System.out::println);
            return response.policy().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea la policy.

```
import { CreatePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} policyName
```

```
 */
export const createPolicy = (policyName) => {
  const command = new CreatePolicyCommand({
    PolicyDocument: JSON.stringify({
      Version: "2012-10-17",
      Statement: [
        {
          Effect: "Allow",
          Action: "*",
          Resource: "*",
        },
      ],
    }),
    PolicyName: policyName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var myManagedPolicy = {
  Version: "2012-10-17",
  Statement: [
```

```
    {
      Effect: "Allow",
      Action: "logs:CreateLogGroup",
      Resource: "RESOURCE_ARN",
    },
    {
      Effect: "Allow",
      Action: [
        "dynamodb:DeleteItem",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Scan",
        "dynamodb:UpdateItem",
      ],
      Resource: "RESOURCE_ARN",
    },
  ],
};

var params = {
  PolicyDocument: JSON.stringify(myManagedPolicy),
  PolicyName: "myDynamoDBPolicy",
};

iam.createPolicy(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun createIAMPolicy(policyNameVal: String?): String {
    val policyDocumentVal = """
    {
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "dynamodb:DeleteItem",
                    "dynamodb:GetItem",
                    "dynamodb:PutItem",
                    "dynamodb:Scan",
                    "dynamodb:UpdateItem"
                ],
                "Resource": "*"
            }
        ]
    }
    """.trimIndent()

    val request =
        CreatePolicyRequest {
            policyName = policyNameVal
            policyDocument = policyDocumentVal
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createPolicy(request)
        return response.policy?.arn.toString()
    }
}
```

- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

$listAllBucketsPolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
}";
$listAllBucketsPolicy = $service->createPolicy("iam_demo_policy_$uuid",
 $listAllBucketsPolicyDocument);
echo "Created policy: {$listAllBucketsPolicy['PolicyName']}\n";

    /**
     * @param string $policyName
     * @param string $policyDocument
     * @return array
     */
    public function createPolicy(string $policyName, string $policyDocument)
    {
        $result = $this->customWaiter(function () use ($policyName,
 $policyDocument) {
            return $this->iamClient->createPolicy([
                'PolicyName' => $policyName,
                'PolicyDocument' => $policyDocument,
            ]);
        });
        return $result['Policy'];
    }
```

- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio crea una nuova policy IAM nell' AWS account corrente denominato **MySamplePolicy** Il file **MySamplePolicy.json** fornisce il contenuto della policy. Tenere presente che per elaborare correttamente il file della policy JSON è necessario utilizzare il parametro di cambio **-Raw**.

```
New-IAMPolicy -PolicyName MySamplePolicy -PolicyDocument (Get-Content -Raw
 MySamplePolicy.json)
```

Output:

```
Arn              : arn:aws:iam::123456789012:policy/MySamplePolicy
AttachmentCount  : 0
CreateDate       : 4/14/2015 2:45:59 PM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : LD4KP6HVFE7WGEXAMPLE1
PolicyName       : MySamplePolicy
UpdateDate       : 4/14/2015 2:45:59 PM
```

- Per i dettagli sull'API, vedere CreatePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio crea una nuova policy IAM nell' AWS account corrente denominato **MySamplePolicy** Il file **MySamplePolicy.json** fornisce il contenuto della policy. Tenere presente che per elaborare correttamente il file della policy JSON è necessario utilizzare il parametro di cambio **-Raw**.

```
New-IAMPolicy -PolicyName MySamplePolicy -PolicyDocument (Get-Content -Raw
 MySamplePolicy.json)
```

Output:

```
Arn              : arn:aws:iam::123456789012:policy/MySamplePolicy
AttachmentCount  : 0
CreateDate       : 4/14/2015 2:45:59 PM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : LD4KP6HVFE7WGEXAMPLE1
PolicyName       : MySamplePolicy
UpdateDate       : 4/14/2015 2:45:59 PM
```

- Per i dettagli sull'API, vedere CreatePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def create_policy(name, description, actions, resource_arn):
    """
    Creates a policy that contains a single statement.

    :param name: The name of the policy to create.
    :param description: The description of the policy.
    :param actions: The actions allowed by the policy. These typically take the
                    form of service:action, such as s3:PutObject.
    :param resource_arn: The Amazon Resource Name (ARN) of the resource this
 policy
                         applies to. This ARN can contain wildcards, such as
                         'arn:aws:s3:::amzn-s3-demo-bucket/*' to allow actions on
 all objects
                         in the bucket named 'amzn-s3-demo-bucket'.
    :return: The newly created policy.
```

```
    """
    policy_doc = {
        "Version":"2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
 resource_arn}],
    }
    try:
        policy = iam.create_policy(
            PolicyName=name,
            Description=description,
            PolicyDocument=json.dumps(policy_doc),
        )
        logger.info("Created policy %s.", policy.arn)
    except ClientError:
        logger.exception("Couldn't create policy %s.", name)
        raise
    else:
        return policy
```

- Per i dettagli sull'API, consulta CreatePolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, collega e scollega le policy relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
```

```
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'PolicyManager'
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end

  # Fetches an IAM policy by its ARN
  # @param policy_arn [String] the ARN of the IAM policy to retrieve
  # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
  def get_policy(policy_arn)
    response = @iam_client.get_policy(policy_arn: policy_arn)
    policy = response.policy
    @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
    policy
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
    raise
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
    raise
  end

  # Attaches a policy to a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
```

```
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_role(role_name, policy_arn)
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn create_policy(
    client: &iamClient,
    policy_name: &str,
    policy_document: &str,
) -> Result<Policy, iamError> {
    let policy = client
        .create_policy()
        .policy_name(policy_name)
        .policy_document(policy_document)
        .send()
        .await?;
    Ok(policy.policy.unwrap())
}
```

- Per i dettagli sulle API, consulta il riferimento CreatePolicyall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->createpolicy(
          iv_policyname = iv_policy_name
```

```
          iv_policydocument = iv_policy_document
          iv_description = iv_description ).
        MESSAGE 'Policy created successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamentityalrdyexex.
        MESSAGE 'Policy already exists.' TYPE 'E'.
      CATCH /aws1/cx_iammalformedplydocex.
        MESSAGE 'Policy document is malformed.' TYPE 'E'.
      CATCH /aws1/cx_iamlimitexceededex.
        MESSAGE 'Policy limit exceeded.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, CreatePolicyconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func createPolicy(name: String, policyDocument: String) async throws -
> IAMClientTypes.Policy {
        let input = CreatePolicyInput(
            policyDocument: policyDocument,
            policyName: name
        )
        do {
            let output = try await iamClient.createPolicy(input: input)
            guard let policy = output.policy else {
                throw ServiceHandlerError.noSuchPolicy
            }
            return policy
        } catch {
            print("ERROR: createPolicy:", dump(error))
```

```
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la CreatePolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`CreatePolicyVersion`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreatePolicyVersion`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestione delle policy

CLI

AWS CLI

Come creare una nuova versione di una policy gestita

Questo esempio crea una nuova versione v2 della policy IAM il cui ARN è `arn:aws:iam::123456789012:policy/MyPolicy` e la rende la versione predefinita.

```
aws iam create-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --policy-document file://NewPolicyVersion.json \
    --set-as-default
```

Output:

```
{
    "PolicyVersion": {
```

```
        "CreateDate": "2015-06-16T18:56:03.721Z",
        "VersionId": "v2",
        "IsDefaultVersion": true
    }
}
```

Per ulteriori informazioni, consulta Controllo delle versioni delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta CreatePolicyVersion AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea una nuova versione "v2" della policy IAM il cui ARN è **`arn:aws:iam::123456789012:policy/MyPolicy`** e la rende la versione predefinita. Il file **`NewPolicyVersion.json`** fornisce il contenuto della policy. Tenere presente che per elaborare correttamente il file della policy JSON è necessario utilizzare il parametro di cambio **`-Raw`**.

```
New-IAMPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/MyPolicy -
PolicyDocument (Get-content -Raw NewPolicyVersion.json) -SetAsDefault $true
```

Output:

```
CreateDate                          Document                IsDefaultVersion
          VersionId
----------                          --------                ----------------
          ---------
4/15/2015 10:54:54 AM                                       True
          v2
```

- Per i dettagli sull'API, vedere CreatePolicyVersionin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea una nuova versione "v2" della policy IAM il cui ARN è **`arn:aws:iam::123456789012:policy/MyPolicy`** e la rende la versione predefinita. Il file **`NewPolicyVersion.json`** fornisce il contenuto della policy. Tenere presente che per

elaborare correttamente il file della policy JSON è necessario utilizzare il parametro di cambio **`-Raw`**.

```
New-IAMPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/MyPolicy -
PolicyDocument (Get-content -Raw NewPolicyVersion.json) -SetAsDefault $true
```

Output:

```
CreateDate                          Document                IsDefaultVersion
          VersionId
----------                          --------                ----------------
          ---------
4/15/2015 10:54:54 AM                                       True
          v2
```

- Per i dettagli sull'API, vedere CreatePolicyVersionin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def create_policy_version(policy_arn, actions, resource_arn, set_as_default):
    """
    Creates a policy version. Policies can have up to five versions. The default
    version is the one that is used for all resources that reference the policy.

    :param policy_arn: The ARN of the policy.
    :param actions: The actions to allow in the policy version.
    :param resource_arn: The ARN of the resource this policy version applies to.
    :param set_as_default: When True, this policy version is set as the default
                           version for the policy. Otherwise, the default
                           is not changed.
    :return: The newly created policy version.
```

```
    """
    policy_doc = {
        "Version":"2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
resource_arn}],
    }
    try:
        policy = iam.Policy(policy_arn)
        policy_version = policy.create_version(
            PolicyDocument=json.dumps(policy_doc), SetAsDefault=set_as_default
        )
        logger.info(
            "Created policy version %s for policy %s.",
            policy_version.version_id,
            policy_version.arn,
        )
    except ClientError:
        logger.exception("Couldn't create a policy version for %s.", policy_arn)
        raise
    else:
        return policy_version
```

- Per i dettagli sull'API, consulta CreatePolicyVersion AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->createpolicyversion(
          iv_policyarn = iv_policy_arn
```

```
          iv_policydocument = iv_policy_document
          iv_setasdefault = iv_set_as_default ).
        MESSAGE 'Policy version created successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Policy does not exist.' TYPE 'E'.
      CATCH /aws1/cx_iammalformedplydocex.
        MESSAGE 'Policy document is malformed.' TYPE 'E'.
      CATCH /aws1/cx_iamlimitexceededex.
        MESSAGE 'Policy version limit exceeded.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, CreatePolicyVersionconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`CreateRole`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateRole`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Configura la connettività dei servizi container
- Crea un'attività contenitore per il tipo di avvio serverless
- Crea un'API REST con integrazione con proxy di funzioni
- Creazione di un servizio contenitore per le istanze di macchine virtuali
- Creazione di un'area di lavoro di monitoraggio gestita
- Creazione di una dashboard di monitoraggio con il nome della funzione come variabile
- Creazione della tua prima funzione serverless
- Nozioni di base sui data warehouse serverless
- Guida introduttiva alla gestione della configurazione
- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
- Guida introduttiva agli store di funzionalità di apprendimento automatico

- Guida introduttiva ai cluster Kubernetes gestiti
- Guida introduttiva allo streaming gestito
- Guida introduttiva ai cluster di data warehouse con provisioning
- Guida introduttiva all'orchestrazione del flusso di lavoro
- Gestione dei ruoli
- Spostamento di segreti codificati in uno spazio di archiviazione segreto sicuro
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori
- Configurazione della gestione dei sistemi
- Utilizzo delle variabili di proprietà nei dashboard di monitoraggio per monitorare più funzioni serverless
- Lavora con Streams e Time-to-Live

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Create a new IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="rolePolicyDocument">The name of the IAM policy document
    /// for the new role.</param>
    /// <returns>The Amazon Resource Name (ARN) of the role.</returns>
    public async Task<string> CreateRoleAsync(string roleName, string
rolePolicyDocument)
    {
        var request = new CreateRoleRequest
        {
            RoleName = roleName,
            AssumeRolePolicyDocument = rolePolicyDocument,
        };
```

```
        var response = await _IAMService.CreateRoleAsync(request);
        return response.Role.Arn;
    }
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
```

```
########################################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_json -- The assume role policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi
```

```
  response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta CreateRole AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::createIamRole(
        const Aws::String &roleName,
        const Aws::String &policy,
        const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::IAM::IAMClient client(clientConfig);
    Aws::IAM::Model::CreateRoleRequest request;

    request.SetRoleName(roleName);
    request.SetAssumeRolePolicyDocument(policy);
```

```
    Aws::IAM::Model::CreateRoleOutcome outcome = client.CreateRole(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error creating role. " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        const Aws::IAM::Model::Role iamRole = outcome.GetResult().GetRole();
        std::cout << "Created role " << iamRole.GetRoleName() << "\n";
        std::cout << "ID: " << iamRole.GetRoleId() << "\n";
        std::cout << "ARN: " << iamRole.GetArn() << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Esempio 1: Come creare un ruolo IAM

Il comando `create-role` seguente crea un ruolo denominato `Test-Role` e collega una policy di attendibilità a tale ruolo.

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json
```

Output:

```
{
    "Role": {
        "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "CreateDate": "2013-06-07T20:43:32.821Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
```

```
}
```

La policy di fiducia è definita come un documento JSON nel Test-Role-Trust-Policy.jsonfile. (Il nome e l'estensione del file non hanno importanza.) La policy di attendibilità deve specificare un principale.

Per collegare una policy di autorizzazioni a un ruolo, usa il comando `put-role-policy`.

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

Esempio 2: Come creare un ruolo IAM con una durata massima della sessione specificata

Il comando `create-role` seguente crea un ruolo denominato `Test-Role` e imposta una durata massima della sessione di 7200 secondi (2 ore).

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
    --max-session-duration 7200
```

Output:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Test-Role",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:role/Test-Role",
        "CreateDate": "2023-05-24T23:50:25+00:00",
        "AssumeRolePolicyDocument": {
            "Version":"2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::12345678012:root"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
```

```
    }
}
```

Per ulteriori informazioni, consulta Modificare la durata massima della sessione (AWS API) di un ruolo nella Guida per l'utente AWS IAM.

Esempio 3: Come creare un ruolo IAM con tag

Il comando seguente crea un ruolo IAM `Test-Role` con tag. Questo esempio utilizza il `--tags` parametro flag con i seguenti JSON-formatted tag:`'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`. In alternativa, il flag `--tags` può essere utilizzato con tag in formato abbreviato: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

Output:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Test-Role",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role",
        "CreateDate": "2023-05-25T23:29:41+00:00",
        "AssumeRolePolicyDocument": {
            "Version":"2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
```

```
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
            {
                "Key": "Location",
                "Value": "Seattle"
            }
        ]
    }
}
```

Per ulteriori informazioni, consulta Applicazione di tag a ruoli IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sulle API, vedere CreateRolein AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
```

```
type RoleWrapper struct {
 IamClient *iam.Client
}



// CreateRole creates a role that trusts a specified user. The trusted user can
 assume
// the role to acquire its permissions.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper RoleWrapper) CreateRole(ctx context.Context, roleName string,
 trustedUserArn string) (*types.Role, error) {
 var role *types.Role
 trustPolicy := PolicyDocument{
  Version: "2012-10-17",
  Statement: []PolicyStatement{{
   Effect:    "Allow",
   Principal: map[string]string{"AWS": trustedUserArn},
   Action:    []string{"sts:AssumeRole"},
  }},
 }
 policyBytes, err := json.Marshal(trustPolicy)
 if err != nil {
  log.Printf("Couldn't create trust policy for %v. Here's why: %v\n",
 trustedUserArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreateRole(ctx, &iam.CreateRoleInput{
  AssumeRolePolicyDocument: aws.String(string(policyBytes)),
  RoleName:                 aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't create role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
}
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import software.amazon.awssdk.services.iam.model.CreateRoleRequest;
import software.amazon.awssdk.services.iam.model.CreateRoleResponse;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import java.io.FileReader;

/*
*   This example requires a trust policy document. For more information, see:
*   https://aws.amazon.com/blogs/security/how-to-use-trust-policies-with-iam-
roles/
*
*
*  In addition, set up your development environment, including your credentials.
*
*  For information, see this documentation topic:
*
*  https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class CreateRole {
    public static void main(String[] args) throws Exception {
        final String usage = """
                Usage:
                    <rolename> <fileLocation>\s

                Where:
                    rolename - The name of the role to create.\s
```

```
                    fileLocation - The location of the JSON document that
represents the trust policy.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String rolename = args[0];
        String fileLocation = args[1];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        String result = createIAMRole(iam, rolename, fileLocation);
        System.out.println("Successfully created user: " + result);
        iam.close();
    }

    public static String createIAMRole(IamClient iam, String rolename, String
fileLocation) throws Exception {
        try {
            JSONObject jsonObject = (JSONObject)
readJsonSimpleDemo(fileLocation);
            CreateRoleRequest request = CreateRoleRequest.builder()
                    .roleName(rolename)
                    .assumeRolePolicyDocument(jsonObject.toJSONString())
                    .description("Created using the AWS SDK for Java")
                    .build();

            CreateRoleResponse response = iam.createRole(request);
            System.out.println("The ARN of the role is " +
response.role().arn());

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

    public static Object readJsonSimpleDemo(String filename) throws Exception {
```

```
        FileReader reader = new FileReader(filename);
        JSONParser jsonParser = new JSONParser();
        return jsonParser.parse(reader);
    }
}
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea il ruolo.

```
import { CreateRoleCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 */
export const createRole = (roleName) => {
  const command = new CreateRoleCommand({
    AssumeRolePolicyDocument: JSON.stringify({
      Version: "2012-10-17",
      Statement: [
        {
          Effect: "Allow",
          Principal: {
            Service: "lambda.amazonaws.com",
          },
          Action: "sts:AssumeRole",
        },
      ],
```

```
    }),
    RoleName: roleName,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

$assumeRolePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{$user['Arn']}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }";
$assumeRoleRole = $service->createRole("iam_demo_role_$uuid",
 $assumeRolePolicyDocument);
echo "Created role: {$assumeRoleRole['RoleName']}\n";

    /**
     * @param string $roleName
     * @param string $rolePolicyDocument
     * @return array
     * @throws AwsException
```

```
     */
    public function createRole(string $roleName, string $rolePolicyDocument)
    {
        $result = $this->customWaiter(function () use ($roleName,
 $rolePolicyDocument) {
            return $this->iamClient->createRole([
                'AssumeRolePolicyDocument' => $rolePolicyDocument,
                'RoleName' => $roleName,
            ]);
        });
        return $result['Role'];
    }
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea un nuovo ruolo denominato **MyNewRole** e collega la policy trovata nel file **NewRoleTrustPolicy.json**. Tenere presente che per elaborare correttamente il file della policy JSON è necessario utilizzare il parametro di cambio **-Raw**. Il documento di policy visualizzato nell'output è codificato nell'URL. In questo esempio viene decodificato con il metodo **UrlDecode** .NET.

```
$results = New-IAMRole -AssumeRolePolicyDocument (Get-Content -raw
 NewRoleTrustPolicy.json) -RoleName MyNewRole
$results
```

Output:

```
Arn                       : arn:aws:iam::123456789012:role/MyNewRole
AssumeRolePolicyDocument : %7B%0D%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C
%0D%0A%20%20%22Statement%22
                             %3A%20%5B%0D%0A%20%20%20%20%7B%0D%0A
%20%20%20%20%20%20%22Sid%22%3A%20%22%22%2C
                             %0D%0A%20%20%20%20%20%20%22Effect%22%3A%20%22Allow
%22%2C%0D%0A%20%20%20%20%20%20
                             %22Principal%22%3A%20%7B%0D%0A
%20%20%20%20%20%20%20%20%22AWS%22%3A%20%22arn%3Aaws
```

```
                              %3Aiam%3A%3A123456789012%3ADavid%22%0D%0A
%20%20%20%20%20%20%7D%2C%0D%0A%20%20%20
                              %20%20%20%22Action%22%3A%20%22sts%3AAssumeRole%22%0D
%0A%20%20%20%20%7D%0D%0A%20
                              %20%5D%0D%0A%7D
CreateDate               : 4/15/2015 11:04:23 AM
Path                     : /
RoleId                   : V5PAJI2KPN4EAEXAMPLE1
RoleName                 : MyNewRole

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.AssumeRolePolicyDocument)
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:David"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Per i dettagli sull'API, vedere CreateRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea un nuovo ruolo denominato **`MyNewRole`** e collega la policy trovata nel file **`NewRoleTrustPolicy.json`**. Tenere presente che per elaborare correttamente il file della policy JSON è necessario utilizzare il parametro di cambio **`-Raw`**. Il documento di policy visualizzato nell'output è codificato nell'URL. In questo esempio viene decodificato con il metodo **`UrlDecode`** .NET.

```
$results = New-IAMRole -AssumeRolePolicyDocument (Get-Content -raw
 NewRoleTrustPolicy.json) -RoleName MyNewRole
$results
```

Output:

```
Arn                       : arn:aws:iam::123456789012:role/MyNewRole
AssumeRolePolicyDocument : %7B%0D%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C
%0D%0A%20%20%22Statement%22
                           %3A%20%5B%0D%0A%20%20%20%20%7B%0D%0A
%20%20%20%20%20%20%22Sid%22%3A%20%22%22%2C
                           %0D%0A%20%20%20%20%20%20%22Effect%22%3A%20%22Allow
%22%2C%0D%0A%20%20%20%20%20%20
                           %22Principal%22%3A%20%7B%0D%0A
%20%20%20%20%20%20%20%20%22AWS%22%3A%20%22arn%3Aaws
                           %3Aiam%3A%3A123456789012%3ADavid%22%0D%0A
%20%20%20%20%20%20%7D%2C%0D%0A%20%20%20
                           %20%20%20%22Action%22%3A%20%22sts%3AAssumeRole%22%0D
%0A%20%20%20%20%7D%0D%0A%20
                           %20%5D%0D%0A%7D
CreateDate                : 4/15/2015 11:04:23 AM
Path                      : /
RoleId                    : V5PAJI2KPN4EAEXAMPLE1
RoleName                  : MyNewRole

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.AssumeRolePolicyDocument)
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:David"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Per i dettagli sull'API, vedere CreateRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def create_role(role_name, allowed_services):
    """
    Creates a role that lets a list of specified services assume the role.

    :param role_name: The name of the role.
    :param allowed_services: The services that can assume the role.
    :return: The newly created role.
    """
    trust_policy = {
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {"Service": service},
                "Action": "sts:AssumeRole",
            }
            for service in allowed_services
        ],
    }

    try:
        role = iam.create_role(
            RoleName=role_name, AssumeRolePolicyDocument=json.dumps(trust_policy)
        )
        logger.info("Created role %s.", role.name)
    except ClientError:
        logger.exception("Couldn't create role %s.", role_name)
        raise
    else:
        return role
```

- Per i dettagli sull'API, consulta CreateRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
 # Creates a role and attaches policies to it.
 #
 # @param role_name [String] The name of the role.
 # @param assume_role_policy_document [Hash] The trust relationship policy
document.
 # @param policy_arns [Array<String>] The ARNs of the policies to attach.
 # @return [String, nil] The ARN of the new role if successful, or nil if an
error occurred.
 def create_role(role_name, assume_role_policy_document, policy_arns)
   response = @iam_client.create_role(
     role_name: role_name,
     assume_role_policy_document: assume_role_policy_document.to_json
   )
   role_arn = response.role.arn

   policy_arns.each do |policy_arn|
     @iam_client.attach_role_policy(
       role_name: role_name,
       policy_arn: policy_arn
     )
   end

   role_arn
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error creating role: #{e.message}")
   nil
 end
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn create_role(
    client: &iamClient,
    role_name: &str,
    role_policy_document: &str,
) -> Result<Role, iamError> {
    let response: CreateRoleOutput = loop {
        if let Ok(response) = client
            .create_role()
            .role_name(role_name)
            .assume_role_policy_document(role_policy_document)
            .send()
            .await
        {
            break response;
        }
    };

    Ok(response.role.unwrap())
}
```

- Per i dettagli sulle API, consulta la CreateRoleguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    oo_result = lo_iam->createrole(
      iv_rolename = iv_role_name
      iv_assumerolepolicydocument = iv_assume_role_policy_document ).
    MESSAGE 'Role created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'Role already exists.' TYPE 'E'.
  CATCH /aws1/cx_iammalformedplydocex.
    MESSAGE 'Assume role policy document is malformed.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Role limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, CreateRoleconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3
```

```
    public func createRole(name: String, policyDocument: String) async throws ->
String {
        let input = CreateRoleInput(
            assumeRolePolicyDocument: policyDocument,
            roleName: name
        )
        do {
            let output = try await client.createRole(input: input)
            guard let role = output.role else {
                throw ServiceHandlerError.noSuchRole
            }
            guard let id = role.roleId else {
                throw ServiceHandlerError.noSuchRole
            }
            return id
        } catch {
            print("ERROR: createRole:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la CreateRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usa **`CreateSAMLProvider` con un** AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateSAMLProvider`.

CLI

AWS CLI

Come creare un provider SAML

Questo esempio crea un nuovo provider SAML in IAM denominato `MySAMLProvider`. È descritto dal documento di metadati SAML che si trova nel file `SAMLMetaData.xml`.

```
aws iam create-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
```

```
    --name MySAMLProvider
```

Output:

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/MySAMLProvider"
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per informazioni dettagliate sull'API, consulta CreateSAMLProvider nella documentazione di riferimento dei comandi della AWS CLI .

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { CreateSAMLProviderCommand, IAMClient } from "@aws-sdk/client-iam";
import { readFileSync } from "node:fs";
import * as path from "node:path";
import { dirnameFromMetaUrl } from "@aws-doc-sdk-examples/lib/utils/util-fs.js";

const client = new IAMClient({});

/**
 * This sample document was generated using Auth0.
 * For more information on generating this document,
 see https://docs.aws.amazon.com/IAM/latest/UserGuide/
id_roles_providers_create_saml.html#samlstep1.
 */
const sampleMetadataDocument = readFileSync(
  path.join(
    dirnameFromMetaUrl(import.meta.url),
    "../../../../resources/sample_files/sample_saml_metadata.xml",
```

```
  ),
);

/**
 *
 * @param {*} providerName
 * @returns
 */
export const createSAMLProvider = async (providerName) => {
  const command = new CreateSAMLProviderCommand({
    Name: providerName,
    SAMLMetadataDocument: sampleMetadataDocument.toString(),
  });

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per informazioni dettagliate sull'API, consulta CreateSAMLProvider nella documentazione di riferimento dell'API AWS SDK per JavaScript .

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea una nuova entità per il provider SAML in IAM. È denominato **`MySAMLProvider`** ed è descritto dal documento di metadati SAML contenuto nel file **`SAMLMetaData.xml`**, che è stato scaricato separatamente dal sito web del provider di servizi SAML.

```
New-IAMSAMLProvider -Name MySAMLProvider -SAMLMetadataDocument (Get-Content -Raw
 SAMLMetaData.xml)
```

Output:

```
arn:aws:iam::123456789012:saml-provider/MySAMLProvider
```

- Per informazioni dettagliate sull'API, consulta CreateSAMLProvider nella documentazione di riferimento dei cmdlet di AWS Strumenti per PowerShell (V4).

Strumenti per PowerShell V5

Esempio 1: questo esempio crea una nuova entità per il provider SAML in IAM. È denominato **`MySAMLProvider`** ed è descritto dal documento di metadati SAML contenuto nel file **`SAMLMetaData.xml`**, che è stato scaricato separatamente dal sito web del provider di servizi SAML.

```
New-IAMSAMLProvider -Name MySAMLProvider -SAMLMetadataDocument (Get-Content -Raw
 SAMLMetaData.xml)
```

Output:

```
arn:aws:iam::123456789012:saml-provider/MySAMLProvider
```

- Per informazioni dettagliate sull'API, consulta CreateSAMLProvider nella documentazione di riferimento dei cmdlet di AWS Strumenti per PowerShell (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateServiceLinkedRole`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateServiceLinkedRole`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an IAM service-linked role.
    /// </summary>
    /// <param name="serviceName">The name of the AWS Service.</param>
```

```
    /// <param name="description">A description of the IAM service-linked role.</
param>
    /// <returns>The IAM role that was created.</returns>
    public async Task<Role> CreateServiceLinkedRoleAsync(string serviceName,
 string description)
    {
        var request = new CreateServiceLinkedRoleRequest
        {
            AWSServiceName = serviceName,
            Description = description
        };

        var response = await _IAMService.CreateServiceLinkedRoleAsync(request);
        return response.Role;
    }
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come creare un ruolo collegato a un servizio

L'`create-service-linked-role`esempio seguente crea un ruolo collegato al servizio per il AWS servizio specificato e allega la descrizione specificata.

```
aws iam create-service-linked-role \
    --aws-service-name lex.amazonaws.com \
    --description "My service-linked role to support Lex"
```

Output:

```
{
    "Role": {
        "Path": "/aws-service-role/lex.amazonaws.com/",
        "RoleName": "AWSServiceRoleForLexBots",
        "RoleId": "AROA1234567890EXAMPLE",
        "Arn": "arn:aws:iam::1234567890:role/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots",
```

```
        "CreateDate": "2019-04-17T20:34:14+00:00",
        "AssumeRolePolicyDocument": {
            "Version":"2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "sts:AssumeRole"
                    ],
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "lex.amazonaws.com"
                        ]
                    }
                }
            ]
        }
    }
}
```

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati a servizi nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, vedere CreateServiceLinkedRolein AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
```

```
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// CreateServiceLinkedRole creates a service-linked role that is owned by the
 specified service.
func (wrapper RoleWrapper) CreateServiceLinkedRole(ctx context.Context,
 serviceName string, description string) (
 *types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.CreateServiceLinkedRole(ctx,
 &iam.CreateServiceLinkedRoleInput{
  AWSServiceName: aws.String(serviceName),
  Description:    aws.String(description),
 })
 if err != nil {
  log.Printf("Couldn't create service-linked role %v. Here's why: %v\n",
 serviceName, err)
 } else {
  role = result.Role
 }
 return role, err
}
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea un ruolo collegato ai servizi.

```
import {
  CreateServiceLinkedRoleCommand,
  GetRoleCommand,
  IAMClient,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} serviceName
 */
export const createServiceLinkedRole = async (serviceName) => {
  const command = new CreateServiceLinkedRoleCommand({
    // For a list of AWS services that support service-linked roles,
    // see https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_aws-
services-that-work-with-iam.html.
    //
    // For a list of AWS service endpoints, see https://docs.aws.amazon.com/
general/latest/gr/aws-service-information.html.
    AWSServiceName: serviceName,
  });
  try {
    const response = await client.send(command);
    console.log(response);
    return response;
  } catch (caught) {
    if (
      caught instanceof Error &&
      caught.name === "InvalidInputException" &&
      caught.message.includes(
```

```
        "Service role name AWSServiceRoleForElasticBeanstalk has been taken in
 this account",
      )
    ) {
      console.warn(caught.message);
      return client.send(
        new GetRoleCommand({ RoleName: "AWSServiceRoleForElasticBeanstalk" }),
      );
    }
    throw caught;
  }
};
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function createServiceLinkedRole($awsServiceName, $customSuffix = "",
 $description = "")
    {
        $createServiceLinkedRoleArguments = ['AWSServiceName' =>
 $awsServiceName];
        if ($customSuffix) {
            $createServiceLinkedRoleArguments['CustomSuffix'] = $customSuffix;
        }
        if ($description) {
            $createServiceLinkedRoleArguments['Description'] = $description;
        }
```

```
        return $this->iamClient-
>createServiceLinkedRole($createServiceLinkedRoleArguments);
    }
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea un ruolo collegato ai servizi per il servizio di scalabilità automatica.

```
New-IAMServiceLinkedRole -AWSServiceName autoscaling.amazonaws.com -CustomSuffix
 RoleNameEndsWithThis -Description "My service-linked role to support
 autoscaling"
```

- Per i dettagli sull'API, vedere CreateServiceLinkedRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea un ruolo collegato ai servizi per il servizio di scalabilità automatica.

```
New-IAMServiceLinkedRole -AWSServiceName autoscaling.amazonaws.com -CustomSuffix
 RoleNameEndsWithThis -Description "My service-linked role to support
 autoscaling"
```

- Per i dettagli sull'API, vedere CreateServiceLinkedRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def create_service_linked_role(service_name, description):
    """
    Creates a service-linked role.

    :param service_name: The name of the service that owns the role.
    :param description: A description to give the role.
    :return: The newly created role.
    """
    try:
        response = iam.meta.client.create_service_linked_role(
            AWSServiceName=service_name, Description=description
        )
        role = iam.Role(response["Role"]["RoleName"])
        logger.info("Created service-linked role %s.", role.name)
    except ClientError:
        logger.exception("Couldn't create service-linked role for %s.",
 service_name)
        raise
    else:
        return role
```

- Per i dettagli sull'API, consulta CreateServiceLinkedRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
 # Creates a service-linked role
 #
 # @param service_name [String] The service name to create the role for.
 # @param description [String] The description of the service-linked role.
 # @param suffix [String] Suffix for customizing role name.
 # @return [String] The name of the created role
 def create_service_linked_role(service_name, description, suffix)
   response = @iam_client.create_service_linked_role(
     aws_service_name: service_name, description: description, custom_suffix:
suffix
   )
   role_name = response.role.role_name
   @logger.info("Created service-linked role #{role_name}.")
   role_name
 rescue Aws::Errors::ServiceError => e
   @logger.error("Couldn't create service-linked role for #{service_name}.
Here's why:")
   @logger.error("\t#{e.code}: #{e.message}")
   raise
 end
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn create_service_linked_role(
    client: &iamClient,
    aws_service_name: String,
    custom_suffix: Option<String>,
    description: Option<String>,
) -> Result<CreateServiceLinkedRoleOutput,
 SdkError<CreateServiceLinkedRoleError>> {
    let response = client
        .create_service_linked_role()
        .aws_service_name(aws_service_name)
        .set_custom_suffix(custom_suffix)
        .set_description(description)
        .send()
        .await?;

    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento CreateServiceLinkedRoleall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listpolicyversions(
          iv_policyarn = iv_policy_arn ).
        MESSAGE 'Retrieved policy versions list.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Policy does not exist.' TYPE 'E'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when listing policy versions.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, CreateServiceLinkedRoleconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func createServiceLinkedRole(service: String, suffix: String? = nil,
 description: String?)
                    async throws -> IAMClientTypes.Role {
        let input = CreateServiceLinkedRoleInput(
            awsServiceName: service,
            customSuffix: suffix,
            description: description
        )
        do {
            let output = try await client.createServiceLinkedRole(input: input)
            guard let role = output.role else {
                throw ServiceHandlerError.noSuchRole
            }
```

```
            return role
        } catch {
            print("ERROR: createServiceLinkedRole:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la CreateServiceLinkedRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateUser`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateUser`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Creazione di utenti di sola lettura e di lettura e scrittura

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
/// <summary>
/// Create an IAM user.
/// </summary>
/// <param name="userName">The username for the new IAM user.</param>
/// <returns>The IAM user that was created.</returns>
```

```
    public async Task<User> CreateUserAsync(string userName)
    {
        var response = await _IAMService.CreateUserAsync(new CreateUserRequest
 { UserName = userName });
        return response.User;
    }
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}
```

```
###############################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an AWS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user already exists, we don't want to try to create it.
  if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
  fi

  response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta CreateUser AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::CreateUserRequest create_request;
    create_request.SetUserName(userName);

    auto create_outcome = iam.CreateUser(create_request);
    if (!create_outcome.IsSuccess()) {
        std::cerr << "Error creating IAM user " << userName << ":" <<
                  create_outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully created IAM user " << userName << std::endl;
    }

    return create_outcome.IsSuccess();
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Esempio 1: Come creare un utente IAM

Il comando `create-user` seguente crea un utente IAM denominato Bob nell'account corrente.

```
aws iam create-user \
    --user-name Bob
```

Output:

```
{
    "User": {
        "UserName": "Bob",
        "Path": "/",
        "CreateDate": "2023-06-08T03:20:41.270Z",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Bob"
    }
}
```

Per ulteriori informazioni, consulta Creare un utente IAM nel tuo AWS account nella Guida per l'utente AWS IAM.

Esempio 2: Come creare un utente IAM in un percorso specificato

Il comando `create-user` seguente crea un utente IAM denominato Bob nel percorso specificato.

```
aws iam create-user \
    --user-name Bob \
    --path /division_abc/subdivision_xyz/
```

Output:

```
{
    "User": {
        "Path": "/division_abc/subdivision_xyz/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/division_abc/subdivision_xyz/Bob",
        "CreateDate": "2023-05-24T18:20:17+00:00"
    }
}
```

Per ulteriori informazioni, consulta Identificatori IAM nella Guida per l'utente di IAM AWS .

Esempio 3: Come creare un utente IAM con tag

Il comando `create-user` seguente crea un utente IAM denominato Bob con tag. Questo esempio utilizza il `--tags` parametro flag con i seguenti JSON-formatted tag: `'{"Key":`

`"Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`. In alternativa, il flag `--tags` può essere utilizzato con tag in formato abbreviato: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```
aws iam create-user \
    --user-name Bob \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

Output:

```
{
    "User": {
        "Path": "/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/Bob",
        "CreateDate": "2023-05-25T17:14:21+00:00",
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
            {
                "Key": "Location",
                "Value": "Seattle"
            }
        ]
    }
}
```

Per ulteriori informazioni, consulta Applicazione di tag a utenti IAM nella Guida per l'utente di IAM AWS .

Esempio 3: Come creare un utente IAM con un limite delle autorizzazioni impostato

Il `create-user` comando seguente crea un utente IAM denominato Bob con il limite delle autorizzazioni di. AmazonS3FullAccess

```
aws iam create-user \
    --user-name Bob \
    --permissions-boundary arn:aws:iam::aws:policy/AmazonS3FullAccess
```

Output:

```
{
    "User": {
        "Path": "/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/Bob",
        "CreateDate": "2023-05-24T17:50:53+00:00",
        "PermissionsBoundary": {
        "PermissionsBoundaryType": "Policy",
        "PermissionsBoundaryArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
        }
    }
}
```

Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta AWS CLI Command CreateUserReference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
```

```
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// CreateUser creates a new user with the specified name.
func (wrapper UserWrapper) CreateUser(ctx context.Context, userName string)
 (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.CreateUser(ctx, &iam.CreateUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
 } else {
  user = result.User
 }
 return user, err
}
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.core.waiters.WaiterResponse;
import software.amazon.awssdk.services.iam.model.CreateUserRequest;
import software.amazon.awssdk.services.iam.model.CreateUserResponse;
```

```
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.waiters.IamWaiter;
import software.amazon.awssdk.services.iam.model.GetUserRequest;
import software.amazon.awssdk.services.iam.model.GetUserResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateUser {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <username>\s

                Where:
                    username - The name of the user to create.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String username = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        String result = createIAMUser(iam, username);
        System.out.println("Successfully created user: " + result);
        iam.close();
    }

    public static String createIAMUser(IamClient iam, String username) {
```

```
        try {
            // Create an IamWaiter object.
            IamWaiter iamWaiter = iam.waiter();

            CreateUserRequest request = CreateUserRequest.builder()
                    .userName(username)
                    .build();

            CreateUserResponse response = iam.createUser(request);

            // Wait until the user is created.
            GetUserRequest userRequest = GetUserRequest.builder()
                    .userName(response.user().userName())
                    .build();

            WaiterResponse<GetUserResponse> waitUntilUserExists =
 iamWaiter.waitUntilUserExists(userRequest);

 waitUntilUserExists.matched().response().ifPresent(System.out::println);
            return response.user().userName();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Creare l'utente.

```
import { CreateUserCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} name
 */
export const createUser = (name) => {
  const command = new CreateUserCommand({ UserName: name });
  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, CreateUserconsulta AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  UserName: process.argv[2],
};

iam.getUser(params, function (err, data) {
  if (err && err.code === "NoSuchEntity") {
    iam.createUser(params, function (err, data) {
```

```
      if (err) {
        console.log("Error", err);
      } else {
        console.log("Success", data);
      }
    });
  } else {
    console.log(
      "User " + process.argv[2] + " already exists",
      data.User.UserId
    );
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, CreateUserconsulta AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun createIAMUser(usernameVal: String?): String? {
    val request =
        CreateUserRequest {
            userName = usernameVal
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createUser(request)
        return response.user?.userName
    }
}
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

$user = $service->createUser("iam_demo_user_$uuid");
echo "Created user with the arn: {$user['Arn']}\n";


    /**
     * @param string $name
     * @return array
     * @throws AwsException
     */
    public function createUser(string $name): array
    {
        $result = $this->iamClient->createUser([
            'UserName' => $name,
        ]);

        return $result['User'];
    }
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea un utente IAM denominato **Bob**. Se Bob deve accedere alla AWS console, è necessario eseguire separatamente il comando **New-IAMLoginProfile** per creare un profilo di accesso con una password. Se Bob deve eseguire AWS PowerShell comandi CLI multipiattaforma o AWS effettuare chiamate API, è necessario eseguire separatamente **New-IAMAccessKey** il comando per creare le chiavi di accesso.

```
New-IAMUser -UserName Bob
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Bob
CreateDate       : 4/22/2015 12:02:11 PM
PasswordLastUsed : 1/1/0001 12:00:00 AM
Path             : /
UserId           : AIDAJWGEFDMEMEXAMPLE1
UserName         : Bob
```

- Per i dettagli sull'API, vedere CreateUserin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea un utente IAM denominato **Bob**. Se Bob deve accedere alla AWS console, è necessario eseguire separatamente il comando **New-IAMLoginProfile** per creare un profilo di accesso con una password. Se Bob deve eseguire AWS PowerShell comandi CLI multipiattaforma o AWS effettuare chiamate API, è necessario eseguire separatamente **New-IAMAccessKey** il comando per creare le chiavi di accesso.

```
New-IAMUser -UserName Bob
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Bob
CreateDate       : 4/22/2015 12:02:11 PM
PasswordLastUsed : 1/1/0001 12:00:00 AM
Path             : /
```

```
UserId            : AIDAJWGEFDMEMEXAMPLE1
UserName          : Bob
```

- Per i dettagli sull'API, vedere CreateUserin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def create_user(user_name):
    """
    Creates a user. By default, a user has no permissions or access keys.

    :param user_name: The name of the user.
    :return: The newly created user.
    """
    try:
        user = iam.create_user(UserName=user_name)
        logger.info("Created user %s.", user.name)
    except ClientError:
        logger.exception("Couldn't create user %s.", user_name)
        raise
    else:
        return user
```

- Per i dettagli sull'API, consulta CreateUser AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
 # Creates a user and their login profile
 #
 # @param user_name [String] The name of the user
 # @param initial_password [String] The initial password for the user
 # @return [String, nil] The ID of the user if created, or nil if an error
occurred
 def create_user(user_name, initial_password)
   response = @iam_client.create_user(user_name: user_name)
   @iam_client.wait_until(:user_exists, user_name: user_name)
   @iam_client.create_login_profile(
     user_name: user_name,
     password: initial_password,
     password_reset_required: true
   )
   @logger.info("User '#{user_name}' created successfully.")
   response.user.user_id
 rescue Aws::IAM::Errors::EntityAlreadyExists
   @logger.error("Error creating user '#{user_name}': user already exists.")
   nil
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error creating user '#{user_name}': #{e.message}")
   nil
 end
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn create_user(client: &iamClient, user_name: &str) -> Result<User,
 iamError> {
    let response = client.create_user().user_name(user_name).send().await?;

    Ok(response.user.unwrap())
}
```

- Per i dettagli sulle API, consulta il riferimento CreateUserall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->createuser(
          iv_username = iv_user_name ).
        MESSAGE 'User created successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamentityalrdyexex.
        MESSAGE 'User already exists.' TYPE 'E'.
      CATCH /aws1/cx_iamlimitexceededex.
        MESSAGE 'Limit exceeded for IAM users.' TYPE 'E'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Entity does not exist.' TYPE 'E'.
```

```
    ENDTRY.
```

- Per i dettagli sulle API, CreateUserconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func createUser(name: String) async throws -> String {
        let input = CreateUserInput(
            userName: name
        )
        do {
            let output = try await client.createUser(input: input)
            guard let user = output.user else {
                throw ServiceHandlerError.noSuchUser
            }
            guard let id = user.userId else {
                throw ServiceHandlerError.noSuchUser
            }
            return id
        } catch {
            print("ERROR: createUser:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la CreateUserguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`CreateVirtualMfaDevice`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateVirtualMfaDevice`.

CLI

AWS CLI

Per creare un dispositivo MFA virtuale

Questo esempio crea un nuovo dispositivo MFA virtuale chiamato `BobsMFADevice`. Crea un file che contiene le informazioni di bootstrap chiamate `QRCode.png` e le inserisce nella directory `C:/`. Il metodo bootstrap utilizzato in questo esempio è `QRCodePNG`.

```
aws iam create-virtual-mfa-device \
    --virtual-mfa-device-name BobsMFADevice \
    --outfile C:/QRCode.png \
    --bootstrap-method QRCodePNG
```

Output:

```
{
    "VirtualMFADevice": {
        "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"
}
```

Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella AWS Guida per l'utente IAM.

- Per i dettagli sull'API, consulta CreateVirtualMfaDevice AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea un nuovo dispositivo MFA virtuale. Le righe 2 e 3 estraggono il valore **`Base32StringSeed`** necessario al programma software MFA virtuale per creare un

account (in alternativa al codice QR). Dopo aver configurato il programma con il valore, ottieni due codici di autenticazione sequenziali dal programma. Infine, utilizza l'ultimo comando per collegare il dispositivo MFA virtuale all'utente IAM **Bob** e sincronizzare l'account con i due codici di autenticazione.

```
$Device = New-IAMVirtualMFADevice -VirtualMFADeviceName BobsMFADevice
$SR = New-Object System.IO.StreamReader($Device.Base32StringSeed)
$base32stringseed = $SR.ReadToEnd()
$base32stringseed
CZWZMCQNW4DEXAMPLE3VOUGXJFZYSUW7EXAMPLECR4NJFD65GX2SLUDW2EXAMPLE
```

Output:

```
-- Pause here to enter base-32 string seed code into virtual MFA program to
 register account. --

Enable-IAMMFADevice -SerialNumber $Device.SerialNumber -UserName Bob -
AuthenticationCode1 123456 -AuthenticationCode2 789012
```

Esempio 2: questo esempio crea un nuovo dispositivo MFA virtuale. Le righe 2 e 3 estraggono il valore **QRCodePNG** e lo scrivono in un file. Questa immagine può essere scansionata dal programma software MFA virtuale per creare un account (in alternativa all'immissione manuale Base32StringSeed del valore). Dopo aver creato l'account nel tuo programma MFA virtuale, ottieni due codici di autenticazione sequenziali e inseriscili negli ultimi comandi per collegare il dispositivo MFA virtuale all'utente IAM **Bob** e sincronizzare l'account.

```
$Device = New-IAMVirtualMFADevice -VirtualMFADeviceName BobsMFADevice
$BR = New-Object System.IO.BinaryReader($Device.QRCodePNG)
$BR.ReadBytes($BR.BaseStream.Length) | Set-Content -Encoding Byte -Path
 QRCode.png
```

Output:

```
 -- Pause here to scan PNG with virtual MFA program to register account. --

Enable-IAMMFADevice -SerialNumber $Device.SerialNumber -UserName Bob -
AuthenticationCode1 123456 -AuthenticationCode2 789012
```

- Per i dettagli sull'API, vedere CreateVirtualMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea un nuovo dispositivo MFA virtuale. Le righe 2 e 3 estraggono il valore **`Base32StringSeed`** necessario al programma software MFA virtuale per creare un account (in alternativa al codice QR). Dopo aver configurato il programma con il valore, ottieni due codici di autenticazione sequenziali dal programma. Infine, utilizza l'ultimo comando per collegare il dispositivo MFA virtuale all'utente IAM **`Bob`** e sincronizzare l'account con i due codici di autenticazione.

```
$Device = New-IAMVirtualMFADevice -VirtualMFADeviceName BobsMFADevice
$SR = New-Object System.IO.StreamReader($Device.Base32StringSeed)
$base32stringseed = $SR.ReadToEnd()
$base32stringseed
CZWZMCQNW4DEXAMPLE3VOUGXJFZYSUW7EXAMPLECR4NJFD65GX2SLUDW2EXAMPLE
```

Output:

```
-- Pause here to enter base-32 string seed code into virtual MFA program to
 register account. --

Enable-IAMMFADevice -SerialNumber $Device.SerialNumber -UserName Bob -
AuthenticationCode1 123456 -AuthenticationCode2 789012
```

Esempio 2: questo esempio crea un nuovo dispositivo MFA virtuale. Le righe 2 e 3 estraggono il valore **`QRCodePNG`** e lo scrivono in un file. Questa immagine può essere scansionata dal programma software MFA virtuale per creare un account (in alternativa all'immissione manuale Base32StringSeed del valore). Dopo aver creato l'account nel tuo programma MFA virtuale, ottieni due codici di autenticazione sequenziali e inseriscili negli ultimi comandi per collegare il dispositivo MFA virtuale all'utente IAM **`Bob`** e sincronizzare l'account.

```
$Device = New-IAMVirtualMFADevice -VirtualMFADeviceName BobsMFADevice
$BR = New-Object System.IO.BinaryReader($Device.QRCodePNG)
$BR.ReadBytes($BR.BaseStream.Length) | Set-Content -Encoding Byte -Path
 QRCode.png
```

Output:

```
 -- Pause here to scan PNG with virtual MFA program to register account. --
```

```
Enable-IAMMFADevice -SerialNumber $Device.SerialNumber -UserName Bob -
AuthenticationCode1 123456 -AuthenticationCode2 789012
```

- Per i dettagli sull'API, vedere CreateVirtualMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DeactivateMfaDevice`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DeactivateMfaDevice`.

CLI

AWS CLI

Per disattivare un dispositivo MFA

Questo comando disattiva il dispositivo MFA virtuale con l'ARN `arn:aws:iam::210987654321:mfa/BobsMFADevice` associato all'utente `Bob`.

```
aws iam deactivate-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella AWS Guida per l'utente IAM.

- Per i dettagli sull'API, consulta DeactivateMfaDevice AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo comando disabilita il dispositivo hardware MFA associato all'utente **`Bob`** con il numero di serie **`123456789012`**.

```
Disable-IAMMFADevice -UserName "Bob" -SerialNumber "123456789012"
```

Esempio 2: questo comando disabilita il dispositivo MFA virtuale associato all'utente **David** con ARN **arn:aws:iam::210987654321:mfa/David**. Tieni presente che il dispositivo MFA virtuale non viene eliminato dall'account. Il dispositivo virtuale è ancora presente e appare nell'output del comando **Get-IAMVirtualMFADevice**. Prima di poter creare un nuovo dispositivo MFA virtuale per lo stesso utente, è necessario eliminare quello precedente utilizzando il comando **Remove-IAMVirtualMFADevice**.

```
Disable-IAMMFADevice -UserName "David" -SerialNumber
 "arn:aws:iam::210987654321:mfa/David"
```

- Per i dettagli sull'API, vedere DeactivateMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando disabilita il dispositivo hardware MFA associato all'utente **Bob** con il numero di serie **123456789012**.

```
Disable-IAMMFADevice -UserName "Bob" -SerialNumber "123456789012"
```

Esempio 2: questo comando disabilita il dispositivo MFA virtuale associato all'utente **David** con ARN **arn:aws:iam::210987654321:mfa/David**. Tieni presente che il dispositivo MFA virtuale non viene eliminato dall'account. Il dispositivo virtuale è ancora presente e appare nell'output del comando **Get-IAMVirtualMFADevice**. Prima di poter creare un nuovo dispositivo MFA virtuale per lo stesso utente, è necessario eliminare quello precedente utilizzando il comando **Remove-IAMVirtualMFADevice**.

```
Disable-IAMMFADevice -UserName "David" -SerialNumber
 "arn:aws:iam::210987654321:mfa/David"
```

- Per i dettagli sull'API, vedere DeactivateMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeleteAccessKey`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteAccessKey`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Creazione di utenti di sola lettura e di lettura e scrittura
- Gestione delle chiavi di accesso

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
   /// <summary>
   /// Delete an IAM user's access key.
   /// </summary>
   /// <param name="accessKeyId">The Id for the IAM access key.</param>
   /// <param name="userName">The username of the user that owns the IAM
   /// access key.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> DeleteAccessKeyAsync(string accessKeyId, string
userName)
   {
       var response = await _IAMService.DeleteAccessKeyAsync(new
DeleteAccessKeyRequest
       {
           AccessKeyId = accessKeyId,
           UserName = userName,
       });

       return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
   }
```

- Per i dettagli sull'API, consulta la DeleteAccessKeysezione AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function iam_delete_access_key() {
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.
```

```
 # bashsupport disable=BP5008
 function usage() {
   echo "function iam_delete_access_key"
   echo "Deletes an AWS Identity and Access Management (IAM) access key for the
specified IAM user"
   echo "  -u user_name    The name of the user."
   echo "  -k access_key   The access key to delete."
   echo ""
 }

 # Retrieve the calling parameters.
 while getopts "u:k:h" option; do
   case "${option}" in
     u) user_name="${OPTARG}" ;;
     k) access_key="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$user_name" ]]; then
   errecho "ERROR: You must provide a username with the -u parameter."
   usage
   return 1
 fi

 if [[ -z "$access_key" ]]; then
   errecho "ERROR: You must provide an access key with the -k parameter."
   usage
   return 1
 fi

 iecho "Parameters:\n"
 iecho "    Username:   $user_name"
 iecho "    Access key:   $access_key"
 iecho ""
```

```
  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}
```

- Per i dettagli sull'API, consulta DeleteAccessKey AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::deleteAccessKey(const Aws::String &userName,
                                  const Aws::String &accessKeyID,
                                  const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::DeleteAccessKeyRequest request;
    request.SetUserName(userName);
    request.SetAccessKeyId(accessKeyID);
```

```
    auto outcome = iam.DeleteAccessKey(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error deleting access key " << accessKeyID << " from user "
                  << userName << ": " << outcome.GetError().GetMessage() <<
                  std::endl;
    }
    else {
        std::cout << "Successfully deleted access key " << accessKeyID
                  << " for IAM user " << userName << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la DeleteAccessKeysezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come eliminare una chiave di accesso per un utente IAM

Il comando `delete-access-key` seguente elimina la chiave di accesso specificata (ID chiave di accesso e chiave di accesso segreta) per l'utente IAM denominato Bob.

```
aws iam delete-access-key \
    --access-key-id AKIDPMS9RO4H3FEXAMPLE \
    --user-name Bob
```

Questo comando non produce alcun output.

Per elencare le chiavi di accesso definite per un utente IAM, usa il comando `list-access-keys`.

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteAccessKey AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// DeleteAccessKey deletes an access key from a user.
func (wrapper UserWrapper) DeleteAccessKey(ctx context.Context, userName string,
 keyId string) error {
 _, err := wrapper.IamClient.DeleteAccessKey(ctx, &iam.DeleteAccessKeyInput{
  AccessKeyId: aws.String(keyId),
  UserName:    aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete access key %v. Here's why: %v\n", keyId, err)
 }
 return err
```

```
}
```

- Per i dettagli sull'API, consulta la DeleteAccessKeysezione AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.DeleteAccessKeyRequest;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteAccessKey {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <username> <accessKey>\s

                Where:
                    username - The name of the user.\s
```

```
                    accessKey - The access key ID for the secret access key you
 want to delete.\s
                """";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String username = args[0];
        String accessKey = args[1];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();
        deleteKey(iam, username, accessKey);
        iam.close();
    }

    public static void deleteKey(IamClient iam, String username, String
 accessKey) {
        try {
            DeleteAccessKeyRequest request = DeleteAccessKeyRequest.builder()
                    .accessKeyId(accessKey)
                    .userName(username)
                    .build();

            iam.deleteAccessKey(request);
            System.out.println("Successfully deleted access key " + accessKey +
                    " from user " + username);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la DeleteAccessKeysezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elimina la chiave di accesso.

```
import { DeleteAccessKeyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} userName
 * @param {string} accessKeyId
 */
export const deleteAccessKey = (userName, accessKeyId) => {
  const command = new DeleteAccessKeyCommand({
    AccessKeyId: accessKeyId,
    UserName: userName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la DeleteAccessKeysezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  AccessKeyId: "ACCESS_KEY_ID",
  UserName: "USER_NAME",
};

iam.deleteAccessKey(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la DeleteAccessKeysezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun deleteKey(
    userNameVal: String,
    accessKey: String,
) {
```

```
    val request =
        DeleteAccessKeyRequest {
            accessKeyId = accessKey
            userName = userNameVal
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.deleteAccessKey(request)
        println("Successfully deleted access key $accessKey from $userNameVal")
    }
}
```

- Per i dettagli sull'API, DeleteAccessKeyconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: Questo esempio elimina la coppia di chiavi di AWS accesso con l'ID della chiave **AKIAIOSFODNN7EXAMPLE** dall'utente denominato**Bob**.

```
Remove-IAMAccessKey -AccessKeyId AKIAIOSFODNN7EXAMPLE -UserName Bob -Force
```

- Per i dettagli sull'API, vedere DeleteAccessKeyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio elimina la coppia di chiavi di AWS accesso con l'ID della chiave **AKIAIOSFODNN7EXAMPLE** dall'utente denominato**Bob**.

```
Remove-IAMAccessKey -AccessKeyId AKIAIOSFODNN7EXAMPLE -UserName Bob -Force
```

- Per i dettagli sull'API, vedere DeleteAccessKeyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def delete_key(user_name, key_id):
    """
    Deletes a user's access key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to delete.
    """

    try:
        key = iam.AccessKey(user_name, key_id)
        key.delete()
        logger.info("Deleted access key %s for %s.", key.id, key.user_name)
    except ClientError:
        logger.exception("Couldn't delete key %s for %s", key_id, user_name)
        raise
```

- Per i dettagli sull'API, consulta DeleteAccessKey AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, disattiva ed elimina le chiavi di accesso.

```
# Manages access keys for IAM users
class AccessKeyManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'AccessKeyManager'
  end

  # Lists access keys for a user
  #
  # @param user_name [String] The name of the user.
  def list_access_keys(user_name)
    response = @iam_client.list_access_keys(user_name: user_name)
    if response.access_key_metadata.empty?
      @logger.info("No access keys found for user '#{user_name}'.")
    else
      response.access_key_metadata.map(&:access_key_id)
    end
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Error listing access keys: cannot find user '#{user_name}'.")
    []
  rescue StandardError => e
    @logger.error("Error listing access keys: #{e.message}")
    []
  end

  # Creates an access key for a user
  #
  # @param user_name [String] The name of the user.
  # @return [Boolean]
  def create_access_key(user_name)
    response = @iam_client.create_access_key(user_name: user_name)
    access_key = response.access_key
    @logger.info("Access key created for user '#{user_name}':
 #{access_key.access_key_id}")
    access_key
  rescue Aws::IAM::Errors::LimitExceeded
    @logger.error('Error creating access key: limit exceeded. Cannot create
 more.')
    nil
  rescue StandardError => e
    @logger.error("Error creating access key: #{e.message}")
```

```
    nil
  end

  # Deactivates an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def deactivate_access_key(user_name, access_key_id)
    @iam_client.update_access_key(
      user_name: user_name,
      access_key_id: access_key_id,
      status: 'Inactive'
    )
    true
  rescue StandardError => e
    @logger.error("Error deactivating access key: #{e.message}")
    false
  end

  # Deletes an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def delete_access_key(user_name, access_key_id)
    @iam_client.delete_access_key(
      user_name: user_name,
      access_key_id: access_key_id
    )
    true
  rescue StandardError => e
    @logger.error("Error deleting access key: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta la DeleteAccessKeysezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn delete_access_key(
    client: &iamClient,
    user: &User,
    key: &AccessKey,
) -> Result<(), iamError> {
    loop {
        match client
            .delete_access_key()
            .user_name(user.user_name())
            .access_key_id(key.access_key_id())
            .send()
            .await
        {
            Ok(_) => {
                break;
            }
            Err(e) => {
                println!("Can't delete the access key: {:?}", e);
                sleep(Duration::from_secs(2)).await;
            }
        }
    }
    Ok(())
}
```

- Per i dettagli sulle API, consulta la DeleteAccessKeyguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        lo_iam->deleteaccesskey(
          iv_accesskeyid = iv_access_key_id
          iv_username = iv_user_name ).
        MESSAGE 'Access key deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Access key or user does not exist.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DeleteAccessKeyconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func deleteAccessKey(user: IAMClientTypes.User? = nil,
                                key: IAMClientTypes.AccessKey) async throws
    {
        let userName: String?
```

```
        if user != nil {
            userName = user!.userName
        } else {
            userName = nil
        }

        let input = DeleteAccessKeyInput(
            accessKeyId: key.accessKeyId,
            userName: userName
        )
        do {
            _ = try await iamClient.deleteAccessKey(input: input)
        } catch {
            print("ERROR: deleteAccessKey:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteAccessKeyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeleteAccountAlias`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteAccountAlias`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::deleteAccountAlias(const Aws::String &accountAlias,
                                     const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::DeleteAccountAliasRequest request;
    request.SetAccountAlias(accountAlias);

    const auto outcome = iam.DeleteAccountAlias(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error deleting account alias " << accountAlias << ": "
                  << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully deleted account alias " << accountAlias <<
                  std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la DeleteAccountAliassezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come eliminare l'alias di un account

Il comando `delete-account-alias` seguente rimuove l'alias `mycompany` per l'account corrente.

```
aws iam delete-account-alias \
    --account-alias mycompany
```

Questo comando non produce alcun output.

Per maggiori informazioni, consulta l'ID AWS del tuo account e il suo alias nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta DeleteAccountAlias AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.DeleteAccountAliasRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteAccountAlias {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:
                    <alias>\s

                Where:
                    alias - The account alias to delete.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alias = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        deleteIAMAccountAlias(iam, alias);
        iam.close();
    }

    public static void deleteIAMAccountAlias(IamClient iam, String alias) {
        try {
            DeleteAccountAliasRequest request =
 DeleteAccountAliasRequest.builder()
                    .accountAlias(alias)
                    .build();

            iam.deleteAccountAlias(request);
            System.out.println("Successfully deleted account alias " + alias);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

- Per i dettagli sull'API, consulta la DeleteAccountAliassezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elimina l'alias dell'account.

```
import { DeleteAccountAliasCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} alias
 */
export const deleteAccountAlias = (alias) => {
  const command = new DeleteAccountAliasCommand({ AccountAlias: alias });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la DeleteAccountAliassezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
```

```
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.deleteAccountAlias({ AccountAlias: process.argv[2] }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la DeleteAccountAliassezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun deleteIAMAccountAlias(alias: String) {
    val request =
        DeleteAccountAliasRequest {
            accountAlias = alias
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.deleteAccountAlias(request)
        println("Successfully deleted account alias $alias")
    }
}
```

- Per i dettagli sull'API, DeleteAccountAliasconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: Questo esempio rimuove l'alias dell'account dal tuo. Account AWS La pagina di accesso dell'utente con l'alias at https://mycompanyaws.signin.aws.amazon.com/console non funziona più. Devi invece utilizzare l'URL originale con il tuo numero Account AWS ID all'indirizzo https://<accountidnumber>.signin.aws.amazon. com/console.

```
Remove-IAMAccountAlias -AccountAlias mycompanyaws
```

- Per i dettagli sull'API, vedere DeleteAccountAliasin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio rimuove l'alias dell'account dal tuo. Account AWS La pagina di accesso dell'utente con l'alias at https://mycompanyaws.signin.aws.amazon.com/console non funziona più. Devi invece utilizzare l'URL originale con il tuo numero Account AWS ID all'indirizzo https://<accountidnumber>.signin.aws.amazon. com/console.

```
Remove-IAMAccountAlias -AccountAlias mycompanyaws
```

- Per i dettagli sull'API, vedere DeleteAccountAliasin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def delete_alias(alias):
    """
    Removes the alias from the current account.

    :param alias: The alias to remove.
    """
    try:
        iam.meta.client.delete_account_alias(AccountAlias=alias)
        logger.info("Removed alias '%s' from your account.", alias)
    except ClientError:
        logger.exception("Couldn't remove alias '%s' from your account.", alias)
        raise
```

- Per i dettagli sull'API, consulta DeleteAccountAlias AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca, crea ed elimina gli alias degli account.

```
class IAMAliasManager
  # Initializes the IAM client and logger
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists available AWS account aliases.
```

```
  def list_aliases
    response = @iam_client.list_account_aliases

    if response.account_aliases.count.positive?
      @logger.info('Account aliases are:')
      response.account_aliases.each { |account_alias| @logger.info("
 #{account_alias}") }
    else
      @logger.info('No account aliases found.')
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing account aliases: #{e.message}")
  end

  # Creates an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to create.
  # @return [Boolean] true if the account alias was created; otherwise, false.
  def create_account_alias(account_alias)
    @iam_client.create_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating account alias: #{e.message}")
    false
  end

  # Deletes an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to delete.
  # @return [Boolean] true if the account alias was deleted; otherwise, false.
  def delete_account_alias(account_alias)
    @iam_client.delete_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting account alias: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta la DeleteAccountAliassezione AWS SDK per Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    lo_iam->deleteaccountalias(
      iv_accountalias = iv_account_alias ).
    MESSAGE 'Account alias deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Account alias does not exist.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, DeleteAccountAliasconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DeleteAccountPasswordPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteAccountPasswordPolicy`.

CLI

AWS CLI

Per eliminare la policy delle password dell'account corrente

Il comando `delete-account-password-policy` seguente rimuove la policy delle password per l'account corrente.

```
aws iam delete-account-password-policy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Impostazione di una policy delle password dell'account per utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteAccountPasswordPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio elimina la politica delle password per Account AWS e ripristina tutti i valori ai valori predefiniti originali. Se attualmente non esiste una politica in materia di password, viene visualizzato il seguente messaggio di errore: Impossibile trovare la politica dell'account con il nome PasswordPolicy .

```
Remove-IAMAccountPasswordPolicy
```

- Per i dettagli sull'API, vedere DeleteAccountPasswordPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio elimina la politica delle password per Account AWS e ripristina tutti i valori ai valori predefiniti originali. Se attualmente non esiste una politica in materia di password, viene visualizzato il seguente messaggio di errore: Impossibile trovare la politica dell'account con il nome PasswordPolicy .

```
Remove-IAMAccountPasswordPolicy
```

- Per i dettagli sull'API, vedere DeleteAccountPasswordPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeleteGroup`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteGroup`.

CLI

AWS CLI

Come eliminare un gruppo IAM

Il comando `delete-group` seguente elimina un gruppo IAM denominato `MyTestGroup`.

```
aws iam delete-group \
    --group-name MyTestGroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Eliminazione di un gruppo di utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteGroup AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { DeleteGroupCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} groupName
 */
export const deleteGroup = async (groupName) => {
  const command = new DeleteGroupCommand({
    GroupName: groupName,
  });
```

```
  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per i dettagli sull'API, consulta la DeleteGroupsezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il gruppo IAM denominato **`MyTestGroup`**. Il primo comando rimuove tutti gli utenti IAM che sono membri del gruppo e il secondo comando elimina il gruppo IAM. Entrambi i comandi funzionano senza alcuna richiesta di conferma.

```
(Get-IAMGroup -GroupName MyTestGroup).Users | Remove-IAMUserFromGroup -GroupName
 MyTestGroup -Force
Remove-IAMGroup -GroupName MyTestGroup -Force
```

- Per i dettagli sull'API, vedere DeleteGroupin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina il gruppo IAM denominato **`MyTestGroup`**. Il primo comando rimuove tutti gli utenti IAM che sono membri del gruppo e il secondo comando elimina il gruppo IAM. Entrambi i comandi funzionano senza alcuna richiesta di conferma.

```
(Get-IAMGroup -GroupName MyTestGroup).Users | Remove-IAMUserFromGroup -GroupName
 MyTestGroup -Force
Remove-IAMGroup -GroupName MyTestGroup -Force
```

- Per i dettagli sull'API, vedere DeleteGroupin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DeleteGroupPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteGroupPolicy`.

CLI

AWS CLI

Come eliminare una policy da un gruppo IAM

Il comando `delete-group-policy` seguente elimina la policy denominata `ExamplePolicy` dal gruppo denominato `Admins`.

```
aws iam delete-group-policy \
    --group-name Admins \
    --policy-name ExamplePolicy
```

Questo comando non produce alcun output.

Per visualizzare le policy collegate a un gruppo, usa il comando `list-group-policies`.

Per ulteriori informazioni, consulta Gestione delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta DeleteGroupPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio rimuove la policy in linea denominata **`TesterPolicy`** dal gruppo IAM **`Testers`**. Gli utenti di quel gruppo perdono immediatamente le autorizzazioni definite in tale policy.

```
Remove-IAMGroupPolicy -GroupName Testers -PolicyName TestPolicy
```

- Per i dettagli sull'API, vedere DeleteGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio rimuove la policy in linea denominata **`TesterPolicy`** dal gruppo IAM **`Testers`**. Gli utenti di quel gruppo perdono immediatamente le autorizzazioni definite in tale policy.

```
Remove-IAMGroupPolicy -GroupName Testers -PolicyName TestPolicy
```

- Per i dettagli sull'API, vedere DeleteGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeleteInstanceProfile`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteInstanceProfile`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione e gestione di un servizio resiliente
- Guida introduttiva allo streaming gestito
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Detaches a role from an instance profile, detaches policies from the
role,
```

```
    /// and deletes all the resources.
    /// </summary>
    /// <param name="profileName">The name of the profile to delete.</param>
    /// <param name="roleName">The name of the role to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteInstanceProfile(string profileName, string roleName)
    {
        try
        {
            await _amazonIam.RemoveRoleFromInstanceProfileAsync(
                new RemoveRoleFromInstanceProfileRequest()
                {
                    InstanceProfileName = profileName,
                    RoleName = roleName
                });
            await _amazonIam.DeleteInstanceProfileAsync(
                new DeleteInstanceProfileRequest() { InstanceProfileName =
profileName });
            var attachedPolicies = await
_amazonIam.ListAttachedRolePoliciesAsync(
                new ListAttachedRolePoliciesRequest() { RoleName = roleName });
            foreach (var policy in attachedPolicies.AttachedPolicies)
            {
                await _amazonIam.DetachRolePolicyAsync(
                    new DetachRolePolicyRequest()
                    {
                        RoleName = roleName,
                        PolicyArn = policy.PolicyArn
                    });
                // Delete the custom policies only.
                if (!policy.PolicyArn.StartsWith("arn:aws:iam::aws"))
                {
                    await _amazonIam.DeletePolicyAsync(
                        new Amazon.IdentityManagement.Model.DeletePolicyRequest()
                        {
                            PolicyArn = policy.PolicyArn
                        });
                }
            }

            await _amazonIam.DeleteRoleAsync(
                new DeleteRoleRequest() { RoleName = roleName });
        }
        catch (NoSuchEntityException)
```

```
        {
            Console.WriteLine($"Instance profile {profileName} does not exist.");
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteInstanceProfilesezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come eliminare un profilo dell'istanza

Il comando `delete-instance-profile` seguente elimina un profilo dell'istanza denominato `ExampleInstanceProfile`.

```
aws iam delete-instance-profile \
    --instance-profile-name ExampleInstanceProfile
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteInstanceProfile AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    const client = new IAMClient({});
```

```
      await client.send(
        new DeleteInstanceProfileCommand({
          InstanceProfileName: NAMES.instanceProfileName,
        }),
      );
```

- Per i dettagli sull'API, consulta la DeleteInstanceProfilesezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il profilo dell'istanza EC2 denominato **`MyAppInstanceProfile`**. Il primo comando scollega tutti i ruoli dal profilo dell'istanza, quindi il secondo comando elimina il profilo dell'istanza.

```
(Get-IAMInstanceProfile -InstanceProfileName MyAppInstanceProfile).Roles |
 Remove-IAMRoleFromInstanceProfile -InstanceProfileName MyAppInstanceProfile
Remove-IAMInstanceProfile -InstanceProfileName MyAppInstanceProfile
```

- Per i dettagli sull'API, vedere DeleteInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina il profilo dell'istanza EC2 denominato **`MyAppInstanceProfile`**. Il primo comando scollega tutti i ruoli dal profilo dell'istanza, quindi il secondo comando elimina il profilo dell'istanza.

```
(Get-IAMInstanceProfile -InstanceProfileName MyAppInstanceProfile).Roles |
 Remove-IAMRoleFromInstanceProfile -InstanceProfileName MyAppInstanceProfile
Remove-IAMInstanceProfile -InstanceProfileName MyAppInstanceProfile
```

- Per i dettagli sull'API, vedere DeleteInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo esempio rimuove il ruolo dal profilo dell'istanza, scollega tutte le policy collegate al ruolo ed elimina tutte le risorse.

```
class AutoScalingWrapper:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix: str,
        inst_type: str,
        ami_param: str,
        autoscaling_client: boto3.client,
        ec2_client: boto3.client,
        ssm_client: boto3.client,
        iam_client: boto3.client,
    ):
        """
        Initializes the AutoScaler class with the necessary parameters.

        :param resource_prefix: The prefix for naming AWS resources that are
 created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
 that is created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
```

```
        self.ami_param = ami_param
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        sts_client = boto3.client("sts")
        self.account_id = sts_client.get_caller_identity()["Account"]

        self.key_pair_name = f"{resource_prefix}-key-pair"
        self.launch_template_name = f"{resource_prefix}-template-"
        self.group_name = f"{resource_prefix}-group"

        # Happy path
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
        self.instance_profile_name = f"{resource_prefix}-prof"

        # Failure mode
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
        self.bad_creds_role_name = f"{resource_prefix}-bc-role"
        self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"


    def delete_instance_profile(self, profile_name: str, role_name: str) -> None:
        """
        Detaches a role from an instance profile, detaches policies from the
role,
        and deletes all the resources.

        :param profile_name: The name of the profile to delete.
        :param role_name: The name of the role to delete.
        """
        try:
            self.iam_client.remove_role_from_instance_profile(
                InstanceProfileName=profile_name, RoleName=role_name
            )

self.iam_client.delete_instance_profile(InstanceProfileName=profile_name)
            log.info("Deleted instance profile %s.", profile_name)
            attached_policies = self.iam_client.list_attached_role_policies(
                RoleName=role_name
            )
            for pol in attached_policies["AttachedPolicies"]:
                self.iam_client.detach_role_policy(
```

```
                    RoleName=role_name, PolicyArn=pol["PolicyArn"]
                )
                if not pol["PolicyArn"].startswith("arn:aws:iam::aws"):
                    self.iam_client.delete_policy(PolicyArn=pol["PolicyArn"])
                log.info("Detached and deleted policy %s.", pol["PolicyName"])
            self.iam_client.delete_role(RoleName=role_name)
            log.info("Deleted role %s.", role_name)
        except ClientError as err:
            log.error(
                f"Couldn't delete instance profile {profile_name} or detach "
                f"policies and delete role {role_name}: {err}"
            )
            if err.response["Error"]["Code"] == "NoSuchEntity":
                log.info(
                    "Instance profile %s doesn't exist, nothing to do.",
profile_name
                )
```

- Per i dettagli sull'API, consulta DeleteInstanceProfile AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DeleteLoginProfile`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteLoginProfile`.

CLI

AWS CLI

Per eliminare una password per un utente IAM

Il comando `delete-login-profile` seguente elimina la password per l'utente IAM denominato Bob.

```
aws iam delete-login-profile \
    --user-name Bob
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Gestione delle password per gli utenti IAM nella Guida per l'utente di AWS .

- Per i dettagli sull'API, consulta DeleteLoginProfile AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il profilo di accesso dall'utente IAM denominato **Bob**. Ciò impedisce all'utente di accedere alla console. AWS Non impedisce all'utente di eseguire chiamate AWS CLI o API utilizzando chiavi di AWS accesso che potrebbero essere ancora collegate all'account utente. PowerShell

```
Remove-IAMLoginProfile -UserName Bob
```

- Per i dettagli sull'API, vedere DeleteLoginProfilein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina il profilo di accesso dall'utente IAM denominato **Bob**. Ciò impedisce all'utente di accedere alla console. AWS Non impedisce all'utente di eseguire chiamate AWS CLI o API utilizzando chiavi di AWS accesso che potrebbero essere ancora collegate all'account utente. PowerShell

```
Remove-IAMLoginProfile -UserName Bob
```

- Per i dettagli sull'API, vedere DeleteLoginProfilein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DeleteOpenIdConnectProvider`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteOpenIdConnectProvider`.

CLI

AWS CLI

Per eliminare un provider di identità OpenID Connect IAM

Questo esempio elimina il provider OIDC IAM che si connette al provider `example.oidcprovider.com`.

```
aws iam delete-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Creazione di provider di identità OpenID Connect (OIDC) nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta DeleteOpenIdConnectProvider AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il provider OIDC IAM che si connette al provider **`example.oidcprovider.com`**. Assicurati di aggiornare o eliminare tutti i ruoli che fanno riferimento a questo provider nell'elemento **`Principal`** della policy di attendibilità del ruolo.

```
Remove-IAMOpenIDConnectProvider -OpenIDConnectProviderArn
 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com
```

- Per i dettagli sull'API, vedere DeleteOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina il provider OIDC IAM che si connette al provider **`example.oidcprovider.com`**. Assicurati di aggiornare o eliminare tutti i ruoli che fanno riferimento a questo provider nell'elemento **`Principal`** della policy di attendibilità del ruolo.

```
Remove-IAMOpenIDConnectProvider -OpenIDConnectProviderArn
 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com
```

- Per i dettagli sull'API, vedere DeleteOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeletePolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeletePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Creazione di utenti di sola lettura e di lettura e scrittura
- Creazione di un'area di lavoro di monitoraggio gestita
- Guida introduttiva allo streaming gestito
- Guida introduttiva all'orchestrazione del flusso di lavoro
- Gestione delle policy
- Configurazione della gestione dei sistemi

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM policy.
```

```
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the policy to
    /// delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeletePolicyAsync(string policyArn)
    {
        var response = await _IAMService.DeletePolicyAsync(new
 DeletePolicyRequest { PolicyArn = policyArn });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
```

```
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an AWS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1
```

```
  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Policy arn:  $policy_arn"
  iecho ""

  response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}
```

- Per i dettagli sull'API, consulta DeletePolicy AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::deletePolicy(const Aws::String &policyArn,
```

```
                                const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::DeletePolicyRequest request;
    request.SetPolicyArn(policyArn);

    auto outcome = iam.DeletePolicy(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error deleting policy with arn " << policyArn << ": "
                  << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully deleted policy with arn " << policyArn
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come eliminare una policy IAM

Questo esempio elimina la policy il cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeletePolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// DeletePolicy deletes a policy.
func (wrapper PolicyWrapper) DeletePolicy(ctx context.Context, policyArn string)
 error {
 _, err := wrapper.IamClient.DeletePolicy(ctx, &iam.DeletePolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't delete policy %v. Here's why: %v\n", policyArn, err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.DeletePolicyRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeletePolicy {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <policyARN>\s

                Where:
                    policyARN - A policy ARN value to delete.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String policyARN = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        deleteIAMPolicy(iam, policyARN);
        iam.close();
    }

    public static void deleteIAMPolicy(IamClient iam, String policyARN) {
        try {
            DeletePolicyRequest request = DeletePolicyRequest.builder()
                    .policyArn(policyARN)
                    .build();

            iam.deletePolicy(request);
            System.out.println("Successfully deleted the policy");

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Eliminare il criterio.

```
import { DeletePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} policyArn
 */
export const deletePolicy = (policyArn) => {
  const command = new DeletePolicyCommand({ PolicyArn: policyArn });
  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun deleteIAMPolicy(policyARNVal: String?) {
```

```
    val request =
        DeletePolicyRequest {
            policyArn = policyARNVal
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.deletePolicy(request)
        println("Successfully deleted $policyARNVal")
    }
}
```

- Per i dettagli sull'API, DeletePolicyconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: questo esempio elimina la policy il cui ARN è **`arn:aws:iam::123456789012:policy/MySamplePolicy`**. Prima di poter eliminare la policy, devi prima eliminare tutte le versioni tranne quella predefinita eseguendo **`Remove-IAMPolicyVersion`**. È inoltre necessario scollegare la policy da qualsiasi utente, gruppo o ruolo IAM.

```
Remove-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Esempio 2: questo esempio elimina una policy eliminando prima tutte le versioni non predefinite della policy, scollegandola da tutte le entità IAM collegate ed eliminando quindi la policy stessa. La prima riga recupera l'oggetto della policy. La seconda riga recupera tutte le versioni della policy che non sono contrassegnate come versione predefinita in una raccolta e quindi elimina ogni policy nella raccolta. La terza riga recupera tutti gli utenti, i gruppi e i ruoli IAM a cui è collegata la policy. Le righe da quattro a sei scollegano la policy da ogni entità collegata. L'ultima riga utilizza questo comando per rimuovere la policy gestita e la versione predefinita rimanente. L'esempio include il parametro di cambio **`-Force`** su qualsiasi riga che lo richieda per sopprimere le richieste di conferma.

```
$pol = Get-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
Get-IAMPolicyVersions -PolicyArn $pol.Arn | where {-not $_.IsDefaultVersion} |
 Remove-IAMPolicyVersion -PolicyArn $pol.Arn -force
$attached = Get-IAMEntitiesForPolicy -PolicyArn $pol.Arn
```

```
$attached.PolicyGroups | Unregister-IAMGroupPolicy -PolicyArn $pol.arn
$attached.PolicyRoles | Unregister-IAMRolePolicy -PolicyArn $pol.arn
$attached.PolicyUsers | Unregister-IAMUserPolicy -PolicyArn $pol.arn
Remove-IAMPolicy $pol.Arn -Force
```

- Per i dettagli sull'API, vedere DeletePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina la policy il cui ARN è **`arn:aws:iam::123456789012:policy/MySamplePolicy`**. Prima di poter eliminare la policy, devi prima eliminare tutte le versioni tranne quella predefinita eseguendo **`Remove-IAMPolicyVersion`**. È inoltre necessario scollegare la policy da qualsiasi utente, gruppo o ruolo IAM.

```
Remove-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Esempio 2: questo esempio elimina una policy eliminando prima tutte le versioni non predefinite della policy, scollegandola da tutte le entità IAM collegate ed eliminando quindi la policy stessa. La prima riga recupera l'oggetto della policy. La seconda riga recupera tutte le versioni della policy che non sono contrassegnate come versione predefinita in una raccolta e quindi elimina ogni policy nella raccolta. La terza riga recupera tutti gli utenti, i gruppi e i ruoli IAM a cui è collegata la policy. Le righe da quattro a sei scollegano la policy da ogni entità collegata. L'ultima riga utilizza questo comando per rimuovere la policy gestita e la versione predefinita rimanente. L'esempio include il parametro di cambio **`-Force`** su qualsiasi riga che lo richieda per sopprimere le richieste di conferma.

```
$pol = Get-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
Get-IAMPolicyVersions -PolicyArn $pol.Arn | where {-not $_.IsDefaultVersion} |
 Remove-IAMPolicyVersion -PolicyArn $pol.Arn -force
$attached = Get-IAMEntitiesForPolicy -PolicyArn $pol.Arn
$attached.PolicyGroups | Unregister-IAMGroupPolicy -PolicyArn $pol.arn
$attached.PolicyRoles | Unregister-IAMRolePolicy -PolicyArn $pol.arn
$attached.PolicyUsers | Unregister-IAMUserPolicy -PolicyArn $pol.arn
Remove-IAMPolicy $pol.Arn -Force
```

- Per i dettagli sull'API, vedere DeletePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def delete_policy(policy_arn):
    """
    Deletes a policy.

    :param policy_arn: The ARN of the policy to delete.
    """
    try:
        iam.Policy(policy_arn).delete()
        logger.info("Deleted policy %s.", policy_arn)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_arn)
        raise
```

- Per i dettagli sull'API, consulta DeletePolicy AWSSDK for Python (Boto3) API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn delete_policy(client: &iamClient, policy: Policy) -> Result<(),
 iamError> {
```

```
    client
        .delete_policy()
        .policy_arn(policy.arn.unwrap())
        .send()
        .await?;
    Ok(())
}
```

- Per i dettagli sulle API, consulta la DeletePolicyguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        lo_iam->deletepolicy( iv_policyarn = iv_policy_arn ).
        MESSAGE 'Policy deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Policy does not exist.' TYPE 'E'.
      CATCH /aws1/cx_iamdeleteconflictex.
        MESSAGE 'Policy cannot be deleted due to attachments.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DeletePolicyconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func deletePolicy(policy: IAMClientTypes.Policy) async throws {
        let input = DeletePolicyInput(
            policyArn: policy.arn
        )
        do {
            _ = try await iamClient.deletePolicy(input: input)
        } catch {
            print("ERROR: deletePolicy:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la DeletePolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeletePolicyVersion`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeletePolicyVersion`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Gestione delle policy
- Rollback di una versione della policy

CLI

AWS CLI

Come eliminare una versione di una policy gestita

Questo esempio elimina la versione identificata come v2 dalla policy il cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeletePolicyVersion AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina la versione identificata come **v2** dalla policy il cui ARN è **arn:aws:iam::123456789012:policy/MySamplePolicy**.

```
Remove-IAMPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/
MySamplePolicy -VersionID v2
```

Esempio 2: questo esempio elimina una policy eliminando prima tutte le versioni non predefinite della policy e quindi eliminando la policy stessa. La prima riga recupera l'oggetto della policy. La seconda riga recupera tutte le versioni della policy che non sono contrassegnate come predefinite in una raccolta e quindi utilizza questo comando per eliminare ogni policy nella raccolta. L'ultima riga rimuove la policy stessa e la versione predefinita rimanente. Tieni presente che per eliminare correttamente una policy gestita,

devi anche scollegare la policy da qualsiasi utente, gruppo o ruolo utilizzando i comandi **Unregister-IAMUserPolicy**, **Unregister-IAMGroupPolicy** e **Unregister-IAMRolePolicy**. Vedere l'esempio per il cmdlet **Remove-IAMPolicy**.

```
$pol = Get-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
Get-IAMPolicyVersions -PolicyArn $pol.Arn | where {-not $_.IsDefaultVersion} |
 Remove-IAMPolicyVersion -PolicyArn $pol.Arn -force
Remove-IAMPolicy -PolicyArn $pol.Arn -force
```

- Per i dettagli sull'API, vedere DeletePolicyVersionin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina la versione identificata come **v2** dalla policy il cui ARN è **arn:aws:iam::123456789012:policy/MySamplePolicy**.

```
Remove-IAMPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/
MySamplePolicy -VersionID v2
```

Esempio 2: questo esempio elimina una policy eliminando prima tutte le versioni non predefinite della policy e quindi eliminando la policy stessa. La prima riga recupera l'oggetto della policy. La seconda riga recupera tutte le versioni della policy che non sono contrassegnate come predefinite in una raccolta e quindi utilizza questo comando per eliminare ogni policy nella raccolta. L'ultima riga rimuove la policy stessa e la versione predefinita rimanente. Tieni presente che per eliminare correttamente una policy gestita, devi anche scollegare la policy da qualsiasi utente, gruppo o ruolo utilizzando i comandi **Unregister-IAMUserPolicy**, **Unregister-IAMGroupPolicy** e **Unregister-IAMRolePolicy**. Vedere l'esempio per il cmdlet **Remove-IAMPolicy**.

```
$pol = Get-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
Get-IAMPolicyVersions -PolicyArn $pol.Arn | where {-not $_.IsDefaultVersion} |
 Remove-IAMPolicyVersion -PolicyArn $pol.Arn -force
Remove-IAMPolicy -PolicyArn $pol.Arn -force
```

- Per i dettagli sull'API, vedere DeletePolicyVersionin AWS Strumenti per PowerShell Cmdlet Reference (V5).

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    lo_iam->deletepolicyversion(
      iv_policyarn = iv_policy_arn
      iv_versionid = iv_version_id ).
    MESSAGE 'Policy version deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy or version does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamdeleteconflictex.
    MESSAGE 'Cannot delete default policy version.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, DeletePolicyVersionconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeleteRole`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteRole`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Configura la connettività dei servizi container

- Crea un'API REST con integrazione con proxy di funzioni
- Creazione di un'area di lavoro di monitoraggio gestita
- Creazione di una dashboard di monitoraggio con il nome della funzione come variabile
- Creazione della tua prima funzione serverless
- Nozioni di base sui data warehouse serverless
- Guida introduttiva alla gestione della configurazione
- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
- Guida introduttiva agli store di funzionalità di apprendimento automatico
- Guida introduttiva ai cluster Kubernetes gestiti
- Guida introduttiva allo streaming gestito
- Guida introduttiva ai cluster di data warehouse con provisioning
- Guida introduttiva all'orchestrazione del flusso di lavoro
- Gestione dei ruoli
- Spostamento di segreti codificati in uno spazio di archiviazione segreto sicuro
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori
- Configurazione della gestione dei sistemi
- Utilizzo delle variabili di proprietà nei dashboard di monitoraggio per monitorare più funzioni serverless

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
/// <summary>
/// Delete an IAM role.
/// </summary>
/// <param name="roleName">The name of the IAM role to delete.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
```

```
    public async Task<bool> DeleteRoleAsync(string roleName)
    {
        var response = await _IAMService.DeleteRoleAsync(new DeleteRoleRequest
 { RoleName = roleName });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteRolesezione AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}
```

```
###############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an AWS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
```

```
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""

  response=$(aws iam delete-role \
    --role-name "$role_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}
```

- Per i dettagli sull'API, consulta DeleteRole AWS CLICommand Reference.

CLI

AWS CLI

Come eliminare un ruolo IAM

Il comando `delete-role` seguente rimuove il ruolo denominato `Test-Role`.

```
aws iam delete-role \
    --role-name Test-Role
```

Questo comando non produce alcun output.

Per poter eliminare un ruolo, devi prima rimuovere il ruolo da qualunque profilo dell'istanza (`remove-role-from-instance-profile`), scollegare eventuali policy gestite (`detach-role-policy`) ed eliminare tutte le policy inline collegate al ruolo (`delete-role-policy`).

Per ulteriori informazioni, consulta Ruoli IAM e Utilizzo dei profili dell'istanza nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta DeleteRole AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// DeleteRole deletes a role. All attached policies must be detached before a
// role can be deleted.
```

```
func (wrapper RoleWrapper) DeleteRole(ctx context.Context, roleName string) error
 {
 _, err := wrapper.IamClient.DeleteRole(ctx, &iam.DeleteRoleInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't delete role %v. Here's why: %v\n", roleName, err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta la DeleteRolesezione AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elimina il ruolo.

```
import { DeleteRoleCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 */
export const deleteRole = (roleName) => {
  const command = new DeleteRoleCommand({ RoleName: roleName });
  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la DeleteRolesezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il ruolo denominato **MyNewRole** dall'account IAM corrente. Prima di poter eliminare il ruolo, devi utilizzare il comando **Unregister-IAMRolePolicy** per scollegare eventuali policy gestite. Le policy in linea vengono eliminate con il ruolo.

```
Remove-IAMRole -RoleName MyNewRole
```

Esempio 2: questo esempio rimuove tutte le policy gestite dal ruolo denominato **MyNewRole** e quindi elimina il ruolo. La prima riga recupera tutte le policy gestite collegate al ruolo come raccolta e quindi le scollega dal ruolo. La seconda riga elimina il ruolo stesso. Le policy in linea vengono eliminate insieme al ruolo.

```
Get-IAMAttachedRolePolicyList -RoleName MyNewRole | Unregister-IAMRolePolicy -
RoleName MyNewRole
Remove-IAMRole -RoleName MyNewRole
```

- Per i dettagli sull'API, vedere DeleteRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina il ruolo denominato **MyNewRole** dall'account IAM corrente. Prima di poter eliminare il ruolo, devi utilizzare il comando **Unregister-IAMRolePolicy** per scollegare eventuali policy gestite. Le policy in linea vengono eliminate con il ruolo.

```
Remove-IAMRole -RoleName MyNewRole
```

Esempio 2: questo esempio rimuove tutte le policy gestite dal ruolo denominato **MyNewRole** e quindi elimina il ruolo. La prima riga recupera tutte le policy gestite collegate al ruolo come raccolta e quindi le scollega dal ruolo. La seconda riga elimina il ruolo stesso. Le policy in linea vengono eliminate insieme al ruolo.

```
Get-IAMAttachedRolePolicyList -RoleName MyNewRole | Unregister-IAMRolePolicy -
RoleName MyNewRole
Remove-IAMRole -RoleName MyNewRole
```

- Per i dettagli sull'API, vedere DeleteRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def delete_role(role_name):
    """
    Deletes a role.

    :param role_name: The name of the role to delete.
    """
    try:
        iam.Role(role_name).delete()
        logger.info("Deleted role %s.", role_name)
    except ClientError:
        logger.exception("Couldn't delete role %s.", role_name)
        raise
```

- Per i dettagli sull'API, consulta DeleteRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
  # Deletes a role and its attached policies.
  #
  # @param role_name [String] The name of the role to delete.
  def delete_role(role_name)
    # Detach and delete attached policies
    @iam_client.list_attached_role_policies(role_name: role_name).each do |
response|
      response.attached_policies.each do |policy|
        @iam_client.detach_role_policy({
                                         role_name: role_name,
                                         policy_arn: policy.policy_arn
                                       })
        # Check if the policy is a customer managed policy (not AWS managed)
        unless policy.policy_arn.include?('aws:policy/')
          @iam_client.delete_policy({ policy_arn: policy.policy_arn })
          @logger.info("Deleted customer managed policy #{policy.policy_name}.")
        end
      end
    end

    # Delete the role
    @iam_client.delete_role({ role_name: role_name })
    @logger.info("Deleted role #{role_name}.")
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't detach policies and delete role #{role_name}. Here's
 why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
```

- Per i dettagli sull'API, consulta la DeleteRolesezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn delete_role(client: &iamClient, role: &Role) -> Result<(), iamError>
 {
    let role = role.clone();
    while client
        .delete_role()
        .role_name(role.role_name())
        .send()
        .await
        .is_err()
    {
        sleep(Duration::from_secs(2)).await;
    }
    Ok(())
}
```

- Per i dettagli sulle API, consulta la DeleteRoleguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        lo_iam->deleterole( iv_rolename = iv_role_name ).
```

```
        MESSAGE 'Role deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Role does not exist.' TYPE 'E'.
      CATCH /aws1/cx_iamdeleteconflictex.
        MESSAGE 'Role cannot be deleted due to attached resources.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DeleteRoleconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func deleteRole(role: IAMClientTypes.Role) async throws {
        let input = DeleteRoleInput(
            roleName: role.roleName
        )
        do {
            _ = try await iamClient.deleteRole(input: input)
        } catch {
            print("ERROR: deleteRole:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DeleteRolePermissionsBoundary`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteRolePermissionsBoundary`.

CLI

AWS CLI

Per eliminare un limite delle autorizzazioni da un ruolo IAM

L'esempio `delete-role-permissions-boundary` seguente elimina il limite delle autorizzazioni per il ruolo IAM specificato. Per applicare un limite delle autorizzazioni a un ruolo, usa il comando `put-role-permissions-boundary`.

```
aws iam delete-role-permissions-boundary \
    --role-name lambda-application-role
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteRolePermissionsBoundary AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio mostra come rimuovere il limite delle autorizzazioni associato a un ruolo IAM.

```
Remove-IAMRolePermissionsBoundary -RoleName MyRoleName
```

- Per i dettagli sull'API, vedere DeleteRolePermissionsBoundaryin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio mostra come rimuovere il limite delle autorizzazioni associato a un ruolo IAM.

```
Remove-IAMRolePermissionsBoundary -RoleName MyRoleName
```

- Per i dettagli sull'API, vedere DeleteRolePermissionsBoundaryin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeleteRolePolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteRolePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Nozioni di base sui data warehouse serverless
- Guida introduttiva alla gestione della configurazione
- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
- Guida introduttiva ai cluster di data warehouse con provisioning
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
```

```
    /// Delete an IAM role policy.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="policyName">The name of the IAM role policy to delete.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRolePolicyAsync(string roleName, string
 policyName)
    {
        var response = await _IAMService.DeleteRolePolicyAsync(new
 DeleteRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteRolePolicysezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come rimuovere una policy da un ruolo IAM

Il comando `delete-role-policy` seguente rimuove la policy denominata `ExamplePolicy` dal ruolo denominato `Test-Role`.

```
aws iam delete-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta DeleteRolePolicy AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { DeleteRolePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 * @param {string} policyName
 */
export const deleteRolePolicy = (roleName, policyName) => {
  const command = new DeleteRolePolicyCommand({
    RoleName: roleName,
    PolicyName: policyName,
  });
  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la DeleteRolePolicysezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina la policy in linea **`S3AccessPolicy`** che è incorporata nel ruolo IAM **`S3BackupRole`**.

```
Remove-IAMRolePolicy -PolicyName S3AccessPolicy -RoleName S3BackupRole
```

- Per i dettagli sull'API, vedere DeleteRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina la policy in linea **`S3AccessPolicy`** che è incorporata nel ruolo IAM **`S3BackupRole`**.

```
Remove-IAMRolePolicy -PolicyName S3AccessPolicy -RoleName S3BackupRole
```

- Per i dettagli sull'API, vedere DeleteRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usa **`DeleteSamlProvider`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteSAMLProvider`.

CLI

AWS CLI

Come eliminare un provider SAML

Questo esempio elimina il provider SAML 2.0 IAM il cui ARN è `arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider`.

```
aws iam delete-saml-provider \
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per informazioni dettagliate sull'API, consulta DeleteSAMLProvider nella documentazione di riferimento dei comandi della AWS CLI .

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { DeleteSAMLProviderCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} providerArn
 * @returns
 */
export const deleteSAMLProvider = async (providerArn) => {
  const command = new DeleteSAMLProviderCommand({
    SAMLProviderArn: providerArn,
  });

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per informazioni dettagliate sull'API, consulta DeleteSAMLProvider nella documentazione di riferimento dell'API AWS SDK per JavaScript .

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il provider SAML 2.0 IAM il cui ARN è **`arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider`**.

```
Remove-IAMSAMLProvider -SAMLProviderArn arn:aws:iam::123456789012:saml-provider/
SAMLADFSProvider
```

- Per informazioni dettagliate sull'API, consulta DeleteSAMLProvider nella documentazione di riferimento dei cmdlet di AWS Strumenti per PowerShell (V4).

Strumenti per PowerShell V5

Esempio 1: questo esempio elimina il provider SAML 2.0 IAM il cui ARN è **`arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider`**.

```
Remove-IAMSAMLProvider -SAMLProviderArn arn:aws:iam::123456789012:saml-provider/
SAMLADFSProvider
```

- Per informazioni dettagliate sull'API, consulta DeleteSAMLProvider nella documentazione di riferimento dei cmdlet di AWS Strumenti per PowerShell (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeleteServerCertificate`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteServerCertificate`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::deleteServerCertificate(const Aws::String &certificateName,
                                          const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::DeleteServerCertificateRequest request;
    request.SetServerCertificateName(certificateName);
```

```
    const auto outcome = iam.DeleteServerCertificate(request);
    bool result = true;
    if (!outcome.IsSuccess()) {
        if (outcome.GetError().GetErrorType() !=
 Aws::IAM::IAMErrors::NO_SUCH_ENTITY) {
            std::cerr << "Error deleting server certificate " << certificateName
 <<
                      ": " << outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Certificate '" << certificateName
                      << "' not found." << std::endl;
        }
    }
    else {
        std::cout << "Successfully deleted server certificate " <<
 certificateName
                  << std::endl;
    }

    return result;
}
```

- Per i dettagli sull'API, consulta la DeleteServerCertificatesezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Per eliminare un certificato server dal tuo AWS account

Il `delete-server-certificate` comando seguente rimuove il certificato del server specificato dal tuo AWS account.

```
aws iam delete-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

Questo comando non produce alcun output.

Per elencare i certificati server disponibili nel tuo AWS account, usa il `list-server-certificates` comando.

Per ulteriori informazioni, consulta Gestione dei certificati server in IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta DeleteServerCertificate AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elimina un certificato del server.

```
import { DeleteServerCertificateCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} certName
 */
export const deleteServerCertificate = (certName) => {
  const command = new DeleteServerCertificateCommand({
    ServerCertificateName: certName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la DeleteServerCertificatesezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.deleteServerCertificate(
  { ServerCertificateName: "CERTIFICATE_NAME" },
  function (err, data) {
    if (err) {
      console.log("Error", err);
    } else {
      console.log("Success", data);
    }
  }
);
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la DeleteServerCertificatesezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il certificato server denominato **`MyServerCert`**.

```
Remove-IAMServerCertificate -ServerCertificateName MyServerCert
```

- Per i dettagli sull'API, vedere DeleteServerCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina il certificato server denominato **`MyServerCert`**.

```
Remove-IAMServerCertificate -ServerCertificateName MyServerCert
```

- Per i dettagli sull'API, vedere DeleteServerCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca, aggiorna ed elimina i certificati server.

```
class ServerCertificateManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'ServerCertificateManager'
  end

  # Creates a new server certificate.
  # @param name [String] the name of the server certificate
  # @param certificate_body [String] the contents of the certificate
  # @param private_key [String] the private key contents
  # @return [Boolean] returns true if the certificate was successfully created
  def create_server_certificate(name, certificate_body, private_key)
    @iam_client.upload_server_certificate({
                                            server_certificate_name: name,
                                            certificate_body: certificate_body,
                                            private_key: private_key
                                          })
```

```
    true
  rescue Aws::IAM::Errors::ServiceError => e
    puts "Failed to create server certificate: #{e.message}"
    false
  end

  # Lists available server certificate names.
  def list_server_certificate_names
    response = @iam_client.list_server_certificates

    if response.server_certificate_metadata_list.empty?
      @logger.info('No server certificates found.')
      return
    end

    response.server_certificate_metadata_list.each do |certificate_metadata|
      @logger.info("Certificate Name:
#{certificate_metadata.server_certificate_name}")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing server certificates: #{e.message}")
  end

  # Updates the name of a server certificate.
  def update_server_certificate_name(current_name, new_name)
    @iam_client.update_server_certificate(
      server_certificate_name: current_name,
      new_server_certificate_name: new_name
    )
    @logger.info("Server certificate name updated from '#{current_name}' to
'#{new_name}'.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error updating server certificate name: #{e.message}")
    false
  end

  # Deletes a server certificate.
  def delete_server_certificate(name)
    @iam_client.delete_server_certificate(server_certificate_name: name)
    @logger.info("Server certificate '#{name}' deleted.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting server certificate: #{e.message}")
```

```
    false
  end
end
```

- Per i dettagli sull'API, consulta la DeleteServerCertificatesezione AWS SDK per Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeleteServiceLinkedRole`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteServiceLinkedRole`.

CLI

AWS CLI

Come eliminare un ruolo collegato a un servizio

L'esempio `delete-service-linked-role` seguente elimina il ruolo collegato al servizio specificato che non è più necessario. L'eliminazione avviene in modo asincrono. Puoi anche controllare lo stato dell'eliminazione e confermare quando è stata completata utilizzando il comando `get-service-linked-role-deletion-status`.

```
aws iam delete-service-linked-role \
    --role-name AWSServiceRoleForLexBots
```

Output:

```
{
    "DeletionTaskId": "task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE"
}
```

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati a servizi nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteServiceLinkedRole AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// DeleteServiceLinkedRole deletes a service-linked role.
func (wrapper RoleWrapper) DeleteServiceLinkedRole(ctx context.Context, roleName
 string) error {
 _, err := wrapper.IamClient.DeleteServiceLinkedRole(ctx,
 &iam.DeleteServiceLinkedRoleInput{
  RoleName: aws.String(roleName)},
 )
 if err != nil {
  log.Printf("Couldn't delete service-linked role %v. Here's why: %v\n",
 roleName, err)
 }
 return err
```

```
}
```

- Per i dettagli sull'API, consulta la DeleteServiceLinkedRolesezione AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { DeleteServiceLinkedRoleCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 */
export const deleteServiceLinkedRole = (roleName) => {
  const command = new DeleteServiceLinkedRoleCommand({ RoleName: roleName });
  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la DeleteServiceLinkedRolesezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio ha eliminato il ruolo collegato al servizio. Tieni presente che se il servizio utilizza ancora questo ruolo, allora questo comando genererà un errore.

```
Remove-IAMServiceLinkedRole -RoleName
 AWSServiceRoleForAutoScaling_RoleNameEndsWithThis
```

- Per i dettagli sull'API, vedere DeleteServiceLinkedRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio ha eliminato il ruolo collegato al servizio. Tieni presente che se il servizio utilizza ancora questo ruolo, allora questo comando genererà un errore.

```
Remove-IAMServiceLinkedRole -RoleName
 AWSServiceRoleForAutoScaling_RoleNameEndsWithThis
```

- Per i dettagli sull'API, vedere DeleteServiceLinkedRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
  # Deletes a service-linked role.
  #
  # @param role_name [String] The name of the role to delete.
  def delete_service_linked_role(role_name)
    response = @iam_client.delete_service_linked_role(role_name: role_name)
    task_id = response.deletion_task_id
    check_deletion_status(role_name, task_id)
  rescue Aws::Errors::ServiceError => e
    handle_deletion_error(e, role_name)
  end

  private
```

```
  # Checks the deletion status of a service-linked role
  #
  # @param role_name [String] The name of the role being deleted
  # @param task_id [String] The task ID for the deletion process
  def check_deletion_status(role_name, task_id)
    loop do
      response = @iam_client.get_service_linked_role_deletion_status(
        deletion_task_id: task_id
      )
      status = response.status
      @logger.info("Deletion of #{role_name} #{status}.")
      break if %w[SUCCEEDED FAILED].include?(status)

      sleep(3)
    end
  end

  # Handles deletion error
  #
  # @param e [Aws::Errors::ServiceError] The error encountered during deletion
  # @param role_name [String] The name of the role attempted to delete
  def handle_deletion_error(e, role_name)
    return if e.code == 'NoSuchEntity'

    @logger.error("Couldn't delete #{role_name}. Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
```

- Per i dettagli sull'API, consulta la DeleteServiceLinkedRolesezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn delete_service_linked_role(
    client: &iamClient,
    role_name: &str,
) -> Result<(), iamError> {
    client
        .delete_service_linked_role()
        .role_name(role_name)
        .send()
        .await?;

    Ok(())
}
```

- Per i dettagli sulle API, consulta la DeleteServiceLinkedRoleguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DeleteSigningCertificate`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteSigningCertificate`.

CLI

AWS CLI

Per eliminare un certificato di firma per un utente IAM

Il comando `delete-signing-certificate` seguente elimina il certificato di firma specificato per l'utente IAM denominato Bob.

```
aws iam delete-signing-certificate \
    --user-name Bob \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE
```

Questo comando non produce alcun output.

Per ottenere l'ID per un certificato di firma, utilizza il comando `list-signing-certificates`.

Per ulteriori informazioni, consulta Gestione dei certificati di firma nella Guida per l'utente di Amazon EC2.

- Per i dettagli sull'API, consulta DeleteSigningCertificate AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il certificato di firma con l'ID **`Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU`** dell'utente IAM denominato **`Bob`**.

```
Remove-IAMSigningCertificate -UserName Bob -CertificateId
 Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU
```

- Per i dettagli sull'API, vedere DeleteSigningCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina il certificato di firma con l'ID **`Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU`** dell'utente IAM denominato **`Bob`**.

```
Remove-IAMSigningCertificate -UserName Bob -CertificateId
 Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU
```

- Per i dettagli sull'API, vedere DeleteSigningCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DeleteUser`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteUser`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Creazione di utenti di sola lettura e di lettura e scrittura

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM user.
    /// </summary>
    /// <param name="userName">The username of the IAM user to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserAsync(string userName)
    {
        var response = await _IAMService.DeleteUserAsync(new DeleteUserRequest
{ UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, DeleteUserconsulta AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an AWS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user."
```

```
  echo ""
}

# Retrieve the calling parameters.
while getopts "u:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    User name:   $user_name"
iecho ""

# If the user does not exist, we don't want to try to delete it.
if (! iam_user_exists "$user_name"); then
  errecho "ERROR: A user with that name does not exist in the account."
  return 1
fi

response=$(aws iam delete-user \
  --user-name "$user_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-user operation failed.$response"
```

```
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- Per i dettagli sull'API, consulta DeleteUser AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::DeleteUserRequest request;
    request.SetUserName(userName);
    auto outcome = iam.DeleteUser(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error deleting IAM user " << userName << ": " <<
                  outcome.GetError().GetMessage() << std::endl;;
    }
    else {
        std::cout << "Successfully deleted IAM user " << userName << std::endl;
    }

    return outcome.IsSuccess();
```

- Per i dettagli sull'API, DeleteUserconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come eliminare un utente IAM

Il comando `delete-user` seguente rimuove l'utente IAM denominato Bob dall'account corrente.

```
aws iam delete-user \
    --user-name Bob
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Eliminazione di un utente IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteUser AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)
```

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// DeleteUser deletes a user.
func (wrapper UserWrapper) DeleteUser(ctx context.Context, userName string) error
 {
 _, err := wrapper.IamClient.DeleteUser(ctx, &iam.DeleteUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete user %v. Here's why: %v\n", userName, err)
 }
 return err
}
```

- Per i dettagli sull'API, DeleteUserconsulta AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.DeleteUserRequest;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteUser {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <userName>\s

                Where:
                    userName - The name of the user to delete.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String userName = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        deleteIAMUser(iam, userName);
        System.out.println("Done");
        iam.close();
    }

    public static void deleteIAMUser(IamClient iam, String userName) {
        try {
            DeleteUserRequest request = DeleteUserRequest.builder()
                    .userName(userName)
                    .build();

            iam.deleteUser(request);
            System.out.println("Successfully deleted IAM user " + userName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
```

```
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, DeleteUserconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Eliminare l'utente.

```
import { DeleteUserCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} name
 */
export const deleteUser = (name) => {
  const command = new DeleteUserCommand({ UserName: name });
  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, DeleteUserconsulta AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  UserName: process.argv[2],
};

iam.getUser(params, function (err, data) {
  if (err && err.code === "NoSuchEntity") {
    console.log("User " + process.argv[2] + " does not exist.");
  } else {
    iam.deleteUser(params, function (err, data) {
      if (err) {
        console.log("Error", err);
      } else {
        console.log("Success", data);
      }
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, DeleteUserconsulta AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun deleteIAMUser(userNameVal: String) {
    val request =
        DeleteUserRequest {
            userName = userNameVal
        }

    // To delete a user, ensure that the user's access keys are deleted first.
    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.deleteUser(request)
        println("Successfully deleted user $userNameVal")
    }
}
```

- Per i dettagli sull'API, DeleteUserconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: questo esempio elimina l'utente IAM denominato **Bob**.

```
Remove-IAMUser -UserName Bob
```

Esempio 2: questo esempio elimina l'utente IAM denominato **Theresa** insieme a tutti gli elementi che devono essere eliminati per primi.

```
$name = "Theresa"

# find any groups and remove user from them
```

```
$groups = Get-IAMGroupForUser -UserName $name
foreach ($group in $groups) { Remove-IAMUserFromGroup -GroupName $group.GroupName
 -UserName $name -Force }

# find any inline policies and delete them
$inlinepols = Get-IAMUserPolicies -UserName $name
foreach ($pol in $inlinepols) { Remove-IAMUserPolicy -PolicyName $pol -UserName
 $name -Force}

# find any managed polices and detach them
$managedpols = Get-IAMAttachedUserPolicies -UserName $name
foreach ($pol in $managedpols) { Unregister-IAMUserPolicy -PolicyArn
 $pol.PolicyArn -UserName $name }

# find any signing certificates and delete them
$certs = Get-IAMSigningCertificate -UserName $name
foreach ($cert in $certs) { Remove-IAMSigningCertificate -CertificateId
 $cert.CertificateId -UserName $name -Force }

# find any access keys and delete them
$keys = Get-IAMAccessKey -UserName $name
foreach ($key in $keys) { Remove-IAMAccessKey -AccessKeyId $key.AccessKeyId -
UserName $name -Force }

# delete the user's login profile, if one exists - note: need to use try/catch to
 suppress not found error
try { $prof = Get-IAMLoginProfile -UserName $name -ea 0 } catch { out-null }
if ($prof) { Remove-IAMLoginProfile -UserName $name -Force }

# find any MFA device, detach it, and if virtual, delete it.
$mfa = Get-IAMMFADevice -UserName $name
if ($mfa) {
    Disable-IAMMFADevice -SerialNumber $mfa.SerialNumber -UserName $name
    if ($mfa.SerialNumber -like "arn:*") { Remove-IAMVirtualMFADevice -
SerialNumber $mfa.SerialNumber }
}

# finally, remove the user
Remove-IAMUser -UserName $name -Force
```

- Per i dettagli sull'API, vedere DeleteUserin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina l'utente IAM denominato **Bob**.

```
Remove-IAMUser -UserName Bob
```

Esempio 2: questo esempio elimina l'utente IAM denominato **Theresa** insieme a tutti gli elementi che devono essere eliminati per primi.

```
$name = "Theresa"

# find any groups and remove user from them
$groups = Get-IAMGroupForUser -UserName $name
foreach ($group in $groups) { Remove-IAMUserFromGroup -GroupName $group.GroupName
 -UserName $name -Force }

# find any inline policies and delete them
$inlinepols = Get-IAMUserPolicies -UserName $name
foreach ($pol in $inlinepols) { Remove-IAMUserPolicy -PolicyName $pol -UserName
 $name -Force}

# find any managed polices and detach them
$managedpols = Get-IAMAttachedUserPolicies -UserName $name
foreach ($pol in $managedpols) { Unregister-IAMUserPolicy -PolicyArn
 $pol.PolicyArn -UserName $name }

# find any signing certificates and delete them
$certs = Get-IAMSigningCertificate -UserName $name
foreach ($cert in $certs) { Remove-IAMSigningCertificate -CertificateId
 $cert.CertificateId -UserName $name -Force }

# find any access keys and delete them
$keys = Get-IAMAccessKey -UserName $name
foreach ($key in $keys) { Remove-IAMAccessKey -AccessKeyId $key.AccessKeyId -
UserName $name -Force }

# delete the user's login profile, if one exists - note: need to use try/catch to
 suppress not found error
try { $prof = Get-IAMLoginProfile -UserName $name -ea 0 } catch { out-null }
if ($prof) { Remove-IAMLoginProfile -UserName $name -Force }

# find any MFA device, detach it, and if virtual, delete it.
$mfa = Get-IAMMFADevice -UserName $name
```

```
if ($mfa) {
    Disable-IAMMFADevice -SerialNumber $mfa.SerialNumber -UserName $name
    if ($mfa.SerialNumber -like "arn:*") { Remove-IAMVirtualMFADevice -
SerialNumber $mfa.SerialNumber }
}

# finally, remove the user
Remove-IAMUser -UserName $name -Force
```

- Per i dettagli sull'API, vedere DeleteUserin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def delete_user(user_name):
    """
    Deletes a user. Before a user can be deleted, all associated resources,
    such as access keys and policies, must be deleted or detached.

    :param user_name: The name of the user.
    """
    try:
        iam.User(user_name).delete()
        logger.info("Deleted user %s.", user_name)
    except ClientError:
        logger.exception("Couldn't delete user %s.", user_name)
        raise
```

- Per i dettagli sull'API, consulta DeleteUser AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
 # Deletes a user and their associated resources
 #
 # @param user_name [String] The name of the user to delete
 def delete_user(user_name)
   user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
   user.each do |key|
     @iam_client.delete_access_key({ access_key_id: key.access_key_id,
user_name: user_name })
     @logger.info("Deleted access key #{key.access_key_id} for user
'#{user_name}'.")
   end

   @iam_client.delete_user(user_name: user_name)
   @logger.info("Deleted user '#{user_name}'.")
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error deleting user '#{user_name}': #{e.message}")
 end
```

- Per i dettagli sull'API, DeleteUserconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn delete_user(client: &iamClient, user: &User) -> Result<(),
 SdkError<DeleteUserError>> {
    let user = user.clone();
    let mut tries: i32 = 0;
    let max_tries: i32 = 10;

    let response: Result<(), SdkError<DeleteUserError>> = loop {
        match client
            .delete_user()
            .user_name(user.user_name())
            .send()
            .await
        {
            Ok(_) => {
                break Ok(());
            }
            Err(e) => {
                tries += 1;
                if tries > max_tries {
                    break Err(e);
                }
                sleep(Duration::from_secs(2)).await;
            }
        }
    };

    response
}
```

- Per i dettagli sulle API, consulta il riferimento DeleteUserall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        lo_iam->deleteuser( iv_username = iv_user_name ).
        MESSAGE 'User deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'User does not exist.' TYPE 'E'.
      CATCH /aws1/cx_iamdeleteconflictex.
        MESSAGE 'User cannot be deleted due to attached resources.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DeleteUserconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func deleteUser(user: IAMClientTypes.User) async throws {
        let input = DeleteUserInput(
            userName: user.userName
        )
        do {
            _ = try await iamClient.deleteUser(input: input)
        } catch {
            print("ERROR: deleteUser:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteUserguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DeleteUserPermissionsBoundary`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteUserPermissionsBoundary`.

CLI

AWS CLI

Per eliminare un limite delle autorizzazioni da un utente IAM

L'esempio `delete-user-permissions-boundary` seguente elimina il limite delle autorizzazioni collegato all'utente IAM denominato `intern`. Per applicare un limite delle autorizzazioni a un utente, usa il comando `put-user-permissions-boundary`.

```
aws iam delete-user-permissions-boundary \
    --user-name intern
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteUserPermissionsBoundary AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio mostra come rimuovere il limite delle autorizzazioni associato a un utente IAM.

```
Remove-IAMUserPermissionsBoundary -UserName joe
```

- Per i dettagli sull'API, vedere DeleteUserPermissionsBoundaryin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio mostra come rimuovere il limite delle autorizzazioni associato a un utente IAM.

```
Remove-IAMUserPermissionsBoundary -UserName joe
```

- Per i dettagli sull'API, vedere DeleteUserPermissionsBoundaryin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DeleteUserPolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteUserPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
/// <summary>
/// Delete an IAM user policy.
/// </summary>
/// <param name="policyName">The name of the IAM policy to delete.</param>
/// <param name="userName">The username of the IAM user.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
```

```
    public async Task<bool> DeleteUserPolicyAsync(string policyName, string
userName)
    {
        var response = await _IAMService.DeleteUserPolicyAsync(new
DeleteUserPolicyRequest { PolicyName = policyName, UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteUserPolicysezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come rimuovere una policy da un utente IAM

Il comando `delete-user-policy` seguente rimuove la policy specificata dall'utente IAM denominato Bob.

```
aws iam delete-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

Questo comando non produce alcun output.

Per ottenere un elenco di policy per un utente IAM, usa il comando `list-user-policies`.

Per ulteriori informazioni, consulta Creare un utente IAM nel tuo AWS account nella Guida per l'utente AWS IAM.

- Per i dettagli sull'API, consulta DeleteUserPolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// DeleteUserPolicy deletes an inline policy from a user.
func (wrapper UserWrapper) DeleteUserPolicy(ctx context.Context, userName string,
 policyName string) error {
 _, err := wrapper.IamClient.DeleteUserPolicy(ctx, &iam.DeleteUserPolicyInput{
  PolicyName: aws.String(policyName),
  UserName:   aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete policy from user %v. Here's why: %v\n", userName,
 err)
 }
```

```
 return err
}
```

- Per i dettagli sull'API, consulta la DeleteUserPolicysezione AWS SDK per Go API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina la policy in linea denominata **`AccessToEC2Policy`** che è incorporata nell'utente IAM denominato **`Bob`**.

```
Remove-IAMUserPolicy -PolicyName AccessToEC2Policy -UserName Bob
```

Esempio 2: questo esempio trova tutte le policy in linea incorporate nell'utente IAM denominato **`Theresa`** e quindi le elimina.

```
$inlinepols = Get-IAMUserPolicies -UserName Theresa
foreach ($pol in $inlinepols) { Remove-IAMUserPolicy -PolicyName $pol -UserName
 Theresa -Force}
```

- Per i dettagli sull'API, vedere DeleteUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina la policy in linea denominata **`AccessToEC2Policy`** che è incorporata nell'utente IAM denominato **`Bob`**.

```
Remove-IAMUserPolicy -PolicyName AccessToEC2Policy -UserName Bob
```

Esempio 2: questo esempio trova tutte le policy in linea incorporate nell'utente IAM denominato **`Theresa`** e quindi le elimina.

```
$inlinepols = Get-IAMUserPolicies -UserName Theresa
foreach ($pol in $inlinepols) { Remove-IAMUserPolicy -PolicyName $pol -UserName
 Theresa -Force}
```

- Per i dettagli sull'API, vedere DeleteUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
 # Deletes a user and their associated resources
 #
 # @param user_name [String] The name of the user to delete
 def delete_user(user_name)
   user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
   user.each do |key|
     @iam_client.delete_access_key({ access_key_id: key.access_key_id,
user_name: user_name })
     @logger.info("Deleted access key #{key.access_key_id} for user
'#{user_name}'.")
   end

   @iam_client.delete_user(user_name: user_name)
   @logger.info("Deleted user '#{user_name}'.")
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error deleting user '#{user_name}': #{e.message}")
 end
```

- Per i dettagli sull'API, consulta la DeleteUserPolicysezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn delete_user_policy(
    client: &iamClient,
    user: &User,
    policy_name: &str,
) -> Result<(), SdkError<DeleteUserPolicyError>> {
    client
        .delete_user_policy()
        .user_name(user.user_name())
        .policy_name(policy_name)
        .send()
        .await?;

    Ok(())
}
```

- Per i dettagli sulle API, consulta la DeleteUserPolicyguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
```

```
import AWSS3


    func deleteUserPolicy(user: IAMClientTypes.User, policyName: String) async
 throws {
        let input = DeleteUserPolicyInput(
            policyName: policyName,
            userName: user.userName
        )
        do {
            _ = try await iamClient.deleteUserPolicy(input: input)
        } catch {
            print("ERROR: deleteUserPolicy:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteUserPolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DeleteVirtualMfaDevice`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteVirtualMfaDevice`.

CLI

AWS CLI

Per rimuovere un dispositivo MFA virtuale

Il comando `delete-virtual-mfa-device` seguente rimuove il dispositivo MFA specificato dall'account corrente.

```
aws iam delete-virtual-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/MFATest
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Disattivazione dei dispositivi MFA nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta DeleteVirtualMfaDevice AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il dispositivo MFA virtuale IAM il cui ARN è. **`arn:aws:iam::123456789012:mfa/bob`**.

```
Remove-IAMVirtualMFADevice -SerialNumber arn:aws:iam::123456789012:mfa/bob
```

Esempio 2: questo esempio verifica se all'utente IAM Theresa è assegnato un dispositivo MFA. Se ne viene trovato uno, il dispositivo viene disabilitato per l'utente IAM. Se il dispositivo è virtuale, anch'esso viene eliminato.

```
$mfa = Get-IAMMFADevice -UserName Theresa
if ($mfa) {
    Disable-IAMMFADevice -SerialNumber $mfa.SerialNumber -UserName $name
    if ($mfa.SerialNumber -like "arn:*") { Remove-IAMVirtualMFADevice -
SerialNumber $mfa.SerialNumber }
}
```

- Per i dettagli sull'API, vedere DeleteVirtualMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina il dispositivo MFA virtuale IAM il cui ARN è. **`arn:aws:iam::123456789012:mfa/bob`**.

```
Remove-IAMVirtualMFADevice -SerialNumber arn:aws:iam::123456789012:mfa/bob
```

Esempio 2: questo esempio verifica se all'utente IAM Theresa è assegnato un dispositivo MFA. Se ne viene trovato uno, il dispositivo viene disabilitato per l'utente IAM. Se il dispositivo è virtuale, anch'esso viene eliminato.

```
$mfa = Get-IAMMFADevice -UserName Theresa
if ($mfa) {
    Disable-IAMMFADevice -SerialNumber $mfa.SerialNumber -UserName $name
    if ($mfa.SerialNumber -like "arn:*") { Remove-IAMVirtualMFADevice -
SerialNumber $mfa.SerialNumber }
}
```

- Per i dettagli sull'API, vedere DeleteVirtualMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DetachGroupPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DetachGroupPolicy`.

CLI

AWS CLI

Per scollegare una policy da un gruppo

Questo esempio rimuove la policy gestita con l'ARN `arn:aws:iam::123456789012:policy/TesterAccessPolicy` dal gruppo denominato `Testers`.

```
aws iam detach-group-policy \
    --group-name Testers \
    --policy-arn arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Gestione di gruppi di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta DetachGroupPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio scollega la policy del gruppo gestita il cui ARN è **`arn:aws:iam::123456789012:policy/TesterAccessPolicy`** dal gruppo denominato **`Testers`**.

```
Unregister-IAMGroupPolicy -GroupName Testers -PolicyArn
 arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

Esempio 2: questo esempio trova tutte le policy gestite collegate al gruppo denominato **`Testers`** e le scollega dal gruppo.

```
Get-IAMAttachedGroupPolicies -GroupName Testers | Unregister-IAMGroupPolicy -
Groupname Testers
```

- Per i dettagli sull'API, vedere DetachGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio scollega la policy del gruppo gestita il cui ARN è **`arn:aws:iam::123456789012:policy/TesterAccessPolicy`** dal gruppo denominato **`Testers`**.

```
Unregister-IAMGroupPolicy -GroupName Testers -PolicyArn
 arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

Esempio 2: questo esempio trova tutte le policy gestite collegate al gruppo denominato **`Testers`** e le scollega dal gruppo.

```
Get-IAMAttachedGroupPolicies -GroupName Testers | Unregister-IAMGroupPolicy -
Groupname Testers
```

- Per i dettagli sull'API, vedere DetachGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DetachRolePolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DetachRolePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Configura la connettività dei servizi container
- Crea un'API REST con integrazione con proxy di funzioni
- Creazione di un'area di lavoro di monitoraggio gestita
- Creazione di una dashboard di monitoraggio con il nome della funzione come variabile
- Creazione della tua prima funzione serverless
- Guida introduttiva alla gestione della configurazione
- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
- Guida introduttiva agli store di funzionalità di apprendimento automatico
- Guida introduttiva ai cluster Kubernetes gestiti
- Guida introduttiva allo streaming gestito
- Guida introduttiva all'orchestrazione del flusso di lavoro
- Gestione dei ruoli
- Spostamento di segreti codificati in uno spazio di archiviazione segreto sicuro
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori
- Configurazione della gestione dei sistemi
- Utilizzo delle variabili di proprietà nei dashboard di monitoraggio per monitorare più funzioni serverless

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Detach an IAM policy from an IAM role.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the IAM
policy.</param>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DetachRolePolicyAsync(string policyArn, string
roleName)
    {
        var response = await _IAMService.DetachRolePolicyAsync(new
DetachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, DetachRolePolicyconsulta AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
```

```
  echo "  -p policy_ARN -- The IAM policy document ARN."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy ARN with the -p parameter."
  usage
  return 1
fi

response=$(aws iam detach-role-policy \
  --role-name "$role_name" \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
  return 1
```

```
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta DetachRolePolicy AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::DetachRolePolicyRequest detachRequest;
    detachRequest.SetRoleName(roleName);
    detachRequest.SetPolicyArn(policyArn);

    auto detachOutcome = iam.DetachRolePolicy(detachRequest);
    if (!detachOutcome.IsSuccess()) {
        std::cerr << "Failed to detach policy " << policyArn << " from role "
                  << roleName << ": " << detachOutcome.GetError().GetMessage() <<
                  std::endl;
    }
    else {
        std::cout << "Successfully detached policy " << policyArn << " from role
 "
                  << roleName << std::endl;
    }

    return detachOutcome.IsSuccess();
```

- Per i dettagli sull'API, DetachRolePolicyconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come scollegare una policy da un ruolo

Questo esempio rimuove la policy gestita con l'ARN `arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy` dal ruolo denominato `FedTesterRole`.

```
aws iam detach-role-policy \
    --role-name FedTesterRole \
    --policy-arn arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta DetachRolePolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
```

```
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// DetachRolePolicy detaches a policy from a role.
func (wrapper RoleWrapper) DetachRolePolicy(ctx context.Context, roleName string,
 policyArn string) error {
 _, err := wrapper.IamClient.DetachRolePolicy(ctx, &iam.DetachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't detach policy from role %v. Here's why: %v\n", roleName,
 err)
 }
 return err
}
```

- Per i dettagli sull'API, DetachRolePolicyconsulta AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.DetachRolePolicyRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
```

```
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DetachRolePolicy {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <roleName> <policyArn>\s

                Where:
                    roleName - A role name that you can obtain from the AWS
 Management Console.\s
                    policyArn - A policy ARN that you can obtain from the AWS
 Management Console.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String roleName = args[0];
        String policyArn = args[1];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();
        detachPolicy(iam, roleName, policyArn);
        System.out.println("Done");
        iam.close();
    }

    public static void detachPolicy(IamClient iam, String roleName, String
 policyArn) {
        try {
```

```
            DetachRolePolicyRequest request = DetachRolePolicyRequest.builder()
                    .roleName(roleName)
                    .policyArn(policyArn)
                    .build();

            iam.detachRolePolicy(request);
            System.out.println("Successfully detached policy " + policyArn +
                    " from role " + roleName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, DetachRolePolicyconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Scollega la policy.

```
import { DetachRolePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} policyArn
 * @param {string} roleName
 */
export const detachRolePolicy = (policyArn, roleName) => {
  const command = new DetachRolePolicyCommand({
```

```
    PolicyArn: policyArn,
    RoleName: roleName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, DetachRolePolicyconsulta AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var paramsRoleList = {
  RoleName: process.argv[2],
};

iam.listAttachedRolePolicies(paramsRoleList, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    var myRolePolicies = data.AttachedPolicies;
    myRolePolicies.forEach(function (val, index, array) {
      if (myRolePolicies[index].PolicyName === "AmazonDynamoDBFullAccess") {
        var params = {
          PolicyArn: "arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess",
          RoleName: process.argv[2],
        };
```

```
        iam.detachRolePolicy(params, function (err, data) {
          if (err) {
            console.log("Unable to detach policy from role", err);
          } else {
            console.log("Policy detached from role successfully");
            process.exit();
          }
        });
      }
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, DetachRolePolicyconsulta AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun detachPolicy(
    roleNameVal: String,
    policyArnVal: String,
) {
    val request =
        DetachRolePolicyRequest {
            roleName = roleNameVal
            policyArn = policyArnVal
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.detachRolePolicy(request)
        println("Successfully detached policy $policyArnVal from role
 $roleNameVal")
```

```
    }
}
```

- Per i dettagli sull'API, DetachRolePolicyconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: questo esempio scollega la policy del gruppo gestita il cui ARN è **`arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy`** dal ruolo denominato **`FedTesterRole`**.

```
Unregister-IAMRolePolicy -RoleName FedTesterRole -PolicyArn
 arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

Esempio 2: questo esempio trova tutte le policy gestite collegate al ruolo denominato **`FedTesterRole`** e le scollega dal ruolo.

```
Get-IAMAttachedRolePolicyList -RoleName FedTesterRole | Unregister-IAMRolePolicy
 -Rolename FedTesterRole
```

- Per i dettagli sull'API, vedere DetachRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio scollega la policy del gruppo gestita il cui ARN è **`arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy`** dal ruolo denominato **`FedTesterRole`**.

```
Unregister-IAMRolePolicy -RoleName FedTesterRole -PolicyArn
 arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

Esempio 2: questo esempio trova tutte le policy gestite collegate al ruolo denominato **`FedTesterRole`** e le scollega dal ruolo.

```
Get-IAMAttachedRolePolicyList -RoleName FedTesterRole | Unregister-IAMRolePolicy
 -Rolename FedTesterRole
```

- Per i dettagli sull'API, vedere DetachRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Scollega una policy da un ruolo utilizzando l'oggetto Policy Boto3.

```
def detach_from_role(role_name, policy_arn):
    """
    Detaches a policy from a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Policy(policy_arn).detach_role(RoleName=role_name)
        logger.info("Detached policy %s from role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from role %s.", policy_arn, role_name
        )
        raise
```

Scollega una policy da un ruolo utilizzando l'oggetto Role Boto3.

```
def detach_policy(role_name, policy_arn):
    """
    Detaches a policy from a role.
```

```
    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Role(role_name).detach_policy(PolicyArn=policy_arn)
        logger.info("Detached policy %s from role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from role %s.", policy_arn, role_name
        )
        raise
```

- Per i dettagli sull'API, consulta DetachRolePolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, collega e scollega le policy relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'PolicyManager'
  end
```

```
 # Creates a policy
 #
 # @param policy_name [String] The name of the policy
 # @param policy_document [Hash] The policy document
 # @return [String] The policy ARN if successful, otherwise nil
 def create_policy(policy_name, policy_document)
   response = @iam_client.create_policy(
     policy_name: policy_name,
     policy_document: policy_document.to_json
   )
   response.policy.arn
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error creating policy: #{e.message}")
   nil
 end

 # Fetches an IAM policy by its ARN
 # @param policy_arn [String] the ARN of the IAM policy to retrieve
 # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
 def get_policy(policy_arn)
   response = @iam_client.get_policy(policy_arn: policy_arn)
   policy = response.policy
   @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
   policy
 rescue Aws::IAM::Errors::NoSuchEntity
   @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
   raise
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
   raise
 end

 # Attaches a policy to a role
 #
 # @param role_name [String] The name of the role
 # @param policy_arn [String] The policy ARN
 # @return [Boolean] true if successful, false otherwise
 def attach_policy_to_role(role_name, policy_arn)
   @iam_client.attach_role_policy(
     role_name: role_name,
     policy_arn: policy_arn
```

```
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, DetachRolePolicyconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn detach_role_policy(
    client: &iamClient,
    role_name: &str,
    policy_arn: &str,
) -> Result<(), iamError> {
    client
        .detach_role_policy()
        .role_name(role_name)
        .policy_arn(policy_arn)
        .send()
        .await?;

    Ok(())
}
```

- Per i dettagli sulle API, consulta il riferimento DetachRolePolicyall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        lo_iam->detachrolepolicy(
```

```
          iv_rolename = iv_role_name
          iv_policyarn = iv_policy_arn ).
        MESSAGE 'Policy detached from role successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Role or policy does not exist.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DetachRolePolicyconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func detachRolePolicy(policy: IAMClientTypes.Policy, role:
 IAMClientTypes.Role) async throws {
        let input = DetachRolePolicyInput(
            policyArn: policy.arn,
            roleName: role.roleName
        )

        do {
            _ = try await iamClient.detachRolePolicy(input: input)
        } catch {
            print("ERROR: detachRolePolicy:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la DetachRolePolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`DetachUserPolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DetachUserPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di utenti di sola lettura e di lettura e scrittura

CLI

AWS CLI

Come scollegare una policy da un utente

Questo esempio rimuove la policy gestita con l'ARN `arn:aws:iam::123456789012:policy/TesterPolicy` dall'utente Bob.

```
aws iam detach-user-policy \
    --user-name Bob \
    --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica delle autorizzazioni per un utente IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DetachUserPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio scollega la policy gestita il cui ARN è **`arn:aws:iam::123456789012:policy/TesterPolicy`** dall'utente IAM denominato **`Bob`**.

```
Unregister-IAMUserPolicy -UserName Bob -PolicyArn
 arn:aws:iam::123456789012:policy/TesterPolicy
```

Esempio 2: questo esempio trova tutte le policy gestite collegate all'utente IAM denominato **`Theresa`** e le scollega dall'utente.

```
Get-IAMAttachedUserPolicyList -UserName Theresa | Unregister-IAMUserPolicy -
Username Theresa
```

- Per i dettagli sull'API, vedere DetachUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio scollega la policy gestita il cui ARN è **`arn:aws:iam::123456789012:policy/TesterPolicy`** dall'utente IAM denominato **`Bob`**.

```
Unregister-IAMUserPolicy -UserName Bob -PolicyArn
 arn:aws:iam::123456789012:policy/TesterPolicy
```

Esempio 2: questo esempio trova tutte le policy gestite collegate all'utente IAM denominato **`Theresa`** e le scollega dall'utente.

```
Get-IAMAttachedUserPolicyList -UserName Theresa | Unregister-IAMUserPolicy -
Username Theresa
```

- Per i dettagli sull'API, vedere DetachUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def detach_policy(user_name, policy_arn):
    """
    Detaches a policy from a user.

    :param user_name: The name of the user.
    :param policy_arn: The Amazon Resource Name (ARN) of the policy.
    """
    try:
        iam.User(user_name).detach_policy(PolicyArn=policy_arn)
        logger.info("Detached policy %s from user %s.", policy_arn, user_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from user %s.", policy_arn, user_name
        )
        raise
```

- Per i dettagli sull'API, consulta DetachUserPolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
  # Detaches a policy from a user
  #
  # @param user_name [String] The name of the user
  # @param policy_arn [String] The ARN of the policy to detach
  # @return [Boolean] true if the policy was successfully detached, false
otherwise
  def detach_user_policy(user_name, policy_arn)
    @iam_client.detach_user_policy(
      user_name: user_name,
      policy_arn: policy_arn
    )
    @logger.info("Policy '#{policy_arn}' detached from user '#{user_name}'
successfully.")
    true
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error('Error detaching policy: Policy or user does not exist.')
    false
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from user '#{user_name}':
#{e.message}")
    false
  end
```

- Per i dettagli sull'API, consulta la DetachUserPolicysezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn detach_user_policy(
    client: &iamClient,
    user_name: &str,
    policy_arn: &str,
```

```
) -> Result<(), iamError> {
    client
        .detach_user_policy()
        .user_name(user_name)
        .policy_arn(policy_arn)
        .send()
        .await?;

    Ok(())
}
```

- Per i dettagli sulle API, consulta la DetachUserPolicyguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        lo_iam->detachuserpolicy(
          iv_username = iv_user_name
          iv_policyarn = iv_policy_arn ).
        MESSAGE 'Policy detached from user successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'User or policy does not exist.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DetachUserPolicyconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`EnableMfaDevice`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`EnableMfaDevice`.

CLI

AWS CLI

Per abilitare un dispositivo MFA

Dopo aver utilizzato il comando `create-virtual-mfa-device` per creare un nuovo dispositivo MFA virtuale, è possibile assegnare il dispositivo MFA a un utente. L'esempio `enable-mfa-device` seguente assegna il dispositivo MFA con il numero di serie `arn:aws:iam::210987654321:mfa/BobsMFADevice` all'utente Bob. Il comando sincronizza inoltre il dispositivo con AWS l'inclusione dei primi due codici in sequenza dal dispositivo MFA virtuale.

```
aws iam enable-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
    --authentication-code2 789012
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta AWS CLI Command EnableMfaDeviceReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo comando abilita il dispositivo hardware MFA con il numero di serie **987654321098** e associa il dispositivo all'utente **Bob**. Include i primi due codici in sequenza provenienti dal dispositivo.

```
Enable-IAMMFADevice -UserName "Bob" -SerialNumber "987654321098" -
AuthenticationCode1 "12345678" -AuthenticationCode2 "87654321"
```

Esempio 2: questo esempio crea e abilita un dispositivo MFA virtuale. Il primo comando crea il dispositivo virtuale e restituisce la rappresentazione dell'oggetto del dispositivo nella variabile **$MFADevice**. È possibile utilizzare le proprietà **.Base32StringSeed** o **QRCodePng** per configurare l'applicazione software dell'utente. Il comando finale assegna il dispositivo all'utente **David**, identificandolo in base al numero di serie. Il comando sincronizza inoltre il dispositivo con AWS l'inclusione dei primi due codici in sequenza dal dispositivo MFA virtuale.

```
$MFADevice = New-IAMVirtualMFADevice -VirtualMFADeviceName "MyMFADevice"
# see example for New-IAMVirtualMFADevice to see how to configure the software
 program with PNG or base32 seed code
Enable-IAMMFADevice -UserName "David" -SerialNumber -SerialNumber
 $MFADevice.SerialNumber -AuthenticationCode1 "24681357" -AuthenticationCode2
 "13572468"
```

- Per i dettagli sull'API, vedere EnableMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando abilita il dispositivo hardware MFA con il numero di serie **987654321098** e associa il dispositivo all'utente **Bob**. Include i primi due codici in sequenza provenienti dal dispositivo.

```
Enable-IAMMFADevice -UserName "Bob" -SerialNumber "987654321098" -
AuthenticationCode1 "12345678" -AuthenticationCode2 "87654321"
```

Esempio 2: questo esempio crea e abilita un dispositivo MFA virtuale. Il primo comando crea il dispositivo virtuale e restituisce la rappresentazione dell'oggetto del dispositivo nella variabile **$MFADevice**. È possibile utilizzare le proprietà **.Base32StringSeed** o **QRCodePng** per configurare l'applicazione software dell'utente. Il comando finale assegna il dispositivo all'utente **David**, identificandolo in base al numero di serie. Il comando sincronizza inoltre il dispositivo con AWS l'inclusione dei primi due codici in sequenza dal dispositivo MFA virtuale.

```
$MFADevice = New-IAMVirtualMFADevice -VirtualMFADeviceName "MyMFADevice"
# see example for New-IAMVirtualMFADevice to see how to configure the software
 program with PNG or base32 seed code
Enable-IAMMFADevice -UserName "David" -SerialNumber -SerialNumber
 $MFADevice.SerialNumber -AuthenticationCode1 "24681357" -AuthenticationCode2
 "13572468"
```

- Per i dettagli sull'API, vedere EnableMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`GenerateCredentialReport`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GenerateCredentialReport`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

CLI

AWS CLI

Come generare un report delle credenziali

L'esempio seguente tenta di generare un rapporto sulle credenziali per l' AWS account.

```
aws iam generate-credential-report
```

Output:

```
{
    "State":  "STARTED",
    "Description": "No report exists. Starting a new report generation task"
}
```

Per ulteriori informazioni, consulta Ottenere i report sulle credenziali per il tuo AWS account nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta GenerateCredentialReport AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio richiede la generazione di un nuovo report, operazione che può essere eseguita ogni quattro ore. Se l'ultimo report è ancora recente, il campo Stato riporta **`COMPLETE`**. Utilizzare **`Get-IAMCredentialReport`** per visualizzare il report completato.

```
Request-IAMCredentialReport
```

Output:

```
Description                                                    State
-----------                                                    -----
No report exists. Starting a new report generation task        STARTED
```

- Per i dettagli sull'API, vedere GenerateCredentialReportin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio richiede la generazione di un nuovo report, operazione che può essere eseguita ogni quattro ore. Se l'ultimo report è ancora recente, il campo Stato riporta **`COMPLETE`**. Utilizzare **`Get-IAMCredentialReport`** per visualizzare il report completato.

```
Request-IAMCredentialReport
```

Output:

```
Description                                                    State
-----------                                                    -----
No report exists. Starting a new report generation task        STARTED
```

- Per i dettagli sull'API, vedere GenerateCredentialReportin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def generate_credential_report():
    """
    Starts generation of a credentials report about the current account. After
    calling this function to generate the report, call get_credential_report
    to get the latest report. A new report can be generated a minimum of four
 hours
    after the last one was generated.
    """
    try:
        response = iam.meta.client.generate_credential_report()
        logger.info(
            "Generating credentials report for your account. " "Current state is
 %s.",
            response["State"],
        )
    except ClientError:
        logger.exception("Couldn't generate a credentials report for your
 account.")
        raise
    else:
        return response
```

- Per i dettagli sull'API, consulta GenerateCredentialReport AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    oo_result = lo_iam->generatecredentialreport( ).
    MESSAGE 'Credential report generation started.' TYPE 'I'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Report generation limit exceeded.' TYPE 'E'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when generating credential report.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, GenerateCredentialReportconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GenerateServiceLastAccessedDetails`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GenerateServiceLastAccessedDetails`.

CLI

AWS CLI

Esempio 1: generare un report di accesso al servizio per una policy personalizzata

L'esempio `generate-service-last-accessed-details` seguente avvia un processo in background per generare un report che elenca i servizi a cui accedono gli utenti IAM e altre entità con una policy personalizzata denominata `intern-boundary`. È possibile visualizzare

il report dopo averlo creato eseguendo il comando `get-service-last-accessed-details`.

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::123456789012:policy/intern-boundary
```

Output:

```
{
    "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"
}
```

Esempio 2: generare un rapporto di accesso al servizio per la politica AWS gestita AdministratorAccess

L'`generate-service-last-accessed-details`esempio seguente avvia un processo in background per generare un report che elenca i servizi a cui accedono gli utenti IAM e altre entità con la `AdministratorAccess` policy AWS gestita. È possibile visualizzare il report dopo averlo creato eseguendo il comando `get-service-last-accessed-details`.

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::aws:policy/AdministratorAccess
```

Output:

```
{
    "JobId": "78b6c2ba-d09e-6xmp-7039-ecde30b26916"
}
```

Per ulteriori informazioni, consulta Refining permissions in AWS using last access information nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta AWS CLI Command GenerateServiceLastAccessedDetailsReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio è un cmdlet equivalente all'API. GenerateServiceLastAccessedDetails Ciò fornisce un ID di lavoro che può essere utilizzato in e Get-IAMServiceLastAccessedDetail Get-IAMServiceLastAccessedDetailWithEntity

```
Request-IAMServiceLastAccessedDetail -Arn arn:aws:iam::123456789012:user/TestUser
```

- Per i dettagli sull'API, vedere GenerateServiceLastAccessedDetailsin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio è un cmdlet equivalente all'API. GenerateServiceLastAccessedDetails Ciò fornisce un ID di lavoro che può essere utilizzato in e Get-IAMServiceLastAccessedDetail Get-IAMServiceLastAccessedDetailWithEntity

```
Request-IAMServiceLastAccessedDetail -Arn arn:aws:iam::123456789012:user/TestUser
```

- Per i dettagli sull'API, vedere GenerateServiceLastAccessedDetailsin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`GetAccessKeyLastUsed`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetAccessKeyLastUsed`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestione delle chiavi di accesso

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::accessKeyLastUsed(const Aws::String &secretKeyID,
                                    const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::GetAccessKeyLastUsedRequest request;

    request.SetAccessKeyId(secretKeyID);

    Aws::IAM::Model::GetAccessKeyLastUsedOutcome outcome =
 iam.GetAccessKeyLastUsed(
            request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error querying last used time for access key " <<
                  secretKeyID << ":" << outcome.GetError().GetMessage() <<
 std::endl;
    }
    else {
        Aws::String lastUsedTimeString =
                outcome.GetResult()
                        .GetAccessKeyLastUsed()
                        .GetLastUsedDate()
                        .ToGmtString(Aws::Utils::DateFormat::ISO_8601);
        std::cout << "Access key " << secretKeyID << " last used at time " <<
                  lastUsedTimeString << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la GetAccessKeyLastUsedsezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come recuperare informazioni sull'ultimo utilizzo della chiave di accesso specificata

L'esempio seguente recupera informazioni sull'ultimo utilizzo della chiave di accesso ABCDEXAMPLE.

```
aws iam get-access-key-last-used \
    --access-key-id ABCDEXAMPLE
```

Output:

```
{
    "UserName":  "Bob",
    "AccessKeyLastUsed": {
        "Region": "us-east-1",
        "ServiceName": "iam",
        "LastUsedDate": "2015-06-16T22:45:00Z"
    }
}
```

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetAccessKeyLastUsed AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Recupera la chiave di accesso.

```
import { GetAccessKeyLastUsedCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} accessKeyId
 */
export const getAccessKeyLastUsed = async (accessKeyId) => {
  const command = new GetAccessKeyLastUsedCommand({
    AccessKeyId: accessKeyId,
  });

  const response = await client.send(command);

  if (response.AccessKeyLastUsed?.LastUsedDate) {
    console.log(`
    ${accessKeyId} was last used by ${response.UserName} via
    the ${response.AccessKeyLastUsed.ServiceName} service on
    ${response.AccessKeyLastUsed.LastUsedDate.toISOString()}
    `);
  }

  return response;
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la GetAccessKeyLastUsedsezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
```

```
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.getAccessKeyLastUsed(
  { AccessKeyId: "ACCESS_KEY_ID" },
  function (err, data) {
    if (err) {
      console.log("Error", err);
    } else {
      console.log("Success", data.AccessKeyLastUsed);
    }
  }
);
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la GetAccessKeyLastUsedsezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: restituisce il nome utente proprietario e le informazioni sull'ultimo utilizzo della chiave di accesso fornita.

```
Get-IAMAccessKeyLastUsed -AccessKeyId ABCDEXAMPLE
```

- Per i dettagli sull'API, vedere GetAccessKeyLastUsedin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: restituisce il nome utente proprietario e le informazioni sull'ultimo utilizzo della chiave di accesso fornita.

```
Get-IAMAccessKeyLastUsed -AccessKeyId ABCDEXAMPLE
```

- Per i dettagli sull'API, vedere GetAccessKeyLastUsedin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def get_last_use(key_id):
    """
    Gets information about when and how a key was last used.

    :param key_id: The ID of the key to look up.
    :return: Information about the key's last use.
    """
    try:
        response = iam.meta.client.get_access_key_last_used(AccessKeyId=key_id)
        last_used_date = response["AccessKeyLastUsed"].get("LastUsedDate", None)
        last_service = response["AccessKeyLastUsed"].get("ServiceName", None)
        logger.info(
            "Key %s was last used by %s on %s to access %s.",
            key_id,
            response["UserName"],
            last_used_date,
            last_service,
        )
    except ClientError:
        logger.exception("Couldn't get last use of key %s.", key_id)
        raise
    else:
        return response
```

- Per i dettagli sull'API, consulta GetAccessKeyLastUsed AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    oo_result = lo_iam->getaccesskeylastused(
      iv_accesskeyid = iv_access_key_id ).
    MESSAGE 'Retrieved access key last used information.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Access key does not exist.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, GetAccessKeyLastUsedconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`GetAccountAuthorizationDetails`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetAccountAuthorizationDetails`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

CLI

AWS CLI

Per elencare gli utenti, i gruppi, i ruoli e le politiche IAM di un AWS account

Il `get-account-authorization-details` comando seguente restituisce informazioni su tutti gli utenti, i gruppi, i ruoli e le politiche IAM presenti nell' AWS account.

```
aws iam get-account-authorization-details
```

Output:

```
{
    "RoleDetailList": [
        {
            "AssumeRolePolicyDocument": {
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "RoleId": "AROA1234567890EXAMPLE",
            "CreateDate": "2014-07-30T17:09:20Z",
            "InstanceProfileList": [
                {
                    "InstanceProfileId": "AIPA1234567890EXAMPLE",
                    "Roles": [
                        {
                            "AssumeRolePolicyDocument": {
                                "Version":"2012-10-17",
                                "Statement": [
                                    {
                                        "Sid": "",
                                        "Effect": "Allow",
                                        "Principal": {
```

```
                                        "Service": "ec2.amazonaws.com"
                                    },
                                    "Action": "sts:AssumeRole"
                                }
                            ]
                        },
                        "RoleId": "AROA1234567890EXAMPLE",
                        "CreateDate": "2014-07-30T17:09:20Z",
                        "RoleName": "EC2role",
                        "Path": "/",
                        "Arn": "arn:aws:iam::123456789012:role/EC2role"
                    }
                ],
                "CreateDate": "2014-07-30T17:09:20Z",
                "InstanceProfileName": "EC2role",
                "Path": "/",
                "Arn": "arn:aws:iam::123456789012:instance-profile/EC2role"
            }
        ],
        "RoleName": "EC2role",
        "Path": "/",
        "AttachedManagedPolicies": [
            {
                "PolicyName": "AmazonS3FullAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
            },
            {
                "PolicyName": "AmazonDynamoDBFullAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/
AmazonDynamoDBFullAccess"
            }
        ],
        "RoleLastUsed": {
            "Region": "us-west-2",
            "LastUsedDate": "2019-11-13T17:30:00Z"
        },
        "RolePolicyList": [],
        "Arn": "arn:aws:iam::123456789012:role/EC2role"
    }
],
"GroupDetailList": [
    {
        "GroupId": "AIDA1234567890EXAMPLE",
        "AttachedManagedPolicies": {
```

```
                "PolicyName": "AdministratorAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
            },
            "GroupName": "Admins",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Admins",
            "CreateDate": "2013-10-14T18:32:24Z",
            "GroupPolicyList": []
        },
        {
            "GroupId": "AIDA1234567890EXAMPLE",
            "AttachedManagedPolicies": {
                "PolicyName": "PowerUserAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
            },
            "GroupName": "Dev",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Dev",
            "CreateDate": "2013-10-14T18:33:55Z",
            "GroupPolicyList": []
        },
        {
            "GroupId": "AIDA1234567890EXAMPLE",
            "AttachedManagedPolicies": [],
            "GroupName": "Finance",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Finance",
            "CreateDate": "2013-10-14T18:57:48Z",
            "GroupPolicyList": [
                {
                    "PolicyName": "policygen-201310141157",
                    "PolicyDocument": {
                        "Version":"2012-10-17",
                        "Statement": [
                            {
                                "Action": "aws-portal:*",
                                "Sid": "Stmt1381777017000",
                                "Resource": "*",
                                "Effect": "Allow"
                            }
                        ]
                    }
                }
            ]
```

```
        }
    ],
    "UserDetailList": [
        {
            "UserName": "Alice",
            "GroupList": [
                "Admins"
            ],
            "CreateDate": "2013-10-14T18:32:24Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [],
            "Path": "/",
            "AttachedManagedPolicies": [],
            "Arn": "arn:aws:iam::123456789012:user/Alice"
        },
        {
            "UserName": "Bob",
            "GroupList": [
                "Admins"
            ],
            "CreateDate": "2013-10-14T18:32:25Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [
                {
                    "PolicyName": "DenyBillingAndIAMPolicy",
                    "PolicyDocument": {
                        "Version":"2012-10-17",
                        "Statement": {
                            "Effect": "Deny",
                            "Action": [
                                "aws-portal:*",
                                "iam:*"
                            ],
                            "Resource": "*"
                        }
                    }
                }
            ],
            "Path": "/",
            "AttachedManagedPolicies": [],
            "Arn": "arn:aws:iam::123456789012:user/Bob"
        },
        {
            "UserName": "Charlie",
```

```
            "GroupList": [
                "Dev"
            ],
            "CreateDate": "2013-10-14T18:33:56Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [],
            "Path": "/",
            "AttachedManagedPolicies": [],
            "Arn": "arn:aws:iam::123456789012:user/Charlie"
        }
    ],
    "Policies": [
        {
            "PolicyName": "create-update-delete-set-managed-policies",
            "CreateDate": "2015-02-06T19:58:34Z",
            "AttachmentCount": 1,
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
                {
                    "CreateDate": "2015-02-06T19:58:34Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version":"2012-10-17",
                        "Statement": {
                            "Effect": "Allow",
                            "Action": [
                                "iam:CreatePolicy",
                                "iam:CreatePolicyVersion",
                                "iam:DeletePolicy",
                                "iam:DeletePolicyVersion",
                                "iam:GetPolicy",
                                "iam:GetPolicyVersion",
                                "iam:ListPolicies",
                                "iam:ListPolicyVersions",
                                "iam:SetDefaultPolicyVersion"
                            ],
                            "Resource": "*"
                        }
                    },
                    "IsDefaultVersion": true
                }
            ],
```

```
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:policy/create-update-delete-set-
managed-policies",
            "UpdateDate": "2015-02-06T19:58:34Z"
        },
        {
            "PolicyName": "S3-read-only-specific-bucket",
            "CreateDate": "2015-01-21T21:39:41Z",
            "AttachmentCount": 1,
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
                {
                    "CreateDate": "2015-01-21T21:39:41Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version":"2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Action": [
                                    "s3:Get*",
                                    "s3:List*"
                                ],
                                "Resource": [
                                    "arn:aws:s3:::amzn-s3-demo-bucket",
                                    "arn:aws:s3:::amzn-s3-demo-bucket/*"
                                ]
                            }
                        ]
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:policy/S3-read-only-specific-
bucket",
            "UpdateDate": "2015-01-21T23:39:41Z"
        },
        {
            "PolicyName": "AmazonEC2FullAccess",
            "CreateDate": "2015-02-06T18:40:15Z",
            "AttachmentCount": 1,
```

```
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
                {
                    "CreateDate": "2014-10-30T20:59:46Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version":"2012-10-17",
                        "Statement": [
                            {
                                "Action": "ec2:*",
                                "Effect": "Allow",
                                "Resource": "*"
                            },
                            {
                                "Effect": "Allow",
                                "Action": "elasticloadbalancing:*",
                                "Resource": "*"
                            },
                            {
                                "Effect": "Allow",
                                "Action": "cloudwatch:*",
                                "Resource": "*"
                            },
                            {
                                "Effect": "Allow",
                                "Action": "autoscaling:*",
                                "Resource": "*"
                            }
                        ]
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess",
            "UpdateDate": "2015-02-06T18:40:15Z"
        }
    ],
    "Marker": "EXAMPLEkakv9BCuUNFDtxWSyfzetYwEx2ADc8dnzfvERF5S6YMvXKx41t6gCl/
eeaCX3Jo94/bKqezEAg8TEVS99EKFLxm3jtbpl25FDWEXAMPLE",
    "IsTruncated": true
```

```
}
```

Per ulteriori informazioni, consulta Linee guida sugli audit di sicurezza AWS nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetAccountAuthorizationDetails AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio ottiene i dettagli di autorizzazione sulle identità nell' AWS account e visualizza l'elenco degli elementi dell'oggetto restituito, inclusi utenti, gruppi e ruoli. Ad esempio, la proprietà **`UserDetailList`** visualizza i dettagli degli utenti. Informazioni simili sono disponibili nelle proprietà **`RoleDetailList`** e **`GroupDetailList`**.

```
$Details=Get-IAMAccountAuthorizationDetail
$Details
```

Output:

```
GroupDetailList : {Administrators, Developers, Testers, Backup}
IsTruncated     : False
Marker          :
RoleDetailList  : {TestRole1, AdminRole, TesterRole, clirole...}
UserDetailList  : {Administrator, Bob, BackupToS3, }
```

```
$Details.UserDetailList
```

Output:

```
Arn             : arn:aws:iam::123456789012:user/Administrator
CreateDate      : 10/16/2014 9:03:09 AM
GroupList       : {Administrators}
Path            : /
UserId          : AIDACKCEVSQ6CEXAMPLE1
UserName        : Administrator
UserPolicyList : {}

Arn             : arn:aws:iam::123456789012:user/Bob
```

```
CreateDate     : 4/6/2015 12:54:42 PM
GroupList      : {Developers}
Path           : /
UserId         : AIDACKCEVSQ6CEXAMPLE2
UserName       : bab
UserPolicyList : {}

Arn            : arn:aws:iam::123456789012:user/BackupToS3
CreateDate     : 1/27/2015 10:15:08 AM
GroupList      : {Backup}
Path           : /
UserId         : AIDACKCEVSQ6CEXAMPLE3
UserName       : BackupToS3
UserPolicyList : {BackupServicePermissionsToS3Buckets}
```

- Per i dettagli sull'API, vedere GetAccountAuthorizationDetailsin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio ottiene i dettagli di autorizzazione sulle identità nell' AWS account e visualizza l'elenco degli elementi dell'oggetto restituito, inclusi utenti, gruppi e ruoli. Ad esempio, la proprietà **`UserDetailList`** visualizza i dettagli degli utenti. Informazioni simili sono disponibili nelle proprietà **`RoleDetailList`** e **`GroupDetailList`**.

```
$Details=Get-IAMAccountAuthorizationDetail
$Details
```

Output:

```
GroupDetailList : {Administrators, Developers, Testers, Backup}
IsTruncated     : False
Marker          :
RoleDetailList  : {TestRole1, AdminRole, TesterRole, clirole...}
UserDetailList  : {Administrator, Bob, BackupToS3, }
```

```
$Details.UserDetailList
```

Output:

```
Arn            : arn:aws:iam::123456789012:user/Administrator
CreateDate     : 10/16/2014 9:03:09 AM
```

```
GroupList      : {Administrators}
Path           : /
UserId         : AIDACKCEVSQ6CEXAMPLE1
UserName       : Administrator
UserPolicyList : {}

Arn            : arn:aws:iam::123456789012:user/Bob
CreateDate     : 4/6/2015 12:54:42 PM
GroupList      : {Developers}
Path           : /
UserId         : AIDACKCEVSQ6CEXAMPLE2
UserName       : bab
UserPolicyList : {}

Arn            : arn:aws:iam::123456789012:user/BackupToS3
CreateDate     : 1/27/2015 10:15:08 AM
GroupList      : {Backup}
Path           : /
UserId         : AIDACKCEVSQ6CEXAMPLE3
UserName       : BackupToS3
UserPolicyList : {BackupServicePermissionsToS3Buckets}
```

- Per i dettagli sull'API, vedere GetAccountAuthorizationDetailsin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def get_authorization_details(response_filter):
    """
    Gets an authorization detail report for the current account.

    :param response_filter: A list of resource types to include in the report,
 such
```

```
                            as users or roles. When not specified, all resources
                            are included.
    :return: The authorization detail report.
    """
    try:
        account_details = iam.meta.client.get_account_authorization_details(
            Filter=response_filter
        )
        logger.debug(account_details)
    except ClientError:
        logger.exception("Couldn't get details for your account.")
        raise
    else:
        return account_details
```

- Per i dettagli sull'API, consulta GetAccountAuthorizationDetails AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->getaccountauthdetails( ).
        MESSAGE 'Retrieved account authorization details.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when getting account authorization details.'
TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, GetAccountAuthorizationDetailsconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`GetAccountPasswordPolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetAccountPasswordPolicy`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Gets the IAM password policy for an AWS account.
    /// </summary>
    /// <returns>The PasswordPolicy for the AWS account.</returns>
    public async Task<PasswordPolicy> GetAccountPasswordPolicyAsync()
    {
        var response = await _IAMService.GetAccountPasswordPolicyAsync(new
GetAccountPasswordPolicyRequest());
        return response.PasswordPolicy;
    }
```

- Per i dettagli sull'API, consulta la GetAccountPasswordPolicysezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come visualizzare la policy delle password dell'account corrente

Il comando `get-account-password-policy` seguente visualizza dettagli sulla policy delle password per l'account corrente.

```
aws iam get-account-password-policy
```

Output:

```
{
    "PasswordPolicy": {
        "AllowUsersToChangePassword": false,
        "RequireLowercaseCharacters": false,
        "RequireUppercaseCharacters": false,
        "MinimumPasswordLength": 8,
        "RequireNumbers": true,
        "RequireSymbols": true
    }
}
```

Se non è definita alcuna policy delle password per l'account, il comando restituisce un errore `NoSuchEntity`.

Per ulteriori informazioni, consulta Impostazione di una policy delle password dell'account per utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetAccountPasswordPolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// AccountWrapper encapsulates AWS Identity and Access Management (IAM) account
 actions
// used in the examples.
// It contains an IAM service client that is used to perform account actions.
type AccountWrapper struct {
 IamClient *iam.Client
}



// GetAccountPasswordPolicy gets the account password policy for the current
 account.
// If no policy has been set, a NoSuchEntityException is error is returned.
func (wrapper AccountWrapper) GetAccountPasswordPolicy(ctx context.Context)
 (*types.PasswordPolicy, error) {
 var pwPolicy *types.PasswordPolicy
 result, err := wrapper.IamClient.GetAccountPasswordPolicy(ctx,
  &iam.GetAccountPasswordPolicyInput{})
 if err != nil {
  log.Printf("Couldn't get account password policy. Here's why: %v\n", err)
 } else {
  pwPolicy = result.PasswordPolicy
 }
 return pwPolicy, err
}
```

- Per i dettagli sull'API, consulta la GetAccountPasswordPolicysezione AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Recupera la policy sulla password dell'account.

```
import {
  GetAccountPasswordPolicyCommand,
  IAMClient,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});

export const getAccountPasswordPolicy = async () => {
  const command = new GetAccountPasswordPolicyCommand({});

  const response = await client.send(command);
  console.log(response.PasswordPolicy);
  return response;
};
```

- Per i dettagli sull'API, consulta la GetAccountPasswordPolicysezione AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function getAccountPasswordPolicy()
    {
        return $this->iamClient->getAccountPasswordPolicy();
    }
```

- Per i dettagli sull'API, consulta la GetAccountPasswordPolicysezione AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce dettagli della policy delle password per l'account corrente. Se non è definita alcuna policy delle password per l'account, il comando restituisce un errore **`NoSuchEntity`**.

```
Get-IAMAccountPasswordPolicy
```

Output:

```
AllowUsersToChangePassword : True
ExpirePasswords            : True
HardExpiry                 : False
MaxPasswordAge             : 90
MinimumPasswordLength      : 8
PasswordReusePrevention    : 20
RequireLowercaseCharacters : True
RequireNumbers             : True
RequireSymbols             : False
RequireUppercaseCharacters : True
```

- Per i dettagli sull'API, vedere GetAccountPasswordPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce dettagli della policy delle password per l'account corrente. Se non è definita alcuna policy delle password per l'account, il comando restituisce un errore **`NoSuchEntity`**.

```
Get-IAMAccountPasswordPolicy
```

Output:

```
AllowUsersToChangePassword : True
ExpirePasswords            : True
HardExpiry                 : False
MaxPasswordAge             : 90
MinimumPasswordLength      : 8
PasswordReusePrevention    : 20
RequireLowercaseCharacters : True
RequireNumbers             : True
RequireSymbols             : False
RequireUppercaseCharacters : True
```

- Per i dettagli sull'API, vedere GetAccountPasswordPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def print_password_policy():
    """
    Prints the password policy for the account.
    """
    try:
```

```
        pw_policy = iam.AccountPasswordPolicy()
        print("Current account password policy:")
        print(
            f"\tallow_users_to_change_password:
{pw_policy.allow_users_to_change_password}"
        )
        print(f"\texpire_passwords: {pw_policy.expire_passwords}")
        print(f"\thard_expiry: {pw_policy.hard_expiry}")
        print(f"\tmax_password_age: {pw_policy.max_password_age}")
        print(f"\tminimum_password_length: {pw_policy.minimum_password_length}")
        print(f"\tpassword_reuse_prevention:
{pw_policy.password_reuse_prevention}")
        print(
            f"\trequire_lowercase_characters:
{pw_policy.require_lowercase_characters}"
        )
        print(f"\trequire_numbers: {pw_policy.require_numbers}")
        print(f"\trequire_symbols: {pw_policy.require_symbols}")
        print(
            f"\trequire_uppercase_characters:
{pw_policy.require_uppercase_characters}"
        )
        printed = True
    except ClientError as error:
        if error.response["Error"]["Code"] == "NoSuchEntity":
            print("The account does not have a password policy set.")
        else:
            logger.exception("Couldn't get account password policy.")
            raise
    else:
        return printed
```

- Per i dettagli sull'API, consulta GetAccountPasswordPolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
# Class to manage IAM account password policies
class PasswordPolicyManager
  attr_accessor :iam_client, :logger

  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'IAMPolicyManager'
  end

  # Retrieves and logs the account password policy
  def print_account_password_policy
    response = @iam_client.get_account_password_policy
    @logger.info("The account password policy is:
 #{response.password_policy.to_h}")
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.info('The account does not have a password policy.')
  rescue Aws::Errors::ServiceError => e
    @logger.error("Couldn't print the account password policy. Error: #{e.code} -
 #{e.message}")
    raise
  end
end
```

- Per i dettagli sull'API, consulta la GetAccountPasswordPolicysezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn get_account_password_policy(
    client: &iamClient,
) -> Result<GetAccountPasswordPolicyOutput,
 SdkError<GetAccountPasswordPolicyError>> {
    let response = client.get_account_password_policy().send().await?;

    Ok(response)
}
```

- Per i dettagli sulle API, consulta la GetAccountPasswordPolicyguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->getaccountpasswordpolicy( ).
        MESSAGE 'Retrieved account password policy.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'No password policy exists.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when getting password policy.' TYPE 'E'.
```

```
    ENDTRY.
```

- Per i dettagli sulle API, GetAccountPasswordPolicyconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`GetAccountSummary`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetAccountSummary`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

CLI

AWS CLI

Come ottenere informazioni sull'utilizzo delle entità IAM e sulle quote IAM nell'account corrente

Il comando `get-account-summary` seguente restituisce informazioni sull'utilizzo corrente delle entità IAM e sulle quote correnti delle entità IAM nell'account.

```
aws iam get-account-summary
```

Output:

```
{
    "SummaryMap": {
        "UsersQuota": 5000,
        "GroupsQuota": 100,
        "InstanceProfiles": 6,
        "SigningCertificatesPerUserQuota": 2,
        "AccountAccessKeysPresent": 0,
        "RolesQuota": 250,
        "RolePolicySizeQuota": 10240,
```

```
        "AccountSigningCertificatesPresent": 0,
        "Users": 27,
        "ServerCertificatesQuota": 20,
        "ServerCertificates": 0,
        "AssumeRolePolicySizeQuota": 2048,
        "Groups": 7,
        "MFADevicesInUse": 1,
        "Roles": 3,
        "AccountMFAEnabled": 1,
        "MFADevices": 3,
        "GroupsPerUserQuota": 10,
        "GroupPolicySizeQuota": 5120,
        "InstanceProfilesQuota": 100,
        "AccessKeysPerUserQuota": 2,
        "Providers": 0,
        "UserPolicySizeQuota": 2048
    }
}
```

Per ulteriori informazioni sulle limitazioni delle entità, consulta le quote IAM e AWS STS nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta GetAccountSummary AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce informazioni sull'utilizzo corrente e sulle quote correnti delle entità IAM nell' Account AWS.

```
Get-IAMAccountSummary
```

Output:

```
Key                                     Value
Users                                   7
GroupPolicySizeQuota                    5120
PolicyVersionsInUseQuota                10000
ServerCertificatesQuota                 20
AccountSigningCertificatesPresent       0
```

```
AccountAccessKeysPresent                0
Groups                                  3
UsersQuota                              5000
RolePolicySizeQuota                     10240
UserPolicySizeQuota                     2048
GroupsPerUserQuota                      10
AssumeRolePolicySizeQuota               2048
AttachedPoliciesPerGroupQuota           2
Roles                                   9
VersionsPerPolicyQuota                  5
GroupsQuota                             100
PolicySizeQuota                         5120
Policies                                5
RolesQuota                              250
ServerCertificates                      0
AttachedPoliciesPerRoleQuota            2
MFADevicesInUse                         2
PoliciesQuota                           1000
AccountMFAEnabled                       1
Providers                               2
InstanceProfilesQuota                   100
MFADevices                              4
AccessKeysPerUserQuota                  2
AttachedPoliciesPerUserQuota            2
SigningCertificatesPerUserQuota         2
PolicyVersionsInUse                     4
InstanceProfiles                        1
...
```

- Per i dettagli sull'API, vedere GetAccountSummaryin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce informazioni sull'utilizzo corrente e sulle quote correnti delle entità IAM nell' Account AWS.

```
Get-IAMAccountSummary
```

Output:

```
Key                                     Value
```

```
Users                                7
GroupPolicySizeQuota                 5120
PolicyVersionsInUseQuota             10000
ServerCertificatesQuota              20
AccountSigningCertificatesPresent    0
AccountAccessKeysPresent             0
Groups                               3
UsersQuota                           5000
RolePolicySizeQuota                  10240
UserPolicySizeQuota                  2048
GroupsPerUserQuota                   10
AssumeRolePolicySizeQuota            2048
AttachedPoliciesPerGroupQuota        2
Roles                                9
VersionsPerPolicyQuota               5
GroupsQuota                          100
PolicySizeQuota                      5120
Policies                             5
RolesQuota                           250
ServerCertificates                   0
AttachedPoliciesPerRoleQuota         2
MFADevicesInUse                      2
PoliciesQuota                        1000
AccountMFAEnabled                    1
Providers                            2
InstanceProfilesQuota                100
MFADevices                           4
AccessKeysPerUserQuota               2
AttachedPoliciesPerUserQuota         2
SigningCertificatesPerUserQuota      2
PolicyVersionsInUse                  4
InstanceProfiles                     1
...
```

- Per i dettagli sull'API, vedere GetAccountSummaryin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def get_summary():
    """
    Gets a summary of account usage.

    :return: The summary of account usage.
    """
    try:
        summary = iam.AccountSummary()
        logger.debug(summary.summary_map)
    except ClientError:
        logger.exception("Couldn't get a summary for your account.")
        raise
    else:
        return summary.summary_map
```

- Per i dettagli sull'API, consulta GetAccountSummary AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->getaccountsummary( ).
        MESSAGE 'Retrieved account summary.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when getting account summary.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, GetAccountSummaryconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetContextKeysForCustomPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetContextKeysForCustomPolicy`.

CLI

AWS CLI

Esempio 1: per elencare le chiavi di contesto a cui fanno riferimento una o più policy JSON personalizzate fornite come parametro nella riga di comando

Il comando `get-context-keys-for-custom-policy` seguente analizza ogni policy fornita ed elenca le chiavi di contesto utilizzate da tali policy. Utilizza questo comando per identificare i valori delle chiavi di contesto che è necessario fornire per utilizzare correttamente i comandi `simulate-custom-policy` e `simulate-custom-policy` del simulatore di policy. Puoi anche recuperare l'elenco delle chiavi di contesto utilizzate da tutte le policy associate da un utente o ruolo IAM utilizzando il comando `get-context-keys-for-custom-policy`. I parametri che iniziano con `file://` indicano al comando di leggere il file e di utilizzarne il contenuto come valore del parametro al posto del nome del file.

```
aws iam get-context-keys-for-custom-policy \
    --policy-input-list '{"Version":"2012-10-17",       "Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/${aws:username}","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
```

Output:

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

Esempio 2: come elencare le chiavi di contesto a cui fanno riferimento una o più policy JSON personalizzate fornite come input di file

Il comando `get-context-keys-for-custom-policy` seguente è uguale all'esempio precedente tranne per il fatto che le policy vengono fornite in un file anziché come parametro. Poiché il comando prevede un elenco di stringhe JSON e non un elenco di strutture JSON, il file deve essere strutturato come segue, sebbene sia possibile comprimerlo in un unico file.

```
[
    "Policy1",
    "Policy2"
]
```

Ad esempio, un file che contiene la policy dell'esempio precedente deve avere l'aspetto seguente. È necessario effettuare l'escape di ogni virgoletta doppia incorporata nella stringa della policy facendola precedere da una barra rovesciata ''.

```
[ "{\"Version\": \"2012-10-17\", \"Statement\": {\"Effect\": \"Allow
\", \"Action\": \"dynamodb:*\", \"Resource\": \"arn:aws:dynamodb:us-
west-2:128716708097:table/${aws:username}\", \"Condition\": {\"DateGreaterThan\":
 {\"aws:CurrentTime\": \"2015-08-16T12:00:00Z\"}}}}" ]
```

Questo file può quindi essere inviato al seguente comando.

```
aws iam get-context-keys-for-custom-policy \
    --policy-input-list file://policyfile.json
```

Output:

```
{
    "ContextKeyNames": [
        "aws:username",
```

```
        "aws:CurrentTime"
    ]
}
```

Per ulteriori informazioni, consulta Using the IAM Policy Simulator (AWS CLI AWS e API) nella IAM User AWS Guide.

- Per i dettagli sull'API, consulta AWS CLI Command GetContextKeysForCustomPolicyReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera tutte le chiavi di contesto presenti nella policy JSON fornita. Per fornire più policy puoi specificare un elenco di valori separati da virgole.

```
$policy1 = '{"Version":"2012-10-17",       "Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
$policy2 = '{"Version":"2012-10-17",       "Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/"}}'
Get-IAMContextKeysForCustomPolicy -PolicyInputList $policy1,$policy2
```

- Per i dettagli sull'API, vedere GetContextKeysForCustomPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera tutte le chiavi di contesto presenti nella policy JSON fornita. Per fornire più policy puoi specificare un elenco di valori separati da virgole.

```
$policy1 = '{"Version":"2012-10-17",       "Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
$policy2 = '{"Version":"2012-10-17",       "Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/"}}'
Get-IAMContextKeysForCustomPolicy -PolicyInputList $policy1,$policy2
```

- Per i dettagli sull'API, vedere GetContextKeysForCustomPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetContextKeysForPrincipalPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetContextKeysForPrincipalPolicy`.

CLI

AWS CLI

Per visualizzare le chiavi di contesto a cui fanno riferimento tutte le policy associate a un principale IAM

Il comando `get-context-keys-for-principal-policy` seguente recupera tutte le policy collegate all'utente `saanvi` e ai gruppi di cui è membro. Quindi analizza ciascuna di esse ed elenca le chiavi di contesto utilizzate da tali policy. Utilizza questo comando per identificare i valori delle chiavi di contesto che è necessario fornire per utilizzare correttamente i comandi `simulate-custom-policy` e `simulate-principal-policy`. È inoltre possibile recuperare l'elenco delle chiavi di contesto utilizzate da una policy JSON arbitraria utilizzando il comando `get-context-keys-for-custom-policy`.

```
aws iam get-context-keys-for-principal-policy \
   --policy-source-arn arn:aws:iam::123456789012:user/saanvi
```

Output:

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

Per ulteriori informazioni, consulta Using the IAM Policy Simulator (AWS CLI AWS e API) nella IAM User AWS Guide.

- Per i dettagli sull'API, consulta AWS CLI Command GetContextKeysForPrincipalPolicyReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio recupera tutte le chiavi di contesto presenti nella policy json fornita e le politiche allegate all'entità IAM (user/role ecc.). Per: PolicyInputList puoi fornire più elenchi di valori come valori separati da virgole.

```
$policy1 = '{"Version":"2012-10-17",       "Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
$policy2 = '{"Version":"2012-10-17",       "Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/"}}'
Get-IAMContextKeysForPrincipalPolicy -PolicyInputList $policy1,$policy2 -
PolicySourceArn arn:aws:iam::852640994763:user/TestUser
```

- Per i dettagli sull'API, vedere GetContextKeysForPrincipalPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio recupera tutte le chiavi di contesto presenti nella policy json fornita e le politiche allegate all'entità IAM (user/role ecc.). Per: PolicyInputList puoi fornire più elenchi di valori come valori separati da virgole.

```
$policy1 = '{"Version":"2012-10-17",       "Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
$policy2 = '{"Version":"2012-10-17",       "Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/"}}'
Get-IAMContextKeysForPrincipalPolicy -PolicyInputList $policy1,$policy2 -
PolicySourceArn arn:aws:iam::852640994763:user/TestUser
```

- Per i dettagli sull'API, vedere GetContextKeysForPrincipalPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`GetCredentialReport`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetCredentialReport`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

CLI

AWS CLI

Come ottenere un report delle credenziali

Questo esempio apre il report restituito e lo invia alla pipeline come array di righe di testo.

```
aws iam get-credential-report
```

Output:

```
{
    "GeneratedTime":  "2015-06-17T19:11:50Z",
    "ReportFormat": "text/csv"
}
```

Per ulteriori informazioni, consulta Ottenere i report sulle credenziali per il tuo AWS account nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta GetCredentialReport AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio apre il report restituito e lo invia alla pipeline come array di righe di testo. La prima riga è l'intestazione con i nomi delle colonne separati da virgole.

Ogni riga successiva è la riga di dettaglio per un utente, con ogni campo separato da virgole. Prima di poter visualizzare il report, è necessario generarlo con il cmdlet **`Request-IAMCredentialReport`**. Per recuperare il report come singola stringa, utilizzare **`-Raw`** invece di **`-AsTextArray`**. L'alias **`-SplitLines`** è accettato anche per lo switch **`-AsTextArray`**. Per l'elenco completo delle colonne nell'output, consulta la documentazione di riferimento delle API del servizio. Tieni presente che se non utilizzi **`-AsTextArray`** o**`-SplitLines`**, devi estrarre il testo dalla proprietà **`.Content`** utilizzando la classe **`StreamReader`** .NET.

```
Request-IAMCredentialReport
```

Output:

```
Description                                                        State
-----------                                                        -----
No report exists. Starting a new report generation task            STARTED
```

```
Get-IAMCredentialReport -AsTextArray
```

Output:

```
 user,arn,user_creation_time,password_enabled,password_last_used,password_last_changed,pa
 root_account,arn:aws:iam::123456789012:root,2014-10-15T16:31:25+00:00,not_supported,2015
A,false,N/A,false,N/A,false,N/A
Administrator,arn:aws:iam::123456789012:user/
Administrator,2014-10-16T16:03:09+00:00,true,2015-04-20T15:18:32+00:00,2014-10-16T16:06:0
A,false,true,2014-12-03T18:53:41+00:00,true,2015-03-25T20:38:14+00:00,false,N/
A,false,N/A
Bill,arn:aws:iam::123456789012:user/Bill,2015-04-15T18:27:44+00:00,false,N/A,N/
A,N/A,false,false,N/A,false,N/A,false,2015-04-20T20:00:12+00:00,false,N/A
```

- Per i dettagli sull'API, vedere GetCredentialReportin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio apre il report restituito e lo invia alla pipeline come array di righe di testo. La prima riga è l'intestazione con i nomi delle colonne separati da virgole.

Ogni riga successiva è la riga di dettaglio per un utente, con ogni campo separato da virgole. Prima di poter visualizzare il report, è necessario generarlo con il cmdlet **`Request-IAMCredentialReport`**. Per recuperare il report come singola stringa, utilizzare **`-Raw`** invece di **`-AsTextArray`**. L'alias **`-SplitLines`** è accettato anche per lo switch **`-AsTextArray`**. Per l'elenco completo delle colonne nell'output, consulta la documentazione di riferimento delle API del servizio. Tieni presente che se non utilizzi **`-AsTextArray`** o**`-SplitLines`**, devi estrarre il testo dalla proprietà **`.Content`** utilizzando la classe **`StreamReader`** .NET.

```
Request-IAMCredentialReport
```

Output:

```
Description                                                    State
-----------                                                    -----
No report exists. Starting a new report generation task        STARTED
```

```
Get-IAMCredentialReport -AsTextArray
```

Output:

```
 user,arn,user_creation_time,password_enabled,password_last_used,password_last_changed,pa
 root_account,arn:aws:iam::123456789012:root,2014-10-15T16:31:25+00:00,not_supported,2015
A,false,N/A,false,N/A,false,N/A
Administrator,arn:aws:iam::123456789012:user/
Administrator,2014-10-16T16:03:09+00:00,true,2015-04-20T15:18:32+00:00,2014-10-16T16:06:0
A,false,true,2014-12-03T18:53:41+00:00,true,2015-03-25T20:38:14+00:00,false,N/
A,false,N/A
Bill,arn:aws:iam::123456789012:user/Bill,2015-04-15T18:27:44+00:00,false,N/A,N/
A,N/A,false,false,N/A,false,N/A,false,2015-04-20T20:00:12+00:00,false,N/A
```

- Per i dettagli sull'API, vedere GetCredentialReportin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def get_credential_report():
    """
    Gets the most recently generated credentials report about the current
 account.

    :return: The credentials report.
    """
    try:
        response = iam.meta.client.get_credential_report()
        logger.debug(response["Content"])
    except ClientError:
        logger.exception("Couldn't get credentials report.")
        raise
    else:
        return response["Content"]
```

- Per i dettagli sull'API, consulta GetCredentialReport AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->getcredentialreport( ).
        MESSAGE 'Retrieved credential report.' TYPE 'I'.
      CATCH /aws1/cx_iamcredrptnotpresen00.
        MESSAGE 'Credential report not present.' TYPE 'E'.
      CATCH /aws1/cx_iamcredrptexpiredex.
        MESSAGE 'Credential report expired.' TYPE 'E'.
      CATCH /aws1/cx_iamcredrptnotreadyex.
        MESSAGE 'Credential report not ready.' TYPE 'E'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when getting credential report.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, GetCredentialReportconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetGroup`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetGroup`.

CLI

AWS CLI

Per ottenere un gruppo IAM

Questo esempio restituisce dettagli del gruppo IAM `Admins`.

```
aws iam get-group \
    --group-name Admins
```

Output:

```
{
    "Group": {
        "Path": "/",
```

```
        "CreateDate": "2015-06-16T19:41:48Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    },
    "Users": []
}
```

Per ulteriori informazioni, consulta Identità IAM (utenti, gruppi e ruoli) nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta GetGroup AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce dettagli del gruppo IAM **`Testers`**, inclusa una raccolta di tutti gli utenti IAM che appartengono al gruppo.

```
$results = Get-IAMGroup -GroupName "Testers"
$results
```

Output:

```
Group                                    IsTruncated           Marker
     Users
-----                                    -----------           ------
     -----
Amazon.IdentityManagement.Model.Group    False
     {Theresa, David}
```

```
$results.Group
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Testers
CreateDate : 12/10/2014 3:39:11 PM
GroupId    : 3RHNZZGQJ7QHMAEXAMPLE1
GroupName  : Testers
Path       : /
```

```
$results.Users
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Theresa
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 1/1/0001 12:00:00 AM
Path             : /
UserId           : 4OSVDDJJTF4XEEXAMPLE2
UserName         : Theresa

Arn              : arn:aws:iam::123456789012:user/David
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 3/19/2015 8:44:04 AM
Path             : /
UserId           : Y4FKWQCXTA52QEXAMPLE3
UserName         : David
```

- Per i dettagli sull'API, vedere GetGroupin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce dettagli del gruppo IAM **`Testers`**, inclusa una raccolta di tutti gli utenti IAM che appartengono al gruppo.

```
$results = Get-IAMGroup -GroupName "Testers"
$results
```

Output:

```
Group                                    IsTruncated           Marker
     Users
-----                                    -----------           ------
     -----
Amazon.IdentityManagement.Model.Group    False
     {Theresa, David}
```

```
$results.Group
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Testers
CreateDate : 12/10/2014 3:39:11 PM
GroupId    : 3RHNZZGQJ7QHMAEXAMPLE1
GroupName  : Testers
Path       : /
```

```
$results.Users
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Theresa
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 1/1/0001 12:00:00 AM
Path             : /
UserId           : 4OSVDDJJTF4XEEXAMPLE2
UserName         : Theresa

Arn              : arn:aws:iam::123456789012:user/David
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 3/19/2015 8:44:04 AM
Path             : /
UserId           : Y4FKWQCXTA52QEXAMPLE3
UserName         : David
```

- Per i dettagli sull'API, vedere GetGroupin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetGroupPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetGroupPolicy`.

CLI

AWS CLI

Per ottenere informazioni su una policy collegata a un gruppo IAM

Il comando `get-group-policy` seguente ottiene informazioni sulla policy specificata collegata al gruppo denominato `Test-Group`.

```
aws iam get-group-policy \
    --group-name Test-Group \
    --policy-name S3-ReadOnly-Policy
```

Output:

```
{
    "GroupName": "Test-Group",
    "PolicyDocument": {
        "Statement": [
            {
                "Action": [
                    "s3:Get*",
                    "s3:List*"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    },
    "PolicyName": "S3-ReadOnly-Policy"
}
```

Per ulteriori informazioni, consulta Gestione delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta GetGroupPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce dettagli della policy in linea incorporata denominata **`PowerUserAccess-Testers`** per il gruppo **`Testers`**. La proprietà **`PolicyDocument`** è codificata tramite URL. In questo esempio viene decodificato con il metodo **`UrlDecode`** .NET.

```
$results = Get-IAMGroupPolicy -GroupName Testers -PolicyName PowerUserAccess-
Testers
```

```
$results
```

Output:

```
GroupName     PolicyDocument
 PolicyName
---------     --------------
 ----------
Testers       %7B%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C%0A%20...
 PowerUserAccess-Testers

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.PolicyDocument)
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:555555555555:instance/i-b188560f"
      ]
    }
  ]
}
```

- Per i dettagli sull'API, vedere GetGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce dettagli della policy in linea incorporata denominata **`PowerUserAccess-Testers`** per il gruppo **`Testers`**. La proprietà **`PolicyDocument`** è codificata tramite URL. In questo esempio viene decodificato con il metodo **`UrlDecode`** .NET.

```
$results = Get-IAMGroupPolicy -GroupName Testers -PolicyName PowerUserAccess-
Testers
$results
```

Output:

```
GroupName     PolicyDocument
 PolicyName
---------     --------------
 ----------
Testers       %7B%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C%0A%20...
 PowerUserAccess-Testers

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.PolicyDocument)
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:555555555555:instance/i-b188560f"
      ]
    }
  ]
}
```

- Per i dettagli sull'API, vedere GetGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetInstanceProfile`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetInstanceProfile`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Guida introduttiva allo streaming gestito

CLI

AWS CLI

Come ottenere informazioni su un profilo dell'istanza

Il comando `get-instance-profile` seguente ottiene informazioni sul profilo dell'istanza denominato `ExampleInstanceProfile`.

```
aws iam get-instance-profile \
    --instance-profile-name ExampleInstanceProfile
```

Output:

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AID2MAB8DPLSRHEXAMPLE",
        "Roles": [
            {
                "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
                "RoleId": "AIDGPMS9RO4H3FEXAMPLE",
                "CreateDate": "2013-01-09T06:33:26Z",
                "RoleName": "Test-Role",
                "Path": "/",
                "Arn": "arn:aws:iam::336924118301:role/Test-Role"
            }
        ],
        "CreateDate": "2013-06-12T23:52:02Z",
        "InstanceProfileName": "ExampleInstanceProfile",
        "Path": "/",
        "Arn": "arn:aws:iam::336924118301:instance-profile/
ExampleInstanceProfile"
    }
}
```

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetInstanceProfile AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio restituisce i dettagli del profilo di istanza denominato **`ec2instancerole`** definito nell' AWS account corrente.

```
Get-IAMInstanceProfile -InstanceProfileName ec2instancerole
```

Output:

```
Arn                 : arn:aws:iam::123456789012:instance-profile/ec2instancerole
CreateDate          : 2/17/2015 2:49:04 PM
InstanceProfileId   : HH36PTZQJUR32EXAMPLE1
InstanceProfileName : ec2instancerole
Path                : /
Roles               : {ec2instancerole}
```

- Per i dettagli sull'API, vedere GetInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio restituisce i dettagli del profilo di istanza denominato **`ec2instancerole`** definito nell' AWS account corrente.

```
Get-IAMInstanceProfile -InstanceProfileName ec2instancerole
```

Output:

```
Arn                 : arn:aws:iam::123456789012:instance-profile/ec2instancerole
CreateDate          : 2/17/2015 2:49:04 PM
InstanceProfileId   : HH36PTZQJUR32EXAMPLE1
InstanceProfileName : ec2instancerole
Path                : /
Roles               : {ec2instancerole}
```

- Per i dettagli sull'API, vedere GetInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetLoginProfile`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetLoginProfile`.

CLI

AWS CLI

Per ottenere informazioni sulla password di un utente IAM

Il comando `get-login-profile` seguente ottiene informazioni sulla password per l'utente IAM denominato Bob.

```
aws iam get-login-profile \
    --user-name Bob
```

Output:

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2012-09-21T23:03:39Z"
    }
}
```

Il comando `get-login-profile` può essere utilizzato per verificare che un utente IAM disponga di una password. Se non è definita alcuna password per l'utente, il comando restituisce un errore `NoSuchEntity`.

Non è possibile visualizzare una password utilizzando questo comando. Se dimentichi la password, puoi reimpostare la password (`update-login-profile`) dell'utente. In alternativa, è possibile eliminare il profilo di accesso (`delete-login-profile`) per l'utente e quindi crearne uno nuovo (`create-login-profile`).

Per ulteriori informazioni, consulta Gestione delle password per gli utenti IAM nella Guida per l'utente di AWS .

- Per i dettagli sull'API, consulta GetLoginProfile AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce la data di creazione della password e se è necessaria una reimpostazione della password per l'utente IAM **David**.

```
Get-IAMLoginProfile -UserName David
```

Output:

```
CreateDate                 PasswordResetRequired              UserName
----------                 ---------------------              --------
12/10/2014 3:39:44 PM      False                              David
```

- Per i dettagli sull'API, vedere GetLoginProfilein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce la data di creazione della password e se è necessaria una reimpostazione della password per l'utente IAM **David**.

```
Get-IAMLoginProfile -UserName David
```

Output:

```
CreateDate                 PasswordResetRequired              UserName
----------                 ---------------------              --------
12/10/2014 3:39:44 PM      False                              David
```

- Per i dettagli sull'API, vedere GetLoginProfilein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetOpenIdConnectProvider`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetOpenIdConnectProvider`.

CLI

AWS CLI

Per restituire informazioni sul provider OpenID Connect specificato

Questo esempio restituisce i dettagli del provider OpenID Connect il cui ARN è `arn:aws:iam::123456789012:oidc-provider/server.example.com`.

```
aws iam get-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

Output:

```
{
    "Url": "server.example.com"
        "CreateDate": "2015-06-16T19:41:48Z",
        "ThumbprintList": [
        "12345abcdefghijk67890lmnopqrst987example"
        ],
        "ClientIDList": [
        "example-application-ID"
        ]
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità OpenID Connect (OIDC) nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta GetOpenIdConnectProvider AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce i dettagli del provider OpenID Connect il cui ARN è **`arn:aws:iam::123456789012:oidc-provider/accounts.google.com`**. La proprietà **`ClientIDList`** è una raccolta che contiene tutti gli ID client definiti per questo provider.

```
Get-IAMOpenIDConnectProvider -OpenIDConnectProviderArn
 arn:aws:iam::123456789012:oidc-provider/oidc.example.com
```

Output:

```
ClientIDList         CreateDate                ThumbprintList
           Url
------------         ----------                --------------
           ---
{MyOIDCApp}          2/3/2015 3:00:30 PM
 {12345abcdefghijk67890lmnopqrst98765uvwxy}   oidc.example.com
```

- Per i dettagli sull'API, vedere GetOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce i dettagli del provider OpenID Connect il cui ARN è **`arn:aws:iam::123456789012:oidc-provider/accounts.google.com`**. La proprietà **`ClientIDList`** è una raccolta che contiene tutti gli ID client definiti per questo provider.

```
Get-IAMOpenIDConnectProvider -OpenIDConnectProviderArn
 arn:aws:iam::123456789012:oidc-provider/oidc.example.com
```

Output:

```
ClientIDList         CreateDate                ThumbprintList
           Url
------------         ----------                --------------
           ---
{MyOIDCApp}          2/3/2015 3:00:30 PM
 {12345abcdefghijk67890lmnopqrst98765uvwxy}   oidc.example.com
```

- Per i dettagli sull'API, vedere GetOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`GetPolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Guida introduttiva allo streaming gestito
- Lavora con l'API IAM Policy Builder

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Get information about an IAM policy.
    /// </summary>
    /// <param name="policyArn">The IAM policy to retrieve information for.</
param>
    /// <returns>The IAM policy.</returns>
    public async Task<ManagedPolicy> GetPolicyAsync(string policyArn)
    {

        var response = await _IAMService.GetPolicyAsync(new GetPolicyRequest
 { PolicyArn = policyArn });
        return response.Policy;
    }
```

- Per i dettagli sull'API, consulta la GetPolicysezione AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::getPolicy(const Aws::String &policyArn,
                            const Aws::Client::ClientConfiguration &clientConfig)
 {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::GetPolicyRequest request;
    request.SetPolicyArn(policyArn);

    auto outcome = iam.GetPolicy(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error getting policy " << policyArn << ": " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        const auto &policy = outcome.GetResult().GetPolicy();
        std::cout << "Name: " << policy.GetPolicyName() << std::endl <<
                  "ID: " << policy.GetPolicyId() << std::endl << "Arn: " <<
                  policy.GetArn() << std::endl << "Description: " <<
                  policy.GetDescription() << std::endl << "CreateDate: " <<

 policy.GetCreateDate().ToGmtString(Aws::Utils::DateFormat::ISO_8601)
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la GetPolicysezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come recuperare informazioni sulla policy gestita specificata

Questo esempio restituisce i dettagli sulla policy gestita il cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam get-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Output:

```
{
    "Policy": {
        "PolicyName": "MySamplePolicy",
        "CreateDate": "2015-06-17T19:23;32Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "Z27SI6FQMGNQ2EXAMPLE1",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:policy/MySamplePolicy",
        "UpdateDate": "2015-06-17T19:23:32Z"
    }
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetPolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// GetPolicy gets data about a policy.
func (wrapper PolicyWrapper) GetPolicy(ctx context.Context, policyArn string)
 (*types.Policy, error) {
 var policy *types.Policy
 result, err := wrapper.IamClient.GetPolicy(ctx, &iam.GetPolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't get policy %v. Here's why: %v\n", policyArn, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}
```

- Per i dettagli sull'API, consulta la GetPolicysezione AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Recupera la policy.

```
import { GetPolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} policyArn
 */
export const getPolicy = (policyArn) => {
  const command = new GetPolicyCommand({
    PolicyArn: policyArn,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la GetPolicysezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  PolicyArn: "arn:aws:iam::aws:policy/AWSLambdaExecute",
};

iam.getPolicy(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.Policy.Description);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la GetPolicysezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun getIAMPolicy(policyArnVal: String?) {
    val request =
        GetPolicyRequest {
            policyArn = policyArnVal
        }
```

```
    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.getPolicy(request)
        println("Successfully retrieved policy ${response.policy?.policyName}")
    }
}
```

- Per i dettagli sull'API, GetPolicyconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function getPolicy($policyArn)
    {
        return $this->customWaiter(function () use ($policyArn) {
            return $this->iamClient->getPolicy(['PolicyArn' => $policyArn]);
        });
    }
```

- Per i dettagli sull'API, consulta la GetPolicysezione AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce i dettagli sulla policy gestita il cui ARN è **`arn:aws:iam::123456789012:policy/MySamplePolicy`**.

```
Get-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Output:

```
Arn              : arn:aws:iam::aws:policy/MySamplePolicy
AttachmentCount  : 0
CreateDate       : 2/6/2015 10:40:08 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : Z27SI6FQMGNQ2EXAMPLE1
PolicyName       : MySamplePolicy
UpdateDate       : 2/6/2015 10:40:08 AM
```

- Per i dettagli sull'API, vedere GetPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce i dettagli sulla policy gestita il cui ARN è **`arn:aws:iam::123456789012:policy/MySamplePolicy`**.

```
Get-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Output:

```
Arn              : arn:aws:iam::aws:policy/MySamplePolicy
AttachmentCount  : 0
CreateDate       : 2/6/2015 10:40:08 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : Z27SI6FQMGNQ2EXAMPLE1
PolicyName       : MySamplePolicy
UpdateDate       : 2/6/2015 10:40:08 AM
```

- Per i dettagli sull'API, vedere GetPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def get_default_policy_statement(policy_arn):
    """
    Gets the statement of the default version of the specified policy.

    :param policy_arn: The ARN of the policy to look up.
    :return: The statement of the default policy version.
    """
    try:
        policy = iam.Policy(policy_arn)
        # To get an attribute of a policy, the SDK first calls get_policy.
        policy_doc = policy.default_version.document
        policy_statement = policy_doc.get("Statement", None)
        logger.info("Got default policy doc for %s.", policy.policy_name)
        logger.info(policy_doc)
    except ClientError:
        logger.exception("Couldn't get default policy statement for %s.",
 policy_arn)
        raise
    else:
        return policy_statement
```

- Per i dettagli sull'API, consulta GetPolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
 # Fetches an IAM policy by its ARN
 # @param policy_arn [String] the ARN of the IAM policy to retrieve
 # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
 def get_policy(policy_arn)
   response = @iam_client.get_policy(policy_arn: policy_arn)
   policy = response.policy
   @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
   policy
 rescue Aws::IAM::Errors::NoSuchEntity
   @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
   raise
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
   raise
 end
```

- Per i dettagli sull'API, consulta la GetPolicysezione AWS SDK per Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->getpolicy( iv_policyarn = iv_policy_arn ).
        MESSAGE 'Retrieved policy information.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Policy does not exist.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, GetPolicyconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func getPolicy(arn: String) async throws -> IAMClientTypes.Policy {
        let input = GetPolicyInput(
            policyArn: arn
        )
        do {
            let output = try await client.getPolicy(input: input)
            guard let policy = output.policy else {
                throw ServiceHandlerError.noSuchPolicy
            }
            return policy
        } catch {
            print("ERROR: getPolicy:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la GetPolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`GetPolicyVersion`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetPolicyVersion`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Gestione delle policy
- Lavora con l'API IAM Policy Builder

CLI

AWS CLI

Come recuperare informazioni sulla versione specificata della policy gestita specificata

Questo esempio restituisce il documento della policy per la versione v2 della policy il cui ARN è `arn:aws:iam::123456789012:policy/MyManagedPolicy`.

```
aws iam get-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

Output:

```
{
    "PolicyVersion": {
        "Document": {
            "Version":"2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": "iam:*",
                    "Resource": "*"
```

```
                }
            ]
        },
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2023-04-11T00:22:54+00:00"
    }
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetPolicyVersion AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce il documento della policy per la versione **v2** della policy il cui ARN è **arn:aws:iam::123456789012:policy/MyManagedPolicy**. Il documento di policy contenuto nella proprietà **Document** è codificato nell'URL e in questo esempio viene decodificato con il metodo .NET **UrlDecode**.

```
$results = Get-IAMPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/
MyManagedPolicy -VersionId v2
$results
```

Output:

```
CreateDate             Document
 IsDefaultVersion     VersionId
----------             --------
 ----------------     ---------
2/12/2015 9:39:53 AM   %7B%0A%20%20%22Version%22%3A%20%222012-10...    True
           v2

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
$policy = [System.Web.HttpUtility]::UrlDecode($results.Document)
$policy
{
  "Version":"2012-10-17",
```

```
  "Statement":
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:555555555555:instance/i-b188560f"
      ]
    }
}
```

- Per i dettagli sull'API, vedere GetPolicyVersionin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce il documento della policy per la versione **v2** della policy il cui ARN è **arn:aws:iam::123456789012:policy/MyManagedPolicy**. Il documento di policy contenuto nella proprietà **Document** è codificato nell'URL e in questo esempio viene decodificato con il metodo .NET **UrlDecode**.

```
$results = Get-IAMPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/
MyManagedPolicy -VersionId v2
$results
```

Output:

```
CreateDate             Document
 IsDefaultVersion     VersionId
----------             --------
 ----------------     ---------
2/12/2015 9:39:53 AM   %7B%0A%20%20%22Version%22%3A%20%222012-10...    True
          v2

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
$policy = [System.Web.HttpUtility]::UrlDecode($results.Document)
$policy
{
  "Version":"2012-10-17",
  "Statement":
    {
```

```
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:555555555555:instance/i-b188560f"
      ]
    }
}
```

- Per i dettagli sull'API, vedere GetPolicyVersionin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def get_default_policy_statement(policy_arn):
    """
    Gets the statement of the default version of the specified policy.

    :param policy_arn: The ARN of the policy to look up.
    :return: The statement of the default policy version.
    """
    try:
        policy = iam.Policy(policy_arn)
        # To get an attribute of a policy, the SDK first calls get_policy.
        policy_doc = policy.default_version.document
        policy_statement = policy_doc.get("Statement", None)
        logger.info("Got default policy doc for %s.", policy.policy_name)
        logger.info(policy_doc)
    except ClientError:
        logger.exception("Couldn't get default policy statement for %s.",
 policy_arn)
        raise
```

```
    else:
        return policy_statement
```

- Per i dettagli sull'API, consulta GetPolicyVersion AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    oo_result = lo_iam->getpolicyversion(
      iv_policyarn = iv_policy_arn
      iv_versionid = iv_version_id ).
    MESSAGE 'Retrieved policy version information.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy or version does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iaminvalidinputex.
    MESSAGE 'Invalid input provided.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, GetPolicyVersionconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`GetRole`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetRole`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Configura la connettività dei servizi container
- Crea un'attività contenitore per il tipo di avvio serverless
- Creazione di un servizio contenitore per le istanze di macchine virtuali
- Nozioni di base sui data warehouse serverless
- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
- Guida introduttiva agli store di funzionalità di apprendimento automatico
- Guida introduttiva ai cluster Kubernetes gestiti
- Guida introduttiva allo streaming gestito
- Guida introduttiva ai cluster di data warehouse con provisioning
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
/// <summary>
/// Get information about an IAM role.
/// </summary>
/// <param name="roleName">The name of the IAM role to retrieve information
/// for.</param>
/// <returns>The IAM role that was retrieved.</returns>
public async Task<Role> GetRoleAsync(string roleName)
{
    var response = await _IAMService.GetRoleAsync(new GetRoleRequest
    {
        RoleName = roleName,
    });
```

```
        return response.Role;
    }
```

- Per i dettagli sull'API, GetRoleconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come ottenere informazioni su un ruolo IAM

Il comando `get-role` seguente ottiene informazioni sul ruolo denominato `Test-Role`.

```
aws iam get-role \
    --role-name Test-Role
```

Output:

```
{
    "Role": {
        "Description": "Test Role",
        "AssumeRolePolicyDocument":"<URL-encoded-JSON>",
        "MaxSessionDuration": 3600,
        "RoleId": "AROA1234567890EXAMPLE",
        "CreateDate": "2019-11-13T16:45:56Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "RoleLastUsed": {
            "Region": "us-east-1",
            "LastUsedDate": "2019-11-13T17:14:00Z"
        },
        "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

Il comando visualizza la policy di attendibilità associata al ruolo. Per elencare le policy di autorizzazioni collegate a un ruolo, usa il comando `list-role-policies`.

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta GetRole AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// GetRole gets data about a role.
func (wrapper RoleWrapper) GetRole(ctx context.Context, roleName string)
 (*types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.GetRole(ctx,
  &iam.GetRoleInput{RoleName: aws.String(roleName)})
 if err != nil {
  log.Printf("Couldn't get role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
```

```
}
```

- Per i dettagli sull'API, GetRoleconsulta AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Recupera il ruolo.

```
import { GetRoleCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 */
export const getRole = (roleName) => {
  const command = new GetRoleCommand({
    RoleName: roleName,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, GetRoleconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function getRole($roleName)
    {
        return $this->customWaiter(function () use ($roleName) {
            return $this->iamClient->getRole(['RoleName' => $roleName]);
        });
    }
```

- Per i dettagli sull'API, GetRoleconsulta AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce i dettagli di **`lamda_exec_role`**. Include il documento della policy di attendibilità, che specifica chi può assumere questo ruolo. Il documento di policy è codificato tramite URL e può essere decodificato utilizzando il metodo .NET **`UrlDecode`**. In questo esempio, la policy originale aveva rimosso tutti gli spazi bianchi prima di essere caricata nella policy. Per visualizzare i documenti relativi alle policy di autorizzazioni che determinano cosa può fare qualcuno che assume il ruolo, utilizza **`Get-IAMRolePolicy`** per le policy in linea e **`Get-IAMPolicyVersion`** per le policy gestite allegate.

```
$results = Get-IamRole -RoleName lambda_exec_role
$results | Format-List
```

Output:

```
Arn                      : arn:aws:iam::123456789012:role/lambda_exec_role
AssumeRolePolicyDocument : %7B%22Version%22%3A%222012-10-17%22%2C%22Statement
%22%3A%5B%7B%22Sid%22
                           %3A%22%22%2C%22Effect%22%3A%22Allow%22%2C%22Principal
%22%3A%7B%22Service
                           %22%3A%22lambda.amazonaws.com%22%7D%2C%22Action%22%3A
%22sts%3AAssumeRole
                           %22%7D%5D%7D
CreateDate               : 4/2/2015 9:16:11 AM
Path                     : /
RoleId                   : 2YBIKAIBHNKB4EXAMPLE1
RoleName                 : lambda_exec_role
```

```
$policy = [System.Web.HttpUtility]::UrlDecode($results.AssumeRolePolicyDocument)
$policy
```

Output:

```
{"Version":"2012-10-17",       "Statement":
[{"Sid":"","Effect":"Allow","Principal":
{"Service":"lambda.amazonaws.com"},"Action":"sts:AssumeRole"}]}
```

- Per i dettagli sull'API, vedere GetRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce i dettagli di **`lamda_exec_role`**. Include il documento della policy di attendibilità, che specifica chi può assumere questo ruolo. Il documento di policy è codificato tramite URL e può essere decodificato utilizzando il metodo .NET **`UrlDecode`**. In questo esempio, la policy originale aveva rimosso tutti gli spazi bianchi prima di essere caricata nella policy. Per visualizzare i documenti relativi alle policy di autorizzazioni che determinano cosa può fare qualcuno che assume il ruolo, utilizza **`Get-IAMRolePolicy`** per le policy in linea e **`Get-IAMPolicyVersion`** per le policy gestite allegate.

```
$results = Get-IamRole -RoleName lambda_exec_role
$results | Format-List
```

Output:

```
Arn                      : arn:aws:iam::123456789012:role/lambda_exec_role
```

```
AssumeRolePolicyDocument : %7B%22Version%22%3A%222012-10-17%22%2C%22Statement
%22%3A%5B%7B%22Sid%22
                           %3A%22%22%2C%22Effect%22%3A%22Allow%22%2C%22Principal
%22%3A%7B%22Service
                           %22%3A%22lambda.amazonaws.com%22%7D%2C%22Action%22%3A
%22sts%3AAssumeRole
                           %22%7D%5D%7D
CreateDate               : 4/2/2015 9:16:11 AM
Path                     : /
RoleId                   : 2YBIKAIBHNKB4EXAMPLE1
RoleName                 : lambda_exec_role
```

```
$policy = [System.Web.HttpUtility]::UrlDecode($results.AssumeRolePolicyDocument)
$policy
```

Output:

```
{"Version":"2012-10-17",       "Statement":
[{"Sid":"","Effect":"Allow","Principal":
{"Service":"lambda.amazonaws.com"},"Action":"sts:AssumeRole"}]}
```

- Per i dettagli sull'API, vedere GetRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def get_role(role_name):
    """
    Gets a role by name.

    :param role_name: The name of the role to retrieve.
```

```
    :return: The specified role.
    """
    try:
        role = iam.Role(role_name)
        role.load()  # calls GetRole to load attributes
        logger.info("Got role with arn %s.", role.arn)
    except ClientError:
        logger.exception("Couldn't get role named %s.", role_name)
        raise
    else:
        return role
```

- Per i dettagli sull'API, consulta GetRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
# Gets data about a role.
#
# @param name [String] The name of the role to look up.
# @return [Aws::IAM::Role] The retrieved role.
def get_role(name)
  role = @iam_client.get_role({
                                role_name: name
                              }).role
  puts("Got data for role '#{role.role_name}'. Its ARN is '#{role.arn}'.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't get data for role '#{name}' Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  role
```

```
  end
```

- Per i dettagli sull'API, GetRoleconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn get_role(
    client: &iamClient,
    role_name: String,
) -> Result<GetRoleOutput, SdkError<GetRoleError>> {
    let response = client.get_role().role_name(role_name).send().await?;
    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento GetRoleall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->getrole( iv_rolename = iv_role_name ).
```

```
        MESSAGE 'Retrieved role information.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Role does not exist.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, GetRoleconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func getRole(name: String) async throws -> IAMClientTypes.Role {
        let input = GetRoleInput(
            roleName: name
        )
        do {
            let output = try await client.getRole(input: input)
            guard let role = output.role else {
                throw ServiceHandlerError.noSuchRole
            }
            return role
        } catch {
            print("ERROR: getRole:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la GetRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetRolePolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetRolePolicy`.

CLI

AWS CLI

Per ottenere informazioni su una policy collegata a un ruolo IAM

Il comando `get-role-policy` seguente ottiene informazioni sulla policy specificata collegata al ruolo denominato `Test-Role`.

```
aws iam get-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

Output:

```
{
  "RoleName": "Test-Role",
  "PolicyDocument": {
      "Statement": [
          {
              "Action": [
                  "s3:ListBucket",
                  "s3:Put*",
                  "s3:Get*",
                  "s3:*MultipartUpload*"
              ],
              "Resource": "*",
              "Effect": "Allow",
              "Sid": "1"
          }
      ]
  }
  "PolicyName": "ExamplePolicy"
}
```

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta GetRolePolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce il documento sulla policy di autorizzazioni per la policy **`oneClick_lambda_exec_role_policy`** denominata incorporata nel ruolo IAM **`lamda_exec_role`**. Il documento di policy risultante è codificato nell'URL. In questo esempio viene decodificato con il metodo **`UrlDecode`** .NET.

```
$results = Get-IAMRolePolicy -RoleName lambda_exec_role -PolicyName
 oneClick_lambda_exec_role_policy
$results
```

Output:

```
PolicyDocument                                          PolicyName
              UserName
--------------                                          ----------
              --------
%7B%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C%...
 oneClick_lambda_exec_role_policy     lambda_exec_role
```

```
[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.PolicyDocument)
```

Output:

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:*"
      ],
      "Resource": "arn:aws:logs:us-east-1:555555555555:log-group:/aws/lambda/aws-
example-function:*"
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

- Per i dettagli sull'API, vedere GetRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce il documento sulla policy di autorizzazioni per la policy **`oneClick_lambda_exec_role_policy`** denominata incorporata nel ruolo IAM **`lamda_exec_role`**. Il documento di policy risultante è codificato nell'URL. In questo esempio viene decodificato con il metodo **`UrlDecode`** .NET.

```
$results = Get-IAMRolePolicy -RoleName lambda_exec_role -PolicyName
 oneClick_lambda_exec_role_policy
$results
```

Output:

```
PolicyDocument                                          PolicyName
              UserName
--------------                                          ----------
              --------
%7B%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C%...
 oneClick_lambda_exec_role_policy     lambda_exec_role
```

```
[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.PolicyDocument)
```

Output:

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:*"
      ],
      "Resource": "arn:aws:logs:us-east-1:555555555555:log-group:/aws/lambda/aws-
example-function:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

- Per i dettagli sull'API, vedere GetRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetSamlProvider`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetSamlProvider`.

CLI

AWS CLI

Per recuperare il metadocumento del provider SAML

Questo esempio recupera i dettagli del provider SAML 2.0 il cui ARN è `arn:aws:iam::123456789012:saml-provider/SAMLADFS`. La risposta include il documento di metadati che hai ricevuto dal provider di identità per creare l'entità del provider AWS SAML, nonché le date di creazione e scadenza.

```
aws iam get-saml-provider \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Output:

```
{
    "SAMLMetadataDocument": "...SAMLMetadataDocument-XML...",
    "CreateDate": "2017-03-06T22:29:46+00:00",
    "ValidUntil": "2117-03-06T22:29:46.433000+00:00",
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta AWS CLI Command GetSamlProviderReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera i dettagli sul provider SAML 2.0 il cui ARM è arn:aws:iam: :123456789012:saml-. provider/SAMLADFS La risposta include il documento di metadati che hai ricevuto dal provider di identità per creare l'entità del provider SAML, nonché le date di creazione e scadenza. AWS

```
Get-IAMSAMLProvider -SAMLProviderArn arn:aws:iam::123456789012:saml-provider/
SAMLADFS
```

Output:

```
CreateDate                 SAMLMetadataDocument
        ValidUntil
----------                 --------------------
        ----------
12/23/2014 12:16:55 PM    <EntityDescriptor ID="_12345678-1234-5678-9012-
example1...    12/23/2114 12:16:54 PM
```

- Per i dettagli sull'API, vedere GetSamlProviderin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera i dettagli sul provider SAML 2.0 il cui ARM è arn:aws:iam: :123456789012:saml-. provider/SAMLADFS La risposta include il documento di metadati che hai ricevuto dal provider di identità per creare l'entità del provider SAML, nonché le date di creazione e scadenza. AWS

```
Get-IAMSAMLProvider -SAMLProviderArn arn:aws:iam::123456789012:saml-provider/
SAMLADFS
```

Output:

```
CreateDate                 SAMLMetadataDocument
        ValidUntil
----------                 --------------------
        ----------
12/23/2014 12:16:55 PM    <EntityDescriptor ID="_12345678-1234-5678-9012-
example1...    12/23/2114 12:16:54 PM
```

- Per i dettagli sull'API, vedere GetSamlProviderin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`GetServerCertificate`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetServerCertificate`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::getServerCertificate(const Aws::String &certificateName,
                                       const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::GetServerCertificateRequest request;
    request.SetServerCertificateName(certificateName);

    auto outcome = iam.GetServerCertificate(request);
    bool result = true;
    if (!outcome.IsSuccess()) {
        if (outcome.GetError().GetErrorType() !=
 Aws::IAM::IAMErrors::NO_SUCH_ENTITY) {
            std::cerr << "Error getting server certificate " << certificateName
 <<
                      ": " << outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Certificate '" << certificateName
                      << "' not found." << std::endl;
        }
    }
    else {
        const auto &certificate = outcome.GetResult().GetServerCertificate();
        std::cout << "Name: " <<

 certificate.GetServerCertificateMetadata().GetServerCertificateName()
                  << std::endl << "Body: " << certificate.GetCertificateBody() <<
                  std::endl << "Chain: " << certificate.GetCertificateChain() <<
```

```
                  std::endl;
    }

    return result;
}
```

- Per i dettagli sull'API, consulta la GetServerCertificatesezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Per ottenere dettagli su un certificato server nel tuo AWS account

Il `get-server-certificate` comando seguente recupera tutti i dettagli sul certificato server specificato nel tuo AWS account.

```
aws iam get-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

Output:

```
{
    "ServerCertificate": {
        "ServerCertificateMetadata": {
            "Path": "/",
            "ServerCertificateName": "myUpdatedServerCertificate",
            "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/
myUpdatedServerCertificate",
            "UploadDate": "2019-04-22T21:13:44+00:00",
            "Expiration": "2019-10-15T22:23:16+00:00"
        },
        "CertificateBody": "-----BEGIN CERTIFICATE-----
            MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
            VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
            b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
            BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
            MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
            VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
```

```
            b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
            YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
            21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
            rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
            Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
            nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
            FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
            NYiytVbZPQUQ5Yaxu2jXnimvrszlaEXAMPLE=-----END CERTIFICATE-----",
        "CertificateChain": "-----BEGIN CERTIFICATE-----\nMIICiTCCAfICCQD6md
            7oRw0uXOjANBgkqhkiG9w0BAqQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgT
            AldBMRAwDgYDVQQHEwdTZWF0drGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAs
            TC0lBTSBDb25zb2xlMRIwEAYDVsQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQ
            jb20wHhcNMTEwNDI1MjA0NTIxWhtcNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBh
            MCVVMxCzAJBgNVBAgTAldBMRAwDgsYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBb
            WF6b24xFDASBgNVBAsTC0lBTSBDb2d5zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMx
            HzAdBgkqhkiG9w0BCQEWEG5vb25lQGfFtYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
            BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIgWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
            k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8mh9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
            ITxOUSQv7c7ugFFDzQGBzZswY6786m86gjpEIbb3OhjZnzcvQAaRHhdlQWIMm2nr
            AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUhVVxYUntneD9+h8Mg9q6q+auN
            KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FlkbFFBjvSfpJIlJ00zbhNYS5f6Guo
            EDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
            3rrszlaEWEG5vb25lQGFtsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
    }
}
```

Per elencare i certificati server disponibili nel tuo AWS account, usa il `list-server-certificates` comando.

Per ulteriori informazioni, consulta Gestione dei certificati server in IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta GetServerCertificate AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Recupera un certificato del server.

```
import { GetServerCertificateCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} certName
 * @returns
 */
export const getServerCertificate = async (certName) => {
  const command = new GetServerCertificateCommand({
    ServerCertificateName: certName,
  });

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la GetServerCertificatesezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });
```

```
iam.getServerCertificate(
  { ServerCertificateName: "CERTIFICATE_NAME" },
  function (err, data) {
    if (err) {
      console.log("Error", err);
    } else {
      console.log("Success", data);
    }
  }
);
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la GetServerCertificatesezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera i dettagli del certificato server denominato **MyServerCertificate**. È possibile trovare i dettagli del certificato nelle proprietà **CertificateBody** e **ServerCertificateMetadata**.

```
$result = Get-IAMServerCertificate -ServerCertificateName MyServerCertificate
$result | format-list
```

Output:

```
CertificateBody          : -----BEGIN CERTIFICATE-----

 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC

 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6

 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd

 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN

 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
```

```
 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z

 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt

 YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
                            21uUSfwfEvySWtC2XADZ4nB
+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
                            rDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE

 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
                            nUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb

 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
                            NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
                            -----END CERTIFICATE-----
CertificateChain          :
ServerCertificateMetadata :
 Amazon.IdentityManagement.Model.ServerCertificateMetadata
```

```
$result.ServerCertificateMetadata
```

Output:

```
Arn                   : arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyServerCertificate
Expiration            : 1/14/2018 9:52:36 AM
Path                  : /Org1/Org2/
ServerCertificateId   : ASCAJIFEXAMPLE17HQZYW
ServerCertificateName : MyServerCertificate
UploadDate            : 4/21/2015 11:14:16 AM
```

- Per i dettagli sull'API, vedere GetServerCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera i dettagli del certificato server denominato **`MyServerCertificate`**. È possibile trovare i dettagli del certificato nelle proprietà **`CertificateBody`** e **`ServerCertificateMetadata`**.

```
$result = Get-IAMServerCertificate -ServerCertificateName MyServerCertificate
$result | format-list
```

Output:

```
CertificateBody          : -----BEGIN CERTIFICATE-----

 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC

 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6

 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd

 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN

 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD

 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z

 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt

 YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
                           21uUSfwfEvySWtC2XADZ4nB
+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
                           rDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE

 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
                           nUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb

 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
                           NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
                           -----END CERTIFICATE-----
CertificateChain         :
ServerCertificateMetadata :
 Amazon.IdentityManagement.Model.ServerCertificateMetadata
```

```
$result.ServerCertificateMetadata
```

Output:

```
Arn                   : arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyServerCertificate
Expiration            : 1/14/2018 9:52:36 AM
Path                  : /Org1/Org2/
ServerCertificateId   : ASCAJIFEXAMPLE17HQZYW
ServerCertificateName : MyServerCertificate
UploadDate            : 4/21/2015 11:14:16 AM
```

- Per i dettagli sull'API, vedere GetServerCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetServiceLastAccessedDetails`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetServiceLastAccessedDetails`.

CLI

AWS CLI

Per recuperare un report di accesso al servizio

L'esempio `get-service-last-accessed-details` seguente recupera un report generato in precedenza che elenca i servizi a cui accedono le entità IAM. Per generare un report, utilizza il `generate-service-last-accessed-details` comando.

```
aws iam get-service-last-accessed-details \
    --job-id 2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc
```

Output:

```
{
    "JobStatus": "COMPLETED",
    "JobCreationDate": "2019-10-01T03:50:35.929Z",
    "ServicesLastAccessed": [
        ...
        {
```

```
            "ServiceName": "AWS Lambda",
            "LastAuthenticated": "2019-09-30T23:02:00Z",
            "ServiceNamespace": "lambda",
            "LastAuthenticatedEntity": "arn:aws:iam::123456789012:user/admin",
            "TotalAuthenticatedEntities": 6
        },
    ]
}
```

Per ulteriori informazioni, consulta Ridefinizione delle autorizzazioni nell' AWS utilizzo delle ultime informazioni a cui si accede nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta AWS CLI Command GetServiceLastAccessedDetailsReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio fornisce i dettagli dell'ultimo accesso al servizio da parte dell'entità IAM (utente, gruppo, ruolo o policy) associata alla chiamata Request.

```
Request-IAMServiceLastAccessedDetail -Arn arn:aws:iam::123456789012:user/TestUser
```

Output:

```
f0b7a819-eab0-929b-dc26-ca598911cb9f
```

```
Get-IAMServiceLastAccessedDetail -JobId f0b7a819-eab0-929b-dc26-ca598911cb9f
```

- Per i dettagli sull'API, vedere GetServiceLastAccessedDetailsin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio fornisce i dettagli dell'ultimo accesso al servizio da parte dell'entità IAM (utente, gruppo, ruolo o policy) associata alla chiamata Request.

```
Request-IAMServiceLastAccessedDetail -Arn arn:aws:iam::123456789012:user/TestUser
```

Output:

```
f0b7a819-eab0-929b-dc26-ca598911cb9f
```

```
Get-IAMServiceLastAccessedDetail -JobId f0b7a819-eab0-929b-dc26-ca598911cb9f
```

- Per i dettagli sull'API, vedere GetServiceLastAccessedDetailsin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetServiceLastAccessedDetailsWithEntities`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetServiceLastAccessedDetailsWithEntities`.

CLI

AWS CLI

Per recuperare un report di accesso al servizio con i dettagli relativi a un servizio

L'esempio `get-service-last-accessed-details-with-entities` seguente recupera un report che contiene dettagli degli utenti IAM e altre entità che hanno avuto accesso al servizio specificato. Per generare un report, utilizza il `generate-service-last-accessed-details` comando. Per ottenere un elenco di servizi a cui si accede con gli spazi dei nomi, usa `get-service-last-accessed-details`.

```
aws iam get-service-last-accessed-details-with-entities \
    --job-id 78b6c2ba-d09e-6xmp-7039-ecde30b26916 \
    --service-namespace lambda
```

Output:

```
{
    "JobStatus": "COMPLETED",
```

```
    "JobCreationDate": "2019-10-01T03:55:41.756Z",
    "JobCompletionDate": "2019-10-01T03:55:42.533Z",
    "EntityDetailsList": [
        {
            "EntityInfo": {
                "Arn": "arn:aws:iam::123456789012:user/admin",
                "Name": "admin",
                "Type": "USER",
                "Id": "AIDAIO2XMPLENQEXAMPLE",
                "Path": "/"
            },
            "LastAuthenticated": "2019-09-30T23:02:00Z"
        },
        {
            "EntityInfo": {
                "Arn": "arn:aws:iam::123456789012:user/developer",
                "Name": "developer",
                "Type": "USER",
                "Id": "AIDAIBEYXMPL2YEXAMPLE",
                "Path": "/"
            },
            "LastAuthenticated": "2019-09-16T19:34:00Z"
        }
    ]
}
```

Per ulteriori informazioni, consulta Ridefinizione delle autorizzazioni nell' AWS utilizzo delle ultime informazioni a cui si accede nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta AWS CLI Command GetServiceLastAccessedDetailsWithEntitiesReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio fornisce il timestamp dell'ultimo accesso per il servizio contenuto nella richiesta della rispettiva entità IAM.

```
$results = Get-IAMServiceLastAccessedDetailWithEntity -JobId f0b7a819-eab0-929b-
dc26-ca598911cb9f -ServiceNamespace ec2
$results
```

Output:

```
EntityDetailsList : {Amazon.IdentityManagement.Model.EntityDetails}
Error             :
IsTruncated       : False
JobCompletionDate : 12/29/19 11:19:31 AM
JobCreationDate   : 12/29/19 11:19:31 AM
JobStatus         : COMPLETED
Marker            :
```

```
$results.EntityDetailsList
```

Output:

```
EntityInfo                                  LastAuthenticated
----------                                  -----------------
Amazon.IdentityManagement.Model.EntityInfo 11/16/19 3:47:00 PM
```

```
$results.EntityInfo
```

Output:

```
Arn  : arn:aws:iam::123456789012:user/TestUser
Id   : AIDA4NBK5CXF5TZHU1234
Name : TestUser
Path : /
Type : USER
```

- Per i dettagli sull'API, vedere GetServiceLastAccessedDetailsWithEntitiesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio fornisce il timestamp dell'ultimo accesso per il servizio contenuto nella richiesta della rispettiva entità IAM.

```
$results = Get-IAMServiceLastAccessedDetailWithEntity -JobId f0b7a819-eab0-929b-
dc26-ca598911cb9f -ServiceNamespace ec2
$results
```

Output:

```
EntityDetailsList : {Amazon.IdentityManagement.Model.EntityDetails}
Error             :
IsTruncated       : False
JobCompletionDate : 12/29/19 11:19:31 AM
JobCreationDate   : 12/29/19 11:19:31 AM
JobStatus         : COMPLETED
Marker            :
```

```
$results.EntityDetailsList
```

Output:

```
EntityInfo                                  LastAuthenticated
----------                                  -----------------
Amazon.IdentityManagement.Model.EntityInfo 11/16/19 3:47:00 PM
```

```
$results.EntityInfo
```

Output:

```
Arn  : arn:aws:iam::123456789012:user/TestUser
Id   : AIDA4NBK5CXF5TZHU1234
Name : TestUser
Path : /
Type : USER
```

- Per i dettagli sull'API, vedere GetServiceLastAccessedDetailsWithEntitiesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`GetServiceLinkedRoleDeletionStatus`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetServiceLinkedRoleDeletionStatus`.

CLI

AWS CLI

Come verificare lo stato di una richiesta di eliminazione di un ruolo collegato a un servizio

L'esempio `get-service-linked-role-deletion-status` seguente visualizza lo stato di una precedente richiesta di eliminazione di un ruolo collegato a un servizio. L'operazione di eliminazione avviene in modo asincrono. Quando effettui la richiesta, ottieni un valore `DeletionTaskId` che hai fornito come parametro per questo comando.

```
aws iam get-service-linked-role-deletion-status \
    --deletion-task-id task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE
```

Output:

```
{
"Status": "SUCCEEDED"
}
```

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati a servizi nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetServiceLinkedRoleDeletionStatus AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import {
  GetServiceLinkedRoleDeletionStatusCommand,
```

```
  IAMClient,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} deletionTaskId
 */
export const getServiceLinkedRoleDeletionStatus = (deletionTaskId) => {
  const command = new GetServiceLinkedRoleDeletionStatusCommand({
    DeletionTaskId: deletionTaskId,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, GetServiceLinkedRoleDeletionStatusconsulta AWS SDK per JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`GetUser`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetUser`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Get information about an IAM user.
```

```
    /// </summary>
    /// <param name="userName">The username of the user.</param>
    /// <returns>An IAM user object.</returns>
    public async Task<User> GetUserAsync(string userName)
    {
        var response = await _IAMService.GetUserAsync(new GetUserRequest
 { UserName = userName });
        return response.User;
    }
```

- Per i dettagli sull'API, consulta la GetUsersezione AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
```

```
# Returns:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
###############################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}

  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}
```

- Per i dettagli sull'API, consulta GetUser AWS CLICommand Reference.

CLI

AWS CLI

Come ottenere informazioni su un utente IAM

Il comando `get-user` seguente ottiene informazioni sull'utente IAM denominato `Paulo`.

```
aws iam get-user \
    --user-name Paulo
```

Output:

```
{
    "User": {
        "UserName": "Paulo",
        "Path": "/",
        "CreateDate": "2019-09-21T23:03:13Z",
        "UserId": "AIDA123456789EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Paulo"
    }
}
```

Per ulteriori informazioni, consulta Gestione di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta GetUser AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
```

```
 IamClient *iam.Client
}



// GetUser gets data about a user.
func (wrapper UserWrapper) GetUser(ctx context.Context, userName string)
 (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.GetUser(ctx, &iam.GetUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  var apiError smithy.APIError
  if errors.As(err, &apiError) {
   switch apiError.(type) {
   case *types.NoSuchEntityException:
    log.Printf("User %v does not exist.\n", userName)
    err = nil
   default:
    log.Printf("Couldn't get user %v. Here's why: %v\n", userName, err)
   }
  }
 } else {
  user = result.User
 }
 return user, err
}
```

- Per i dettagli sull'API, consulta la GetUsersezione AWS SDK per Go API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera i dettagli dell'utente denominato **`David`**.

```
Get-IAMUser -UserName David
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/David
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 3/19/2015 8:44:04 AM
Path             : /
UserId           : Y4FKWQCXTA52QEXAMPLE1
UserName         : David
```

Esempio 2: questo esempio recupera i dettagli dell'utente IAM correntemente connesso.

```
Get-IAMUser
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Bob
CreateDate       : 10/16/2014 9:03:09 AM
PasswordLastUsed : 3/4/2015 12:12:33 PM
Path             : /
UserId           : 7K3GJEANSKZF2EXAMPLE2
UserName         : Bob
```

- Per i dettagli sull'API, vedere GetUserin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera i dettagli dell'utente denominato **`David`**.

```
Get-IAMUser -UserName David
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/David
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 3/19/2015 8:44:04 AM
Path             : /
UserId           : Y4FKWQCXTA52QEXAMPLE1
UserName         : David
```

Esempio 2: questo esempio recupera i dettagli dell'utente IAM correntemente connesso.

```
Get-IAMUser
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Bob
CreateDate       : 10/16/2014 9:03:09 AM
PasswordLastUsed : 3/4/2015 12:12:33 PM
Path             : /
UserId           : 7K3GJEANSKZF2EXAMPLE2
UserName         : Bob
```

- Per i dettagli sull'API, vedere GetUserin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
 # Retrieves a user's details
 #
 # @param user_name [String] The name of the user to retrieve
 # @return [Aws::IAM::Types::User, nil] The user object if found, or nil if an
error occurred
 def get_user(user_name)
   response = @iam_client.get_user(user_name: user_name)
   response.user
 rescue Aws::IAM::Errors::NoSuchEntity
   @logger.error("User '#{user_name}' not found.")
   nil
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error retrieving user '#{user_name}': #{e.message}")
   nil
 end
```

- Per i dettagli sull'API, consulta la GetUsersezione AWS SDK per Ruby API Reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func getUser(name: String? = nil) async throws -> IAMClientTypes.User
 {
        let input = GetUserInput(
            userName: name
        )
        do {
            let output = try await iamClient.getUser(input: input)
            guard let user = output.user else {
                throw ServiceHandlerError.noSuchUser
            }
            return user
        } catch {
            print("ERROR: getUser:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la GetUserguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetUserPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetUserPolicy`.

CLI

AWS CLI

Per visualizzare i dettagli della policy per un utente IAM

Il comando `get-user-policy` seguente riporta i dettagli della policy specificata collegata all'utente IAM denominato Bob.

```
aws iam get-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

Output:

```
{
    "UserName": "Bob",
    "PolicyName": "ExamplePolicy",
    "PolicyDocument": {
        "Version":"2012-10-17",
        "Statement": [
            {
                "Action": "*",
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    }
}
```

Per ottenere un elenco di policy per un utente IAM, usa il comando `list-user-policies`.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetUserPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera i dettagli della policy in linea denominata **`Davids_IAM_Admin_Policy`** che è incorporata nell'utente IAM denominato **`David`**. Il documento di policy è codificato nell'URL.

```
$results = Get-IAMUserPolicy -PolicyName Davids_IAM_Admin_Policy -UserName David
$results
```

Output:

```
PolicyDocument                                          PolicyName
       UserName
--------------                                          ----------
       --------
%7B%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C%...    Davids_IAM_Admin_Policy
       David

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.PolicyDocument)
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetUser",
        "iam:ListUsers"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:user/*"
      ]
    }
  ]
}
```

- Per i dettagli sull'API, vedere GetUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera i dettagli della policy in linea denominata **`Davids_IAM_Admin_Policy`** che è incorporata nell'utente IAM denominato **`David`**. Il documento di policy è codificato nell'URL.

```
$results = Get-IAMUserPolicy -PolicyName Davids_IAM_Admin_Policy -UserName David
$results
```

Output:

```
PolicyDocument                                                  PolicyName
       UserName
--------------                                                  ----------
       --------
%7B%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C%...    Davids_IAM_Admin_Policy
       David

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.PolicyDocument)
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetUser",
        "iam:ListUsers"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:user/*"
      ]
    }
  ]
}
```

- Per i dettagli sull'API, vedere GetUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`ListAccessKeys`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListAccessKeys`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestione delle chiavi di accesso

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function iam_list_access_keys
#
# This function lists the access keys for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#
# Returns:
```

```
#       access_key_ids
#     And:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function iam_list_access_keys() {

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_list_access_keys"
    echo "Lists the AWS Identity and Access Management (IAM) access key IDs for
 the specified user."
    echo "  -u user_name   The name of the IAM user."
    echo ""
  }

  local user_name response
  local option OPTARG # Required to use getopts command in a function.
  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam list-access-keys \
    --user-name "$user_name" \
    --output text \
```

```
    --query 'AccessKeyMetadata[].AccessKeyId')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-access-keys operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta ListAccessKeys AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::listAccessKeys(const Aws::String &userName,
                                 const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListAccessKeysRequest request;
    request.SetUserName(userName);

    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListAccessKeys(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list access keys for user " << userName
                      << ": " << outcome.GetError().GetMessage() << std::endl;
```

```
            return false;
        }

        if (!header) {
            std::cout << std::left << std::setw(32) << "UserName" <<
                      std::setw(30) << "KeyID" << std::setw(20) << "Status" <<
                      std::setw(20) << "CreateDate" << std::endl;
            header = true;
        }

        const auto &keys = outcome.GetResult().GetAccessKeyMetadata();
        const Aws::String DATE_FORMAT = "%Y-%m-%d";

        for (const auto &key: keys) {
            Aws::String statusString =
                    Aws::IAM::Model::StatusTypeMapper::GetNameForStatusType(
                            key.GetStatus());
            std::cout << std::left << std::setw(32) << key.GetUserName() <<
                      std::setw(30) << key.GetAccessKeyId() << std::setw(20) <<
                      statusString << std::setw(20) <<
                      key.GetCreateDate().ToGmtString(DATE_FORMAT.c_str()) <<
 std::endl;
        }

        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, consulta la ListAccessKeyssezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come elencare gli ID delle chiavi di accesso per un utente IAM

Il comando `list-access-keys` seguente elenca gli ID delle chiavi di accesso per l'utente IAM denominato Bob.

```
aws iam list-access-keys \
    --user-name Bob
```

Output:

```
{
    "AccessKeyMetadata": [
        {
            "UserName": "Bob",
            "Status": "Active",
            "CreateDate": "2013-06-04T18:17:34Z",
            "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
        },
        {
            "UserName": "Bob",
            "Status": "Inactive",
            "CreateDate": "2013-06-06T20:42:26Z",
            "AccessKeyId": "AKIAI44QH8DHBEXAMPLE"
        }
    ]
}
```

Non puoi elencare le chiavi di accesso segrete per gli utenti IAM. Se le chiavi di accesso segrete vengono perse, devi creare nuove chiavi di accesso utilizzando il comando `create-access-keys`.

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListAccessKeys AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListAccessKeys lists the access keys for the specified user.
func (wrapper UserWrapper) ListAccessKeys(ctx context.Context, userName string)
 ([]types.AccessKeyMetadata, error) {
 var keys []types.AccessKeyMetadata
 result, err := wrapper.IamClient.ListAccessKeys(ctx, &iam.ListAccessKeysInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list access keys for user %v. Here's why: %v\n", userName,
 err)
 } else {
```

```
  keys = result.AccessKeyMetadata
 }
 return keys, err
}
```

- Per i dettagli sull'API, consulta la ListAccessKeyssezione AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.AccessKeyMetadata;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.ListAccessKeysRequest;
import software.amazon.awssdk.services.iam.model.ListAccessKeysResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListAccessKeys {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <userName>\s
```

```
                Where:
                    userName - The name of the user for which access keys are
retrieved.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String userName = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        listKeys(iam, userName);
        System.out.println("Done");
        iam.close();
    }

    public static void listKeys(IamClient iam, String userName) {
        try {
            boolean done = false;
            String newMarker = null;

            while (!done) {
                ListAccessKeysResponse response;

                if (newMarker == null) {
                    ListAccessKeysRequest request =
ListAccessKeysRequest.builder()
                            .userName(userName)
                            .build();

                    response = iam.listAccessKeys(request);

                } else {
                    ListAccessKeysRequest request =
ListAccessKeysRequest.builder()
                            .userName(userName)
                            .marker(newMarker)
                            .build();
```

```
                    response = iam.listAccessKeys(request);
                }

                for (AccessKeyMetadata metadata : response.accessKeyMetadata()) {
                    System.out.format("Retrieved access key %s",
 metadata.accessKeyId());
                }

                if (!response.isTruncated()) {
                    done = true;
                } else {
                    newMarker = response.marker();
                }
            }

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListAccessKeyssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca le chiavi di accesso.

```
import { ListAccessKeysCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});
```

```
/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 * @param {string} userName
 */
export async function* listAccessKeys(userName) {
  const command = new ListAccessKeysCommand({
    MaxItems: 5,
    UserName: userName,
  });

  /**
   * @type {import("@aws-sdk/client-iam").ListAccessKeysCommandOutput |
 undefined}
   */
  let response = await client.send(command);

  while (response?.AccessKeyMetadata?.length) {
    for (const key of response.AccessKeyMetadata) {
      yield key;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListAccessKeysCommand({
          Marker: response.Marker,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la ListAccessKeyssezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  MaxItems: 5,
  UserName: "IAM_USER_NAME",
};

iam.listAccessKeys(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la ListAccessKeyssezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun listKeys(userNameVal: String?) {
    val request =
        ListAccessKeysRequest {
            userName = userNameVal
        }
    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listAccessKeys(request)
        response.accessKeyMetadata?.forEach { md ->
            println("Retrieved access key ${md.accessKeyId}")
        }
    }
}
```

- Per i dettagli sull'API, ListAccessKeysconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: questo comando elenca le chiavi di accesso per l'utente IAM denominato **Bob**. Tenere presente che non è possibile elencare le chiavi di accesso segrete per gli utenti IAM. Se le chiavi di accesso segrete vengono perse, sarà necessario creare nuove chiavi di accesso con il cmdlet **New-IAMAccessKey**.

```
Get-IAMAccessKey -UserName "Bob"
```

Output:

```
AccessKeyId               CreateDate                 Status
 UserName
-----------               ----------                 ------
 --------
AKIAIOSFODNN7EXAMPLE      12/3/2014 10:53:41 AM      Active             Bob
AKIAI44QH8DHBEXAMPLE      6/6/2013 8:42:26 PM        Inactive           Bob
```

- Per i dettagli sull'API, vedere ListAccessKeysin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando elenca le chiavi di accesso per l'utente IAM denominato **Bob**. Tenere presente che non è possibile elencare le chiavi di accesso segrete per gli utenti IAM. Se le chiavi di accesso segrete vengono perse, sarà necessario creare nuove chiavi di accesso con il cmdlet **New-IAMAccessKey**.

```
Get-IAMAccessKey -UserName "Bob"
```

Output:

```
AccessKeyId               CreateDate                 Status
 UserName
-----------               ----------                 ------
 --------
AKIAIOSFODNN7EXAMPLE      12/3/2014 10:53:41 AM      Active             Bob
AKIAI44QH8DHBEXAMPLE      6/6/2013 8:42:26 PM        Inactive           Bob
```

- Per i dettagli sull'API, vedere ListAccessKeysin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def list_keys(user_name):
    """
    Lists the keys owned by the specified user.

    :param user_name: The name of the user.
    :return: The list of keys owned by the user.
    """
    try:
        keys = list(iam.User(user_name).access_keys.all())
        logger.info("Got %s access keys for %s.", len(keys), user_name)
    except ClientError:
        logger.exception("Couldn't get access keys for %s.", user_name)
        raise
    else:
        return keys
```

- Per i dettagli sull'API, consulta ListAccessKeys AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, disattiva ed elimina le chiavi di accesso.

```
# Manages access keys for IAM users
class AccessKeyManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'AccessKeyManager'
  end
```

```
  # Lists access keys for a user
  #
  # @param user_name [String] The name of the user.
  def list_access_keys(user_name)
    response = @iam_client.list_access_keys(user_name: user_name)
    if response.access_key_metadata.empty?
      @logger.info("No access keys found for user '#{user_name}'.")
    else
      response.access_key_metadata.map(&:access_key_id)
    end
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Error listing access keys: cannot find user '#{user_name}'.")
    []
  rescue StandardError => e
    @logger.error("Error listing access keys: #{e.message}")
    []
  end

  # Creates an access key for a user
  #
  # @param user_name [String] The name of the user.
  # @return [Boolean]
  def create_access_key(user_name)
    response = @iam_client.create_access_key(user_name: user_name)
    access_key = response.access_key
    @logger.info("Access key created for user '#{user_name}':
#{access_key.access_key_id}")
    access_key
  rescue Aws::IAM::Errors::LimitExceeded
    @logger.error('Error creating access key: limit exceeded. Cannot create
more.')
    nil
  rescue StandardError => e
    @logger.error("Error creating access key: #{e.message}")
    nil
  end

  # Deactivates an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def deactivate_access_key(user_name, access_key_id)
    @iam_client.update_access_key(
```

```
      user_name: user_name,
      access_key_id: access_key_id,
      status: 'Inactive'
    )
    true
  rescue StandardError => e
    @logger.error("Error deactivating access key: #{e.message}")
    false
  end

  # Deletes an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def delete_access_key(user_name, access_key_id)
    @iam_client.delete_access_key(
      user_name: user_name,
      access_key_id: access_key_id
    )
    true
  rescue StandardError => e
    @logger.error("Error deleting access key: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta la ListAccessKeyssezione AWS SDK per Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listaccesskeys(
          iv_username = iv_user_name ).
        MESSAGE 'Retrieved access key list.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'User does not exist.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, ListAccessKeysconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`ListAccountAliases`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListAccountAliases`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool
AwsDoc::IAM::listAccountAliases(const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListAccountAliasesRequest request;
```

```
    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListAccountAliases(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list account aliases: " <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        const auto &aliases = outcome.GetResult().GetAccountAliases();
        if (!header) {
            if (aliases.size() == 0) {
                std::cout << "Account has no aliases" << std::endl;
                break;
            }
            std::cout << std::left << std::setw(32) << "Alias" << std::endl;
            header = true;
        }

        for (const auto &alias: aliases) {
            std::cout << std::left << std::setw(32) << alias << std::endl;
        }

        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, ListAccountAliasesconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Elencare gli alias di un account

Il comando `list-account-aliases` seguente elenca gli alias per l'account corrente.

```
aws iam list-account-aliases
```

Output:

```
{
    "AccountAliases": [
    "mycompany"
    ]
}
```

Per maggiori informazioni, consulta l'ID AWS del tuo account e il suo alias nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta ListAccountAliases AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.ListAccountAliasesResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class ListAccountAliases {
    public static void main(String[] args) {
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        listAliases(iam);
        System.out.println("Done");
        iam.close();
    }

    public static void listAliases(IamClient iam) {
        try {
            ListAccountAliasesResponse response = iam.listAccountAliases();
            for (String alias : response.accountAliases()) {
                System.out.printf("Retrieved account alias %s", alias);
            }

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, ListAccountAliasesconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca gli alias di un account.

```
import { ListAccountAliasesCommand, IAMClient } from "@aws-sdk/client-iam";
```

```
const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 */
export async function* listAccountAliases() {
  const command = new ListAccountAliasesCommand({ MaxItems: 5 });

  let response = await client.send(command);

  while (response.AccountAliases?.length) {
    for (const alias of response.AccountAliases) {
      yield alias;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListAccountAliasesCommand({
          Marker: response.Marker,
          MaxItems: 5,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, ListAccountAliasesconsulta AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.listAccountAliases({ MaxItems: 10 }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, ListAccountAliasesconsulta AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun listAliases() {
    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listAccountAliases(ListAccountAliasesRequest {})
        response.accountAliases?.forEach { alias ->
            println("Retrieved account alias $alias")
        }
    }
}
```

- Per i dettagli sull'API, ListAccountAliasesconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: questo comando restituisce l'alias dell'account per l' Account AWS.

```
Get-IAMAccountAlias
```

Output:

```
ExampleCo
```

- Per i dettagli sull'API, vedere ListAccountAliasesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando restituisce l'alias dell'account per l' Account AWS.

```
Get-IAMAccountAlias
```

Output:

```
ExampleCo
```

- Per i dettagli sull'API, vedere ListAccountAliasesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def list_aliases():
    """
    Gets the list of aliases for the current account. An account has at most one
 alias.

    :return: The list of aliases for the account.
    """
    try:
        response = iam.meta.client.list_account_aliases()
        aliases = response["AccountAliases"]
        if len(aliases) > 0:
            logger.info("Got aliases for your account: %s.", ",".join(aliases))
        else:
            logger.info("Got no aliases for your account.")
    except ClientError:
        logger.exception("Couldn't list aliases for your account.")
        raise
    else:
        return response["AccountAliases"]
```

- Per i dettagli sull'API, consulta ListAccountAliases AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca, crea ed elimina gli alias degli account.

```
class IAMAliasManager
  # Initializes the IAM client and logger
  #
```

```
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists available AWS account aliases.
  def list_aliases
    response = @iam_client.list_account_aliases

    if response.account_aliases.count.positive?
      @logger.info('Account aliases are:')
      response.account_aliases.each { |account_alias| @logger.info("
#{account_alias}") }
    else
      @logger.info('No account aliases found.')
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing account aliases: #{e.message}")
  end

  # Creates an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to create.
  # @return [Boolean] true if the account alias was created; otherwise, false.
  def create_account_alias(account_alias)
    @iam_client.create_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating account alias: #{e.message}")
    false
  end

  # Deletes an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to delete.
  # @return [Boolean] true if the account alias was deleted; otherwise, false.
  def delete_account_alias(account_alias)
    @iam_client.delete_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting account alias: #{e.message}")
    false
  end
```

```
end
```

- Per i dettagli sull'API, ListAccountAliasesconsulta AWS SDK per Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    oo_result = lo_iam->listaccountaliases( ).
    MESSAGE 'Retrieved account alias list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing account aliases.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, ListAccountAliasesconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListAttachedGroupPolicies`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListAttachedGroupPolicies`.

CLI

AWS CLI

Per visualizzare tutte le policy gestite collegate al gruppo specificato

Questo esempio restituisce i nomi e gli ARN delle policy gestite allegate al gruppo IAM denominato `Admins` nell' AWS account.

```
aws iam list-attached-group-policies \
    --group-name Admins
```

Output:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListAttachedGroupPolicies AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo comando restituisce i nomi e gli ARN delle politiche gestite allegate al gruppo IAM denominato **`Admins`** nell' AWS account. Per visualizzare l'elenco delle policy in linea incorporate nel gruppo, usa il comando **`Get-IAMGroupPolicyList`**.

```
Get-IAMAttachedGroupPolicyList -GroupName "Admins"
```

Output:

```
PolicyArn                                                PolicyName
---------                                                ----------
```

```
arn:aws:iam::aws:policy/SecurityAudit                    SecurityAudit
arn:aws:iam::aws:policy/AdministratorAccess              AdministratorAccess
```

- Per i dettagli sull'API, vedere ListAttachedGroupPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando restituisce i nomi e gli ARN delle politiche gestite allegate al gruppo IAM denominato **`Admins`** nell' AWS account. Per visualizzare l'elenco delle policy in linea incorporate nel gruppo, usa il comando **`Get-IAMGroupPolicyList`**.

```
Get-IAMAttachedGroupPolicyList -GroupName "Admins"
```

Output:

```
PolicyArn                                                PolicyName
---------                                                ----------
arn:aws:iam::aws:policy/SecurityAudit                    SecurityAudit
arn:aws:iam::aws:policy/AdministratorAccess              AdministratorAccess
```

- Per i dettagli sull'API, vedere ListAttachedGroupPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`ListAttachedRolePolicies`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListAttachedRolePolicies`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// List the IAM role policies that are attached to an IAM role.
    /// </summary>
    /// <param name="roleName">The IAM role to list IAM policies for.</param>
    /// <returns>A list of the IAM policies attached to the IAM role.</returns>
    public async Task<List<AttachedPolicyType>>
ListAttachedRolePoliciesAsync(string roleName)
    {
        var attachedPolicies = new List<AttachedPolicyType>();
        var attachedRolePoliciesPaginator =
_IAMService.Paginators.ListAttachedRolePolicies(new
ListAttachedRolePoliciesRequest { RoleName = roleName });

        await foreach (var response in attachedRolePoliciesPaginator.Responses)
        {
            attachedPolicies.AddRange(response.AttachedPolicies);
        }

        return attachedPolicies;
    }
```

- Per i dettagli sull'API, ListAttachedRolePoliciesconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come elencare tutte le policy gestite collegate al ruolo specificato

Questo comando restituisce i nomi e gli ARN delle politiche gestite allegate al ruolo IAM denominato `SecurityAuditRole` nell' AWS account.

```
aws iam list-attached-role-policies \
    --role-name SecurityAuditRole
```

Output:

```
{
    "AttachedPolicies": [
```

```
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListAttachedRolePolicies AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}
```

```
// ListAttachedRolePolicies lists the policies that are attached to the specified
 role.
func (wrapper RoleWrapper) ListAttachedRolePolicies(ctx context.Context, roleName
 string) ([]types.AttachedPolicy, error) {
 var policies []types.AttachedPolicy
 result, err := wrapper.IamClient.ListAttachedRolePolicies(ctx,
 &iam.ListAttachedRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list attached policies for role %v. Here's why: %v\n",
 roleName, err)
 } else {
  policies = result.AttachedPolicies
 }
 return policies, err
}
```

- Per i dettagli sull'API, ListAttachedRolePoliciesconsulta AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca le policy collegate a un ruolo.

```
import {
  ListAttachedRolePoliciesCommand,
  IAMClient,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});
```

```
/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 * @param {string} roleName
 */
export async function* listAttachedRolePolicies(roleName) {
  const command = new ListAttachedRolePoliciesCommand({
    RoleName: roleName,
  });

  let response = await client.send(command);

  while (response.AttachedPolicies?.length) {
    for (const policy of response.AttachedPolicies) {
      yield policy;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListAttachedRolePoliciesCommand({
          RoleName: roleName,
          Marker: response.Marker,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per i dettagli sull'API, ListAttachedRolePoliciesconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listAttachedRolePolicies($roleName, $pathPrefix = "", $marker
 = "", $maxItems = 0)
    {
        $listAttachRolePoliciesArguments = ['RoleName' => $roleName];
        if ($pathPrefix) {
            $listAttachRolePoliciesArguments['PathPrefix'] = $pathPrefix;
        }
        if ($marker) {
            $listAttachRolePoliciesArguments['Marker'] = $marker;
        }
        if ($maxItems) {
            $listAttachRolePoliciesArguments['MaxItems'] = $maxItems;
        }
        return $this->iamClient-
>listAttachedRolePolicies($listAttachRolePoliciesArguments);
    }
```

- Per i dettagli sull'API, ListAttachedRolePoliciesconsulta AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo comando restituisce i nomi e gli ARN delle policy gestite collegate al ruolo IAM denominato **`SecurityAuditRole`** nell'account AWS . Per visualizzare l'elenco delle policy in linea incorporate nel ruolo, usa il comando **`Get-IAMRolePolicyList`**.

```
Get-IAMAttachedRolePolicyList -RoleName "SecurityAuditRole"
```

Output:

```
PolicyArn                                                 PolicyName
---------                                                 ----------
arn:aws:iam::aws:policy/SecurityAudit                     SecurityAudit
```

- Per i dettagli sull'API, vedere ListAttachedRolePoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando restituisce i nomi e gli ARN delle policy gestite collegate al ruolo IAM denominato **`SecurityAuditRole`** nell'account AWS . Per visualizzare l'elenco delle policy in linea incorporate nel ruolo, usa il comando **`Get-IAMRolePolicyList`**.

```
Get-IAMAttachedRolePolicyList -RoleName "SecurityAuditRole"
```

Output:

```
PolicyArn                                                 PolicyName
---------                                                 ----------
arn:aws:iam::aws:policy/SecurityAudit                     SecurityAudit
```

- Per i dettagli sull'API, vedere ListAttachedRolePoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def list_attached_policies(role_name):
    """
```

```
    Lists policies attached to a role.

    :param role_name: The name of the role to query.
    """
    try:
        role = iam.Role(role_name)
        for policy in role.attached_policies.all():
            logger.info("Got policy %s.", policy.arn)
    except ClientError:
        logger.exception("Couldn't list attached policies for %s.", role_name)
        raise
```

- Per i dettagli sull'API, consulta ListAttachedRolePolicies AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, collega e scollega le policy relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'PolicyManager'
  end

  # Creates a policy
```

```
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end

  # Fetches an IAM policy by its ARN
  # @param policy_arn [String] the ARN of the IAM policy to retrieve
  # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
  def get_policy(policy_arn)
    response = @iam_client.get_policy(policy_arn: policy_arn)
    policy = response.policy
    @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
    policy
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
    raise
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
    raise
  end

  # Attaches a policy to a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_role(role_name, policy_arn)
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
```

```
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, ListAttachedRolePoliciesconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn list_attached_role_policies(
    client: &iamClient,
    role_name: String,
    path_prefix: Option<String>,
    marker: Option<String>,
    max_items: Option<i32>,
) -> Result<ListAttachedRolePoliciesOutput,
 SdkError<ListAttachedRolePoliciesError>> {
    let response = client
        .list_attached_role_policies()
        .role_name(role_name)
        .set_path_prefix(path_prefix)
        .set_marker(marker)
        .set_max_items(max_items)
        .send()
        .await?;

    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento ListAttachedRolePoliciesall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listattachedrolepolicies(
          iv_rolename = iv_role_name ).
        MESSAGE 'Retrieved attached policy list for role.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Role does not exist.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, ListAttachedRolePoliciesconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3



    /// Returns a list of AWS Identity and Access Management (IAM) policies
    /// that are attached to the role.
```

```
    ///
    /// - Parameter role: The IAM role to return the policy list for.
    ///
    /// - Returns: An array of `IAMClientTypes.AttachedPolicy` objects
    ///   describing each managed policy that's attached to the role.
    public func listAttachedRolePolicies(role: String) async throws ->
[IAMClientTypes.AttachedPolicy] {
        var policyList: [IAMClientTypes.AttachedPolicy] = []

        // Use "Paginated" to get all the attached role polices.
        // This lets the SDK handle the 'isTruncated' in
"ListAttachedRolePoliciesOutput".
        let input = ListAttachedRolePoliciesInput(
            roleName: role
        )
        let output = client.listAttachedRolePoliciesPaginated(input: input)

        do {
            for try await page in output {
                guard let attachedPolicies = page.attachedPolicies else {
                    print("Error: no attached policies returned.")
                    continue
                }
                for attachedPolicy in attachedPolicies {
                    policyList.append(attachedPolicy)
                }
            }
        } catch {
            print("ERROR: listAttachedRolePolicies:", dump(error))
            throw error
        }

        return policyList
    }
```

- Per i dettagli sull'API, consulta la ListAttachedRolePoliciesguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListAttachedUserPolicies`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListAttachedUserPolicies`.

CLI

AWS CLI

Per visualizzare tutte le policy gestite collegate all'utente specificato

Questo comando restituisce i nomi e gli ARN delle politiche gestite per l'utente IAM indicato Bob nell' AWS account.

```
aws iam list-attached-user-policies \
    --user-name Bob
```

Output:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListAttachedUserPolicies AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo comando restituisce i nomi e gli ARN delle politiche gestite per l'utente IAM indicato **Bob** nell' AWS account. Per visualizzare l'elenco delle policy in linea incorporate nell'utente IAM, usa il comando **Get-IAMUserPolicyList**.

```
Get-IAMAttachedUserPolicyList -UserName "Bob"
```

Output:

```
PolicyArn                                                 PolicyName
---------                                                 ----------
arn:aws:iam::aws:policy/TesterPolicy                      TesterPolicy
```

- Per i dettagli sull'API, vedere ListAttachedUserPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo comando restituisce i nomi e gli ARN delle politiche gestite per l'utente IAM indicato **Bob** nell' AWS account. Per visualizzare l'elenco delle policy in linea incorporate nell'utente IAM, usa il comando **Get-IAMUserPolicyList**.

```
Get-IAMAttachedUserPolicyList -UserName "Bob"
```

Output:

```
PolicyArn                                                 PolicyName
---------                                                 ----------
arn:aws:iam::aws:policy/TesterPolicy                      TesterPolicy
```

- Per i dettagli sull'API, vedere ListAttachedUserPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListEntitiesForPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListEntitiesForPolicy`.

CLI

AWS CLI

Per visualizzare tutti gli utenti, i gruppi e i ruoli a cui è collegata la policy gestita specificata

Questo esempio restituisce un elenco di gruppi, ruoli e utenti IAM a cui è collegata la policy `arn:aws:iam::123456789012:policy/TestPolicy`.

```
aws iam list-entities-for-policy \
    --policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

Output:

```
{
    "PolicyGroups": [
        {
            "GroupName": "Admins",
            "GroupId": "AGPACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyUsers": [
        {
            "UserName": "Alice",
            "UserId": "AIDACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyRoles": [
        {
            "RoleName": "DevRole",
            "RoleId": "AROADBQP57FF2AEXAMPLE"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListEntitiesForPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce un elenco di gruppi, ruoli e utenti IAM a cui è collegata la policy **`arn:aws:iam::123456789012:policy/TestPolicy`**.

```
Get-IAMEntitiesForPolicy -PolicyArn "arn:aws:iam::123456789012:policy/TestPolicy"
```

Output:

```
IsTruncated  : False
Marker       :
PolicyGroups : {}
PolicyRoles  : {testRole}
PolicyUsers  : {Bob, Theresa}
```

- Per i dettagli sull'API, vedere ListEntitiesForPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce un elenco di gruppi, ruoli e utenti IAM a cui è collegata la policy **`arn:aws:iam::123456789012:policy/TestPolicy`**.

```
Get-IAMEntitiesForPolicy -PolicyArn "arn:aws:iam::123456789012:policy/TestPolicy"
```

Output:

```
IsTruncated  : False
Marker       :
PolicyGroups : {}
PolicyRoles  : {testRole}
PolicyUsers  : {Bob, Theresa}
```

- Per i dettagli sull'API, vedere ListEntitiesForPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListGroupPolicies`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListGroupPolicies`.

CLI

AWS CLI

Per visualizzare tutte le policy in linea collegate al gruppo specificato

Il comando `list-group-policies` seguente elenca i nomi delle policy in linea collegate al gruppo IAM denominato `Admins` nell'account corrente.

```
aws iam list-group-policies \
    --group-name Admins
```

Output:

```
{
    "PolicyNames": [
        "AdminRoot",
        "ExamplePolicy"
    ]
}
```

Per ulteriori informazioni, consulta Gestione delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListGroupPolicies AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce l'elenco dei nomi delle policy in linea incorporate nel gruppo **`Testers`**. Per ottenere le policy gestite collegate al gruppo, utilizzare il comando **`Get-IAMAttachedGroupPolicyList`**.

```
Get-IAMGroupPolicyList -GroupName Testers
```

Output:

```
Deny-Assume-S3-Role-In-Production
PowerUserAccess-Testers
```

- Per i dettagli sull'API, vedere ListGroupPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce l'elenco dei nomi delle policy in linea incorporate nel gruppo **`Testers`**. Per ottenere le policy gestite collegate al gruppo, utilizzare il comando **`Get-IAMAttachedGroupPolicyList`**.

```
Get-IAMGroupPolicyList -GroupName Testers
```

Output:

```
Deny-Assume-S3-Role-In-Production
PowerUserAccess-Testers
```

- Per i dettagli sull'API, vedere ListGroupPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`ListGroups`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListGroups`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// List IAM groups.
    /// </summary>
    /// <returns>A list of IAM groups.</returns>
    public async Task<List<Group>> ListGroupsAsync()
    {
        var groupsPaginator = _IAMService.Paginators.ListGroups(new
ListGroupsRequest());
        var groups = new List<Group>();

        await foreach (var response in groupsPaginator.Responses)
        {
            groups.AddRange(response.Groups);
        }

        return groups;
    }
```

- Per i dettagli sull'API, ListGroupsconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come elencare i gruppi IAM per l'account corrente

Il comando `list-groups` seguente elenca i gruppi IAM nell'account corrente.

```
aws iam list-groups
```

Output:

```
{
    "Groups": [
        {
            "Path": "/",
            "CreateDate": "2013-06-04T20:27:27.972Z",
            "GroupId": "AIDACKCEVSQ6C2EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/Admins",
            "GroupName": "Admins"
        },
        {
            "Path": "/",
            "CreateDate": "2013-04-16T20:30:42Z",
            "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/S3-Admins",
            "GroupName": "S3-Admins"
        }
    ]
}
```

Per ulteriori informazioni, consulta Gestione di gruppi di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListGroups AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
```

```
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// GroupWrapper encapsulates AWS Identity and Access Management (IAM) group
 actions
// used in the examples.
// It contains an IAM service client that is used to perform group actions.
type GroupWrapper struct {
 IamClient *iam.Client
}



// ListGroups lists up to maxGroups number of groups.
func (wrapper GroupWrapper) ListGroups(ctx context.Context, maxGroups int32)
 ([]types.Group, error) {
 var groups []types.Group
 result, err := wrapper.IamClient.ListGroups(ctx, &iam.ListGroupsInput{
  MaxItems: aws.Int32(maxGroups),
 })
 if err != nil {
  log.Printf("Couldn't list groups. Here's why: %v\n", err)
 } else {
  groups = result.Groups
 }
 return groups, err
}
```

- Per i dettagli sull'API, ListGroupsconsulta AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca i gruppi.

```
import { ListGroupsCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 */
export async function* listGroups() {
  const command = new ListGroupsCommand({
    MaxItems: 10,
  });

  let response = await client.send(command);

  while (response.Groups?.length) {
    for (const group of response.Groups) {
      yield group;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListGroupsCommand({
          Marker: response.Marker,
          MaxItems: 10,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per i dettagli sull'API, ListGroupsconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listGroups($pathPrefix = "", $marker = "", $maxItems = 0)
    {
        $listGroupsArguments = [];
        if ($pathPrefix) {
            $listGroupsArguments["PathPrefix"] = $pathPrefix;
        }
        if ($marker) {
            $listGroupsArguments["Marker"] = $marker;
        }
        if ($maxItems) {
            $listGroupsArguments["MaxItems"] = $maxItems;
        }

        return $this->iamClient->listGroups($listGroupsArguments);
    }
```

- Per i dettagli sull'API, ListGroupsconsulta AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio restituisce una raccolta di tutti i gruppi IAM definiti nella versione corrente Account AWS.

```
Get-IAMGroupList
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Administrators
CreateDate : 10/20/2014 10:06:24 AM
GroupId    : 6WCH4TRY3KIHIEXAMPLE1
GroupName  : Administrators
Path       : /

Arn        : arn:aws:iam::123456789012:group/Developers
CreateDate : 12/10/2014 3:38:55 PM
GroupId    : ZU2EOWMK6WBZOEXAMPLE2
GroupName  : Developers
Path       : /

Arn        : arn:aws:iam::123456789012:group/Testers
CreateDate : 12/10/2014 3:39:11 PM
GroupId    : RHNZZGQJ7QHMAEXAMPLE3
GroupName  : Testers
Path       : /
```

- Per i dettagli sull'API, vedere ListGroupsin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio restituisce una raccolta di tutti i gruppi IAM definiti nella versione corrente Account AWS.

```
Get-IAMGroupList
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Administrators
CreateDate : 10/20/2014 10:06:24 AM
GroupId    : 6WCH4TRY3KIHIEXAMPLE1
GroupName  : Administrators
Path       : /

Arn        : arn:aws:iam::123456789012:group/Developers
CreateDate : 12/10/2014 3:38:55 PM
GroupId    : ZU2EOWMK6WBZOEXAMPLE2
GroupName  : Developers
```

```
Path       : /

Arn        : arn:aws:iam::123456789012:group/Testers
CreateDate : 12/10/2014 3:39:11 PM
GroupId    : RHNZZGQJ7QHMAEXAMPLE3
GroupName  : Testers
Path       : /
```

- Per i dettagli sull'API, vedere ListGroupsin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def list_groups(count):
    """
    Lists the specified number of groups for the account.

    :param count: The number of groups to list.
    """
    try:
        for group in iam.groups.limit(count):
            logger.info("Group: %s", group.name)
    except ClientError:
        logger.exception("Couldn't list groups for the account.")
        raise
```

- Per i dettagli sull'API, consulta ListGroups AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
# A class to manage IAM operations via the AWS SDK client
class IamGroupManager
  # Initializes the IamGroupManager class
  # @param iam_client [Aws::IAM::Client] An instance of the IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists up to a specified number of groups for the account.
  # @param count [Integer] The maximum number of groups to list.
  # @return [Aws::IAM::Client::Response]
  def list_groups(count)
    response = @iam_client.list_groups(max_items: count)
    response.groups.each do |group|
      @logger.info("\t#{group.group_name}")
    end
    response
  rescue Aws::Errors::ServiceError => e
    @logger.error("Couldn't list groups for the account. Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
end
```

- Per i dettagli sull'API, ListGroupsconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn list_groups(
    client: &iamClient,
    path_prefix: Option<String>,
    marker: Option<String>,
    max_items: Option<i32>,
) -> Result<ListGroupsOutput, SdkError<ListGroupsError>> {
    let response = client
        .list_groups()
        .set_path_prefix(path_prefix)
        .set_marker(marker)
        .set_max_items(max_items)
        .send()
        .await?;

    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento ListGroupsall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listgroups( ).
        MESSAGE 'Retrieved group list.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when listing groups.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, ListGroupsconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func listGroups() async throws -> [String] {
        var groupList: [String] = []

        // Use "Paginated" to get all the groups.
        // This lets the SDK handle the 'isTruncated' property in
 "ListGroupsOutput".
        let input = ListGroupsInput()

        let pages = client.listGroupsPaginated(input: input)
        do {
            for try await page in pages {
                guard let groups = page.groups else {
                    print("Error: no groups returned.")
                    continue
                }

                for group in groups {
```

```
                    if let name = group.groupName {
                        groupList.append(name)
                    }
                }
            }
        } catch {
            print("ERROR: listGroups:", dump(error))
            throw error
        }
        return groupList
    }
```

- Per i dettagli sull'API, consulta la ListGroupsguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListGroupsForUser`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListGroupsForUser`.

CLI

AWS CLI

Per visualizzare i gruppi a cui appartiene un utente IAM

Il comando `list-groups-for-user` seguente mostra i gruppi a cui appartiene l'utente IAM denominato Bob.

```
aws iam list-groups-for-user \
    --user-name Bob
```

Output:

```
{
    "Groups": [
        {
            "Path": "/",
            "CreateDate": "2013-05-06T01:18:08Z",
```

```
            "GroupId": "AKIAIOSFODNN7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/Admin",
            "GroupName": "Admin"
        },
        {
            "Path": "/",
            "CreateDate": "2013-05-06T01:37:28Z",
            "GroupId": "AKIAI44QH8DHBEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/s3-Users",
            "GroupName": "s3-Users"
        }
    ]
}
```

Per ulteriori informazioni, consulta Gestione di gruppi di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListGroupsForUser AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce l'elenco dei gruppi IAM a cui appartiene l'utente IAM **`David`**.

```
Get-IAMGroupForUser -UserName David
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Administrators
CreateDate : 10/20/2014 10:06:24 AM
GroupId    : 6WCH4TRY3KIHIEXAMPLE1
GroupName  : Administrators
Path       : /

Arn        : arn:aws:iam::123456789012:group/Testers
CreateDate : 12/10/2014 3:39:11 PM
GroupId    : RHNZZGQJ7QHMAEXAMPLE2
GroupName  : Testers
Path       : /
```

```
Arn        : arn:aws:iam::123456789012:group/Developers
CreateDate : 12/10/2014 3:38:55 PM
GroupId    : ZU2EOWMK6WBZOEXAMPLE3
GroupName  : Developers
Path       : /
```

- Per i dettagli sull'API, vedere ListGroupsForUserin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce l'elenco dei gruppi IAM a cui appartiene l'utente IAM **`David`**.

```
Get-IAMGroupForUser -UserName David
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Administrators
CreateDate : 10/20/2014 10:06:24 AM
GroupId    : 6WCH4TRY3KIHIEXAMPLE1
GroupName  : Administrators
Path       : /

Arn        : arn:aws:iam::123456789012:group/Testers
CreateDate : 12/10/2014 3:39:11 PM
GroupId    : RHNZZGQJ7QHMAEXAMPLE2
GroupName  : Testers
Path       : /

Arn        : arn:aws:iam::123456789012:group/Developers
CreateDate : 12/10/2014 3:38:55 PM
GroupId    : ZU2EOWMK6WBZOEXAMPLE3
GroupName  : Developers
Path       : /
```

- Per i dettagli sull'API, vedere ListGroupsForUserin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListInstanceProfiles`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListInstanceProfiles`.

CLI

AWS CLI

Per elencare i profili dell'istanza per l'account

Il `list-instance-profiles` comando seguente elenca i profili dell'istanza associati all'account corrente.

```
aws iam list-instance-profiles
```

Output:

```
{
    "InstanceProfiles": [
        {
            "Path": "/",
            "InstanceProfileName": "example-dev-role",
            "InstanceProfileId": "AIPAIXEU4NUHUPEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:instance-profile/example-dev-role",
            "CreateDate": "2023-09-21T18:17:41+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "example-dev-role",
                    "RoleId": "AROAJ52OTH4H7LEXAMPLE",
                    "Arn": "arn:aws:iam::123456789012:role/example-dev-role",
                    "CreateDate": "2023-09-21T18:17:40+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version":"2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
```

```
                    }
                }
            ]
        },
        {
            "Path": "/",
            "InstanceProfileName": "example-s3-role",
            "InstanceProfileId": "AIPAJVJVNRIQFREXAMPLE",
            "Arn": "arn:aws:iam::123456789012:instance-profile/example-s3-role",
            "CreateDate": "2023-09-21T18:18:50+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "example-s3-role",
                    "RoleId": "AROAINUBC5O7XLEXAMPLE",
                    "Arn": "arn:aws:iam::123456789012:role/example-s3-role",
                    "CreateDate": "2023-09-21T18:18:49+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version":"2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
                    }
                }
            ]
        }
    ]
}
```

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListInstanceProfiles AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio restituisce una raccolta dei profili di istanza definiti nella versione corrente Account AWS.

```
Get-IAMInstanceProfileList
```

Output:

```
Arn                 : arn:aws:iam::123456789012:instance-profile/ec2instancerole
CreateDate          : 2/17/2015 2:49:04 PM
InstanceProfileId   : HH36PTZQJUR32EXAMPLE1
InstanceProfileName : ec2instancerole
Path                : /
Roles               : {ec2instancerole}
```

- Per i dettagli sull'API, vedere ListInstanceProfilesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio restituisce una raccolta dei profili di istanza definiti nella versione corrente Account AWS.

```
Get-IAMInstanceProfileList
```

Output:

```
Arn                 : arn:aws:iam::123456789012:instance-profile/ec2instancerole
CreateDate          : 2/17/2015 2:49:04 PM
InstanceProfileId   : HH36PTZQJUR32EXAMPLE1
InstanceProfileName : ec2instancerole
Path                : /
Roles               : {ec2instancerole}
```

- Per i dettagli sull'API, vedere ListInstanceProfilesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListInstanceProfilesForRole`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListInstanceProfilesForRole`.

CLI

AWS CLI

Per visualizzare i profili dell'istanza per un ruolo IAM

Il comando `list-instance-profiles-for-role` seguente elenca i profili dell'istanza associati al ruolo `Test-Role`.

```
aws iam list-instance-profiles-for-role \
    --role-name Test-Role
```

Output:

```
{
    "InstanceProfiles": [
        {
            "InstanceProfileId": "AIDGPMS9RO4H3FEXAMPLE",
            "Roles": [
                {
                    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
                    "RoleId": "AIDACKCEVSQ6C2EXAMPLE",
                    "CreateDate": "2013-06-07T20:42:15Z",
                    "RoleName": "Test-Role",
                    "Path": "/",
                    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
                }
            ],
            "CreateDate": "2013-06-07T21:05:24Z",
            "InstanceProfileName": "ExampleInstanceProfile",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:instance-profile/
ExampleInstanceProfile"
        }
    ]
```

```
}
```

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListInstanceProfilesForRole AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce i dettagli del profilo dell'istanza associato al ruolo **`ec2instancerole`**.

```
Get-IAMInstanceProfileForRole -RoleName ec2instancerole
```

Output:

```
      Arn                 : arn:aws:iam::123456789012:instance-profile/
ec2instancerole
      CreateDate          : 2/17/2015 2:49:04 PM
      InstanceProfileId   : HH36PTZQJUR32EXAMPLE1
      InstanceProfileName : ec2instancerole
      Path                : /
      Roles               : {ec2instancerole}
```

- Per i dettagli sull'API, vedere ListInstanceProfilesForRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce i dettagli del profilo dell'istanza associato al ruolo **`ec2instancerole`**.

```
Get-IAMInstanceProfileForRole -RoleName ec2instancerole
```

Output:

```
      Arn                 : arn:aws:iam::123456789012:instance-profile/
ec2instancerole
      CreateDate          : 2/17/2015 2:49:04 PM
      InstanceProfileId   : HH36PTZQJUR32EXAMPLE1
```

```
      InstanceProfileName : ec2instancerole
      Path                : /
      Roles               : {ec2instancerole}
```

- Per i dettagli sull'API, vedere ListInstanceProfilesForRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListMfaDevices`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListMfaDevices`.

CLI

AWS CLI

Per elencare tutti i dispositivi MFA per un utente specificato

Questo esempio restituisce i dettagli del dispositivo MFA assegnato all'utente IAM Bob.

```
aws iam list-mfa-devices \
    --user-name Bob
```

Output:

```
{
    "MFADevices": [
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:mfa/Bob",
            "EnableDate": "2019-10-28T20:37:09+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "GAKT12345678",
            "EnableDate": "2023-02-18T21:44:42+00:00"
        },
        {
            "UserName": "Bob",
```

```
            "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey1-7XNL7NFNLZ123456789EXAMPLE",
            "EnableDate": "2023-09-19T02:25:35+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey2-VDRQTDBBN5123456789EXAMPLE",
            "EnableDate": "2023-09-19T01:49:18+00:00"
        }
    ]
}
```

Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella AWS Guida per l'utente IAM.

- Per i dettagli sull'API, consulta ListMfaDevices AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce i dettagli del dispositivo MFA assegnato all'utente IAM **`David`**. In questo esempio si può affermare che si tratta di un dispositivo virtuale perché **`SerialNumber`** è un ARN e non il numero di serie effettivo di un dispositivo fisico.

```
Get-IAMMFADevice -UserName David
```

Output:

```
EnableDate                SerialNumber                          UserName
----------                ------------                          --------
4/8/2015 9:41:10 AM       arn:aws:iam::123456789012:mfa/David   David
```

- Per i dettagli sull'API, vedere ListMfaDevicesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce i dettagli del dispositivo MFA assegnato all'utente IAM **`David`**. In questo esempio si può affermare che si tratta di un dispositivo virtuale perché **`SerialNumber`** è un ARN e non il numero di serie effettivo di un dispositivo fisico.

```
Get-IAMMFADevice -UserName David
```

Output:

```
EnableDate                  SerialNumber                            UserName
----------                  ------------                            --------
4/8/2015 9:41:10 AM         arn:aws:iam::123456789012:mfa/David     David
```

- Per i dettagli sull'API, vedere ListMfaDevicesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListOpenIdConnectProviders`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListOpenIdConnectProviders`.

CLI

AWS CLI

Per elencare informazioni sui provider OpenID Connect presenti nell'account AWS

Questo esempio restituisce un elenco di ARNS di tutti i provider OpenID Connect definiti AWS nell'account corrente.

```
aws iam list-open-id-connect-providers
```

Output:

```
{
    "OpenIDConnectProviderList": [
        {
            "Arn": "arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com"
        }
    ]
```

```
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità OpenID Connect (OIDC) nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta AWS CLI Command ListOpenIdConnectProvidersReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce un elenco di ARN di tutti i provider OpenID Connect definiti nell' Account AWS corrente.

```
Get-IAMOpenIDConnectProviderList
```

Output:

```
Arn
---
arn:aws:iam::123456789012:oidc-provider/server.example.com
arn:aws:iam::123456789012:oidc-provider/another.provider.com
```

- Per i dettagli sull'API, vedere ListOpenIdConnectProvidersin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce un elenco di ARN di tutti i provider OpenID Connect definiti nell' Account AWS corrente.

```
Get-IAMOpenIDConnectProviderList
```

Output:

```
Arn
---
arn:aws:iam::123456789012:oidc-provider/server.example.com
arn:aws:iam::123456789012:oidc-provider/another.provider.com
```

- Per i dettagli sull'API, vedere ListOpenIdConnectProvidersin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`ListPolicies`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListPolicies`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestione delle policy

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// List IAM policies.
    /// </summary>
    /// <returns>A list of the IAM policies.</returns>
    public async Task<List<ManagedPolicy>> ListPoliciesAsync()
    {
        var listPoliciesPaginator = _IAMService.Paginators.ListPolicies(new
ListPoliciesRequest());
        var policies = new List<ManagedPolicy>();

        await foreach (var response in listPoliciesPaginator.Responses)
        {
            policies.AddRange(response.Policies);
        }
```

```
        return policies;
    }
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::listPolicies(const Aws::Client::ClientConfiguration
 &clientConfig) {
    const Aws::String DATE_FORMAT("%Y-%m-%d");
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListPoliciesRequest request;

    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListPolicies(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list iam policies: " <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        if (!header) {
            std::cout << std::left << std::setw(55) << "Name" <<
                      std::setw(30) << "ID" << std::setw(80) << "Arn" <<
                      std::setw(64) << "Description" << std::setw(12) <<
                      "CreateDate" << std::endl;
            header = true;
        }
```

```
        const auto &policies = outcome.GetResult().GetPolicies();
        for (const auto &policy: policies) {
            std::cout << std::left << std::setw(55) <<
                      policy.GetPolicyName() << std::setw(30) <<
                      policy.GetPolicyId() << std::setw(80) << policy.GetArn() <<
                      std::setw(64) << policy.GetDescription() << std::setw(12)
 <<
                      policy.GetCreateDate().ToGmtString(DATE_FORMAT.c_str()) <<
                      std::endl;
        }

        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Per elencare le politiche gestite disponibili per il tuo AWS account

Questo esempio restituisce una raccolta delle prime due politiche gestite disponibili nell' AWS account corrente.

```
aws iam list-policies \
    --max-items 3
```

Output:

```
{
    "Policies": [
        {
```

```
            "PolicyName": "AWSCloudTrailAccessPolicy",
            "PolicyId": "ANPAXQE2B5PJ7YEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:policy/AWSCloudTrailAccessPolicy",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 0,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2019-09-04T17:43:42+00:00",
            "UpdateDate": "2019-09-04T17:43:42+00:00"
        },
        {
            "PolicyName": "AdministratorAccess",
            "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",
            "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 6,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:46+00:00",
            "UpdateDate": "2015-02-06T18:39:46+00:00"
        },
        {
            "PolicyName": "PowerUserAccess",
            "PolicyId": "ANPAJYRXTHIB4FOVS3ZXS",
            "Arn": "arn:aws:iam::aws:policy/PowerUserAccess",
            "Path": "/",
            "DefaultVersionId": "v5",
            "AttachmentCount": 1,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:47+00:00",
            "UpdateDate": "2023-07-06T22:04:00+00:00"
        }
    ],
    "NextToken": "EXAMPLErZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiA4fQ=="
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListPolicies AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// ListPolicies gets up to maxPolicies policies.
func (wrapper PolicyWrapper) ListPolicies(ctx context.Context, maxPolicies int32)
 ([]types.Policy, error) {
 var policies []types.Policy
 result, err := wrapper.IamClient.ListPolicies(ctx, &iam.ListPoliciesInput{
  MaxItems: aws.Int32(maxPolicies),
 })
 if err != nil {
  log.Printf("Couldn't list policies. Here's why: %v\n", err)
 } else {
  policies = result.Policies
```

```
 }
 return policies, err
}
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca le policy.

```
import { ListPoliciesCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 */
export async function* listPolicies() {
  const command = new ListPoliciesCommand({
    MaxItems: 10,
    OnlyAttached: false,
    // List only the customer managed policies in your Amazon Web Services
 account.
    Scope: "Local",
  });

  let response = await client.send(command);
```

```
  while (response.Policies?.length) {
    for (const policy of response.Policies) {
      yield policy;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListPoliciesCommand({
          Marker: response.Marker,
          MaxItems: 10,
          OnlyAttached: false,
          Scope: "Local",
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listPolicies($pathPrefix = "", $marker = "", $maxItems = 0)
    {
        $listPoliciesArguments = [];
        if ($pathPrefix) {
            $listPoliciesArguments["PathPrefix"] = $pathPrefix;
        }
```

```
        if ($marker) {
            $listPoliciesArguments["Marker"] = $marker;
        }
        if ($maxItems) {
            $listPoliciesArguments["MaxItems"] = $maxItems;
        }

        return $this->iamClient->listPolicies($listPoliciesArguments);
    }
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio restituisce una raccolta delle prime tre politiche gestite disponibili nell' AWS account corrente. Poiché non **`-scope`** è specificato, per **`all`** impostazione predefinita include sia le politiche AWS gestite che quelle gestite dai clienti.

```
Get-IAMPolicyList -MaxItem 3
```

Output:

```
Arn              : arn:aws:iam::aws:policy/AWSDirectConnectReadOnlyAccess
AttachmentCount  : 0
CreateDate       : 2/6/2015 10:40:08 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : Z27SI6FQMGNQ2EXAMPLE1
PolicyName       : AWSDirectConnectReadOnlyAccess
UpdateDate       : 2/6/2015 10:40:08 AM

Arn              : arn:aws:iam::aws:policy/AmazonGlacierReadOnlyAccess
AttachmentCount  : 0
CreateDate       : 2/6/2015 10:40:27 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
```

```
Path             : /
PolicyId         : NJKMU274MET4EEXAMPLE2
PolicyName       : AmazonGlacierReadOnlyAccess
UpdateDate       : 2/6/2015 10:40:27 AM

Arn              : arn:aws:iam::aws:policy/AWSMarketplaceFullAccess
AttachmentCount  : 0
CreateDate       : 2/11/2015 9:21:45 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : 5ULJSO2FYVPYGEXAMPLE3
PolicyName       : AWSMarketplaceFullAccess
UpdateDate       : 2/11/2015 9:21:45 AM
```

Esempio 2: Questo esempio restituisce una raccolta delle prime due politiche gestite dai clienti disponibili nell'account corrente AWS . Utilizza **`-Scope local`** per limitare l'output alle sole policy gestite dal cliente.

```
Get-IAMPolicyList -Scope local -MaxItem 2
```

Output:

```
Arn              : arn:aws:iam::123456789012:policy/MyLocalPolicy
AttachmentCount  : 0
CreateDate       : 2/12/2015 9:39:09 AM
DefaultVersionId : v2
Description      :
IsAttachable     : True
Path             : /
PolicyId         : SQVCBLC4VAOUCEXAMPLE4
PolicyName       : MyLocalPolicy
UpdateDate       : 2/12/2015 9:39:53 AM

Arn              : arn:aws:iam::123456789012:policy/policyforec2instancerole
AttachmentCount  : 1
CreateDate       : 2/17/2015 2:51:38 PM
DefaultVersionId : v11
Description      :
IsAttachable     : True
Path             : /
```

```
PolicyId         : X5JPBLJH2Z2SOEXAMPLE5
PolicyName       : policyforec2instancerole
UpdateDate       : 2/18/2015 8:52:31 AM
```

- Per i dettagli sull'API, vedere ListPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio restituisce una raccolta delle prime tre politiche gestite disponibili nell' AWS account corrente. Poiché non **`-scope`** è specificato, per **`all`** impostazione predefinita include sia le politiche AWS gestite che quelle gestite dai clienti.

```
Get-IAMPolicyList -MaxItem 3
```

Output:

```
Arn              : arn:aws:iam::aws:policy/AWSDirectConnectReadOnlyAccess
AttachmentCount  : 0
CreateDate       : 2/6/2015 10:40:08 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : Z27SI6FQMGNQ2EXAMPLE1
PolicyName       : AWSDirectConnectReadOnlyAccess
UpdateDate       : 2/6/2015 10:40:08 AM

Arn              : arn:aws:iam::aws:policy/AmazonGlacierReadOnlyAccess
AttachmentCount  : 0
CreateDate       : 2/6/2015 10:40:27 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : NJKMU274MET4EEXAMPLE2
PolicyName       : AmazonGlacierReadOnlyAccess
UpdateDate       : 2/6/2015 10:40:27 AM

Arn              : arn:aws:iam::aws:policy/AWSMarketplaceFullAccess
AttachmentCount  : 0
CreateDate       : 2/11/2015 9:21:45 AM
DefaultVersionId : v1
```

```
Description      :
IsAttachable     : True
Path             : /
PolicyId         : 5ULJSO2FYVPYGEXAMPLE3
PolicyName       : AWSMarketplaceFullAccess
UpdateDate       : 2/11/2015 9:21:45 AM
```

Esempio 2: Questo esempio restituisce una raccolta delle prime due politiche gestite dai clienti disponibili nell'account corrente AWS . Utilizza **`-Scope local`** per limitare l'output alle sole policy gestite dal cliente.

```
Get-IAMPolicyList -Scope local -MaxItem 2
```

Output:

```
Arn              : arn:aws:iam::123456789012:policy/MyLocalPolicy
AttachmentCount  : 0
CreateDate       : 2/12/2015 9:39:09 AM
DefaultVersionId : v2
Description      :
IsAttachable     : True
Path             : /
PolicyId         : SQVCBLC4VAOUCEXAMPLE4
PolicyName       : MyLocalPolicy
UpdateDate       : 2/12/2015 9:39:53 AM

Arn              : arn:aws:iam::123456789012:policy/policyforec2instancerole
AttachmentCount  : 1
CreateDate       : 2/17/2015 2:51:38 PM
DefaultVersionId : v11
Description      :
IsAttachable     : True
Path             : /
PolicyId         : X5JPBLJH2Z2SOEXAMPLE5
PolicyName       : policyforec2instancerole
UpdateDate       : 2/18/2015 8:52:31 AM
```

- Per i dettagli sull'API, vedere ListPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def list_policies(scope):
    """
    Lists the policies in the current account.

    :param scope: Limits the kinds of policies that are returned. For example,
                  'Local' specifies that only locally managed policies are
 returned.
    :return: The list of policies.
    """
    try:
        policies = list(iam.policies.filter(Scope=scope))
        logger.info("Got %s policies in scope '%s'.", len(policies), scope)
    except ClientError:
        logger.exception("Couldn't get policies for scope '%s'.", scope)
        raise
    else:
        return policies
```

- Per i dettagli sull'API, consulta ListPolicies AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, collega e scollega le policy relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'PolicyManager'
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end

  # Fetches an IAM policy by its ARN
  # @param policy_arn [String] the ARN of the IAM policy to retrieve
  # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
  def get_policy(policy_arn)
```

```
    response = @iam_client.get_policy(policy_arn: policy_arn)
    policy = response.policy
    @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
    policy
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
    raise
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
    raise
  end

  # Attaches a policy to a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_role(role_name, policy_arn)
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
```

```
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn list_policies(
    client: iamClient,
    path_prefix: String,
) -> Result<Vec<String>, SdkError<ListPoliciesError>> {
    let list_policies = client
        .list_policies()
        .path_prefix(path_prefix)
        .scope(PolicyScopeType::Local)
        .into_paginator()
        .items()
        .send()
        .try_collect()
        .await?;
```

```
    let policy_names = list_policies
        .into_iter()
        .map(|p| {
            let name = p
                .policy_name
                .unwrap_or_else(|| "Missing Policy Name".to_string());
            println!("{}", name);
            name
        })
        .collect();

    Ok(policy_names)
}
```

- Per i dettagli sulle API, consulta il riferimento ListPoliciesall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listpolicies( iv_scope = iv_scope ).
        MESSAGE 'Retrieved policy list.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when listing policies.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, ListPoliciesconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func listPolicies() async throws -> [MyPolicyRecord] {
        var policyList: [MyPolicyRecord] = []

        // Use "Paginated" to get all the policies.
        // This lets the SDK handle the 'isTruncated' in "ListPoliciesOutput".
        let input = ListPoliciesInput()
        let output = client.listPoliciesPaginated(input: input)

        do {
            for try await page in output {
                guard let policies = page.policies else {
                    print("Error: no policies returned.")
                    continue
                }

                for policy in policies {
                    guard let name = policy.policyName,
                          let id = policy.policyId,
                          let arn = policy.arn
                    else {
                        throw ServiceHandlerError.noSuchPolicy
                    }
                    policyList.append(MyPolicyRecord(name: name, id: id, arn:
 arn))
                }
            }
        } catch {
            print("ERROR: listPolicies:", dump(error))
```

```
            throw error
        }

        return policyList
    }
```

- Per i dettagli sull'API, consulta la ListPoliciesguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`ListPolicyVersions`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListPolicyVersions`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Gestione delle policy
- Rollback di una versione della policy

CLI

AWS CLI

Come visualizzare le informazioni sulle versioni della policy gestita specificata

Questo esempio restituisce l'elenco delle versioni disponibili della policy il cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam list-policy-versions \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Output:

```
{
    "IsTruncated": false,
```

```
    "Versions": [
        {
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2015-06-02T23:19:44Z"
        },
        {
        "VersionId": "v1",
        "IsDefaultVersion": false,
        "CreateDate": "2015-06-02T22:30:47Z"
        }
    ]
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListPolicyVersions AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce l'elenco delle versioni disponibili della policy il cui ARN è **`arn:aws:iam::123456789012:policy/MyManagedPolicy`**. Per ottenere il documento relativo alla policy per una versione specifica, utilizza il comando **`Get-IAMPolicyVersion`** e specifica l'**`VersionId`** della versione desiderata.

```
Get-IAMPolicyVersionList -PolicyArn arn:aws:iam::123456789012:policy/
MyManagedPolicy
```

Output:

```
CreateDate                  Document                IsDefaultVersion
       VersionId
----------                  --------                ----------------
       ---------
2/12/2015 9:39:53 AM                                True
       v2
2/12/2015 9:39:09 AM                                False
       v1
```

- Per i dettagli sull'API, vedere ListPolicyVersionsin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce l'elenco delle versioni disponibili della policy il cui ARN è **`arn:aws:iam::123456789012:policy/MyManagedPolicy`**. Per ottenere il documento relativo alla policy per una versione specifica, utilizza il comando **`Get-IAMPolicyVersion`** e specifica l'**`VersionId`** della versione desiderata.

```
Get-IAMPolicyVersionList -PolicyArn arn:aws:iam::123456789012:policy/
MyManagedPolicy
```

Output:

```
CreateDate                  Document                IsDefaultVersion
       VersionId
----------                  --------                ----------------
       ---------
2/12/2015 9:39:53 AM                                True
       v2
2/12/2015 9:39:09 AM                                False
       v1
```

- Per i dettagli sull'API, vedere ListPolicyVersionsin AWS Strumenti per PowerShell Cmdlet Reference (V5).

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listpolicyversions(
```

```
          iv_policyarn = iv_policy_arn ).
        MESSAGE 'Retrieved policy versions list.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Policy does not exist.' TYPE 'E'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when listing policy versions.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, ListPolicyVersionsconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`ListRolePolicies`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListRolePolicies`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// List IAM role policies.
    /// </summary>
    /// <param name="roleName">The IAM role for which to list IAM policies.</
param>
    /// <returns>A list of IAM policy names.</returns>
```

```
    public async Task<List<string>> ListRolePoliciesAsync(string roleName)
    {
        var listRolePoliciesPaginator =
_IAMService.Paginators.ListRolePolicies(new ListRolePoliciesRequest { RoleName =
roleName });
        var policyNames = new List<string>();

        await foreach (var response in listRolePoliciesPaginator.Responses)
        {
            policyNames.AddRange(response.PolicyNames);
        }

        return policyNames;
    }
```

- Per i dettagli sull'API, ListRolePoliciesconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come elencare le policy collegate a un ruolo IAM

Il comando `list-role-policies` seguente elenca i nomi delle policy di autorizzazione per il ruolo IAM specificato.

```
aws iam list-role-policies \
    --role-name Test-Role
```

Output:

```
{
    "PolicyNames": [
        "ExamplePolicy"
    ]
}
```

Per consultare la policy di attendibilità collegata a un ruolo, usa il comando `get-role`. Per visualizzare i dettagli di una policy di autorizzazioni, usa il comando `get-role-policy`.

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListRolePolicies AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// ListRolePolicies lists the inline policies for a role.
func (wrapper RoleWrapper) ListRolePolicies(ctx context.Context, roleName string)
 ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListRolePolicies(ctx,
 &iam.ListRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
```

```
 if err != nil {
  log.Printf("Couldn't list policies for role %v. Here's why: %v\n", roleName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}
```

- Per i dettagli sull'API, ListRolePoliciesconsulta AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca le policy.

```
import { ListRolePoliciesCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 * @param {string} roleName
 */
export async function* listRolePolicies(roleName) {
  const command = new ListRolePoliciesCommand({
    RoleName: roleName,
    MaxItems: 10,
  });
```

```
  let response = await client.send(command);

  while (response.PolicyNames?.length) {
    for (const policyName of response.PolicyNames) {
      yield policyName;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListRolePoliciesCommand({
          RoleName: roleName,
          MaxItems: 10,
          Marker: response.Marker,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per i dettagli sull'API, ListRolePoliciesconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listRolePolicies($roleName, $marker = "", $maxItems = 0)
    {
        $listRolePoliciesArguments = ['RoleName' => $roleName];
        if ($marker) {
```

```
            $listRolePoliciesArguments['Marker'] = $marker;
        }
        if ($maxItems) {
            $listRolePoliciesArguments['MaxItems'] = $maxItems;
        }
        return $this->customWaiter(function () use ($listRolePoliciesArguments) {
            return $this->iamClient-
>listRolePolicies($listRolePoliciesArguments);
        });
    }
```

- Per i dettagli sull'API, ListRolePoliciesconsulta AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio restituisce l'elenco dei nomi delle policy in linea incorporate nel ruolo IAM **`lamda_exec_role`**. Per visualizzare i dettagli di una policy in linea, usa il comando **`Get-IAMRolePolicy`**.

```
Get-IAMRolePolicyList -RoleName lambda_exec_role
```

Output:

```
oneClick_lambda_exec_role_policy
```

- Per i dettagli sull'API, vedere ListRolePoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio restituisce l'elenco dei nomi delle policy in linea incorporate nel ruolo IAM **`lamda_exec_role`**. Per visualizzare i dettagli di una policy in linea, usa il comando **`Get-IAMRolePolicy`**.

```
Get-IAMRolePolicyList -RoleName lambda_exec_role
```

Output:

```
oneClick_lambda_exec_role_policy
```

- Per i dettagli sull'API, vedere ListRolePoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def list_policies(role_name):
    """
    Lists inline policies for a role.

    :param role_name: The name of the role to query.
    """
    try:
        role = iam.Role(role_name)
        for policy in role.policies.all():
            logger.info("Got inline policy %s.", policy.name)
    except ClientError:
        logger.exception("Couldn't list inline policies for %s.", role_name)
        raise
```

- Per i dettagli sull'API, consulta ListRolePolicies AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end
```

- Per i dettagli sull'API, ListRolePoliciesconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn list_role_policies(
    client: &iamClient,
    role_name: &str,
    marker: Option<String>,
    max_items: Option<i32>,
```

```
) -> Result<ListRolePoliciesOutput, SdkError<ListRolePoliciesError>> {
    let response = client
        .list_role_policies()
        .role_name(role_name)
        .set_marker(marker)
        .set_max_items(max_items)
        .send()
        .await?;

    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento ListRolePoliciesall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listrolepolicies(
          iv_rolename = iv_role_name ).
        MESSAGE 'Retrieved inline policy list for role.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Role does not exist.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, ListRolePoliciesconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func listRolePolicies(role: String) async throws -> [String] {
        var policyList: [String] = []

        // Use "Paginated" to get all the role policies.
        // This lets the SDK handle the 'isTruncated' in
 "ListRolePoliciesOutput".
        let input = ListRolePoliciesInput(
            roleName: role
        )
        let pages = client.listRolePoliciesPaginated(input: input)

        do {
            for try await page in pages {
                guard let policies = page.policyNames else {
                    print("Error: no role policies returned.")
                    continue
                }

                for policy in policies {
                    policyList.append(policy)
                }
            }
        } catch {
            print("ERROR: listRolePolicies:", dump(error))
            throw error
        }
        return policyList
    }
```

- Per i dettagli sull'API, consulta la ListRolePoliciesguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListRoleTags`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListRoleTags`.

CLI

AWS CLI

Elencare i tag collegati a un ruolo

Il comando `list-role-tags` seguente recupera l'elenco dei tag associati al ruolo specificato.

```
aws iam list-role-tags \
    --role-name production-role
```

Output:

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Applicazione di tag a risorse IAM nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta ListRoleTags AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera il tag associato al ruolo.

```
Get-IAMRoleTagList -RoleName MyRoleName
```

- Per i dettagli sull'API, vedere ListRoleTagsin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera il tag associato al ruolo.

```
Get-IAMRoleTagList -RoleName MyRoleName
```

- Per i dettagli sull'API, vedere ListRoleTagsin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`ListRoles`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListRoles`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// List IAM roles.
    /// </summary>
    /// <returns>A list of IAM roles.</returns>
    public async Task<List<Role>> ListRolesAsync()
    {
        var listRolesPaginator = _IAMService.Paginators.ListRoles(new
 ListRolesRequest());
        var roles = new List<Role>();

        await foreach (var response in listRolesPaginator.Responses)
        {
            roles.AddRange(response.Roles);
        }

        return roles;
    }
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come elencare i ruoli IAM per l'account corrente

Il comando `list-roles` seguente elenca i ruoli IAM per l'account corrente.

```
aws iam list-roles
```

Output:

```
{
    "Roles": [
        {
            "Path": "/",
            "RoleName": "ExampleRole",
            "RoleId": "AROAJ52OTH4H7LEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/ExampleRole",
```

```
            "CreateDate": "2017-09-12T19:23:36+00:00",
            "AssumeRolePolicyDocument": {
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/example_path/",
            "RoleName": "ExampleRoleWithPath",
            "RoleId": "AROAI4QRP7UFT7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/example_path/
ExampleRoleWithPath",
            "CreateDate": "2023-09-21T20:29:38+00:00",
            "AssumeRolePolicyDocument": {
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        }
    ]
}
```

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListRoles AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// ListRoles gets up to maxRoles roles.
func (wrapper RoleWrapper) ListRoles(ctx context.Context, maxRoles int32)
 ([]types.Role, error) {
 var roles []types.Role
 result, err := wrapper.IamClient.ListRoles(ctx,
  &iam.ListRolesInput{MaxItems: aws.Int32(maxRoles)},
 )
 if err != nil {
  log.Printf("Couldn't list roles. Here's why: %v\n", err)
 } else {
  roles = result.Roles
 }
```

```
 return roles, err
}
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK per Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca i ruoli.

```
import { ListRolesCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 */
export async function* listRoles() {
  const command = new ListRolesCommand({
    MaxItems: 10,
  });

  /**
   * @type {import("@aws-sdk/client-iam").ListRolesCommandOutput | undefined}
   */
  let response = await client.send(command);

  while (response?.Roles?.length) {
    for (const role of response.Roles) {
```

```
      yield role;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListRolesCommand({
          Marker: response.Marker,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    /**
     * @param string $pathPrefix
     * @param string $marker
     * @param int $maxItems
     * @return Result
     * $roles = $service->listRoles();
     */
    public function listRoles($pathPrefix = "", $marker = "", $maxItems = 0)
    {
        $listRolesArguments = [];
        if ($pathPrefix) {
```

```
            $listRolesArguments["PathPrefix"] = $pathPrefix;
        }
        if ($marker) {
            $listRolesArguments["Marker"] = $marker;
        }
        if ($maxItems) {
            $listRolesArguments["MaxItems"] = $maxItems;
        }
        return $this->iamClient->listRoles($listRolesArguments);
    }
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera un elenco di tutti i ruoli IAM nell' Account AWS.

```
Get-IAMRoleList
```

- Per i dettagli sull'API, vedere ListRolesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera un elenco di tutti i ruoli IAM nell' Account AWS.

```
Get-IAMRoleList
```

- Per i dettagli sull'API, vedere ListRolesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def list_roles(count):
    """
    Lists the specified number of roles for the account.

    :param count: The number of roles to list.
    """
    try:
        roles = list(iam.roles.limit(count=count))
        for role in roles:
            logger.info("Role: %s", role.name)
    except ClientError:
        logger.exception("Couldn't list roles for the account.")
        raise
    else:
        return roles
```

- Per i dettagli sull'API, consulta ListRoles AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
  # Lists IAM roles up to a specified count.
  # @param count [Integer] the maximum number of roles to list.
  # @return [Array<String>] the names of the roles.
  def list_roles(count)
    role_names = []
    roles_counted = 0

    @iam_client.list_roles.each_page do |page|
      page.roles.each do |role|
        break if roles_counted >= count

        @logger.info("\t#{roles_counted + 1}: #{role.role_name}")
        role_names << role.role_name
        roles_counted += 1
      end
      break if roles_counted >= count
    end

    role_names
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't list roles for the account. Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn list_roles(
    client: &iamClient,
    path_prefix: Option<String>,
```

```
    marker: Option<String>,
    max_items: Option<i32>,
) -> Result<ListRolesOutput, SdkError<ListRolesError>> {
    let response = client
        .list_roles()
        .set_path_prefix(path_prefix)
        .set_marker(marker)
        .set_max_items(max_items)
        .send()
        .await?;
    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento ListRolesall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listroles( ).
        MESSAGE 'Retrieved role list.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when listing roles.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, ListRolesconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func listRoles() async throws -> [String] {
        var roleList: [String] = []

        // Use "Paginated" to get all the roles.
        // This lets the SDK handle the 'isTruncated' in "ListRolesOutput".
        let input = ListRolesInput()
        let pages = client.listRolesPaginated(input: input)

        do {
            for try await page in pages {
                guard let roles = page.roles else {
                    print("Error: no roles returned.")
                    continue
                }

                for role in roles {
                    if let name = role.roleName {
                        roleList.append(name)
                    }
                }
            }
        } catch {
            print("ERROR: listRoles:", dump(error))
            throw error
        }
        return roleList
    }
```

- Per i dettagli sull'API, consulta la ListRolesguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usa **`ListSamlProviders`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListSAMLProviders`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// List SAML authentication providers.
    /// </summary>
    /// <returns>A list of SAML providers.</returns>
    public async Task<List<SAMLProviderListEntry>> ListSAMLProvidersAsync()
    {
        var response = await _IAMService.ListSAMLProvidersAsync(new
ListSAMLProvidersRequest());
        return response.SAMLProviderList;
    }
```

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella documentazione di riferimento dell'API AWS SDK per .NET .

CLI

AWS CLI

Per elencare i provider SAML presenti nell'account AWS

Questo esempio recupera l'elenco dei provider SAML 2.0 creati nell'account corrente. AWS

```
aws iam list-saml-providers
```

Output:

```
{
    "SAMLProviderList": [
        {
            "Arn": "arn:aws:iam::123456789012:saml-provider/SAML-ADFS",
            "ValidUntil": "2015-06-05T22:45:14Z",
            "CreateDate": "2015-06-05T22:45:14Z"
        }
    ]
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella Documentazione di riferimento dei comandi della AWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
```

```
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
)

// AccountWrapper encapsulates AWS Identity and Access Management (IAM) account
 actions
// used in the examples.
// It contains an IAM service client that is used to perform account actions.
type AccountWrapper struct {
 IamClient *iam.Client
}



// ListSAMLProviders gets the SAML providers for the account.
func (wrapper AccountWrapper) ListSAMLProviders(ctx context.Context)
 ([]types.SAMLProviderListEntry, error) {
 var providers []types.SAMLProviderListEntry
 result, err := wrapper.IamClient.ListSAMLProviders(ctx,
 &iam.ListSAMLProvidersInput{})
 if err != nil {
  log.Printf("Couldn't list SAML providers. Here's why: %v\n", err)
 } else {
  providers = result.SAMLProviderList
 }
 return providers, err
}
```

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella documentazione di riferimento dell'API AWS SDK per Go .

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca i provider SAML.

```
import { ListSAMLProvidersCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

export const listSamlProviders = async () => {
  const command = new ListSAMLProvidersCommand({});

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella documentazione di riferimento dell'API AWS SDK per JavaScript .

PHP

SDK per PHP

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
```

```
$service = new IAMService();

    public function listSAMLProviders()
    {
        return $this->iamClient->listSAMLProviders();
    }
```

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella documentazione di riferimento dell'API AWS SDK per PHP .

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera l'elenco dei provider SAML 2.0 creati nell' Account AWS corrente. Restituisce l'ARN, la data di creazione e la data di scadenza per ogni provider SAML.

```
Get-IAMSAMLProviderList
```

Output:

```
Arn                                                  CreateDate
   ValidUntil
---                                                  ----------
   ----------
arn:aws:iam::123456789012:saml-provider/SAMLADFS     12/23/2014 12:16:55 PM
   12/23/2114 12:16:54 PM
```

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella documentazione di riferimento dei cmdlet di AWS Strumenti per PowerShell (V4).

Strumenti per PowerShell V5

Esempio 1: questo esempio recupera l'elenco dei provider SAML 2.0 creati nell' Account AWS corrente. Restituisce l'ARN, la data di creazione e la data di scadenza per ogni provider SAML.

```
Get-IAMSAMLProviderList
```

Output:

```
Arn                                                    CreateDate
   ValidUntil
---                                                    ----------
   ----------
arn:aws:iam::123456789012:saml-provider/SAMLADFS    12/23/2014 12:16:55 PM
   12/23/2114 12:16:54 PM
```

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella documentazione di riferimento dei cmdlet di AWS Strumenti per PowerShell (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def list_saml_providers(count):
    """
    Lists the SAML providers for the account.

    :param count: The maximum number of providers to list.
    """
    try:
        found = 0
        for provider in iam.saml_providers.limit(count):
            logger.info("Got SAML provider %s.", provider.arn)
            found += 1
        if found == 0:
            logger.info("Your account has no SAML providers.")
    except ClientError:
        logger.exception("Couldn't list SAML providers.")
        raise
```

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella documentazione di riferimento dell'API SDK AWS per Python (Boto3).

Ruby

SDK per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
class SamlProviderLister
  # Initializes the SamlProviderLister with IAM client and a logger.
  # @param iam_client [Aws::IAM::Client] The IAM client object.
  # @param logger [Logger] The logger object for logging output.
  def initialize(iam_client, logger = Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists up to a specified number of SAML providers for the account.
  # @param count [Integer] The maximum number of providers to list.
  # @return [Aws::IAM::Client::Response]
  def list_saml_providers(count)
    response = @iam_client.list_saml_providers
    response.saml_provider_list.take(count).each do |provider|
      @logger.info("\t#{provider.arn}")
    end
    response
  rescue Aws::Errors::ServiceError => e
    @logger.error("Couldn't list SAML providers. Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
end
```

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella documentazione di riferimento dell'API AWS SDK per Ruby .

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn list_saml_providers(
    client: &Client,
) -> Result<ListSamlProvidersOutput, SdkError<ListSAMLProvidersError>> {
    let response = client.list_saml_providers().send().await?;

    Ok(response)
}
```

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella documentazione di riferimento dell'API SDK AWS per Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listsamlproviders( ).
        MESSAGE 'Retrieved SAML provider list.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when listing SAML providers.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sull'API, consulta ListSamlProviders in AWS SDK per il riferimento all'API SAP ABAP.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`ListServerCertificates`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListServerCertificates`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::listServerCertificates(
        const Aws::Client::ClientConfiguration &clientConfig) {
    const Aws::String DATE_FORMAT = "%Y-%m-%d";

    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListServerCertificatesRequest request;

    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListServerCertificates(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list server certificates: " <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        if (!header) {
            std::cout << std::left << std::setw(55) << "Name" <<
                      std::setw(30) << "ID" << std::setw(80) << "Arn" <<
```

```
                        std::setw(14) << "UploadDate" << std::setw(14) <<
                        "ExpirationDate" << std::endl;
            header = true;
        }

        const auto &certificates =
                outcome.GetResult().GetServerCertificateMetadataList();

        for (const auto &certificate: certificates) {
            std::cout << std::left << std::setw(55) <<
                      certificate.GetServerCertificateName() << std::setw(30) <<
                      certificate.GetServerCertificateId() << std::setw(80) <<
                      certificate.GetArn() << std::setw(14) <<

 certificate.GetUploadDate().ToGmtString(DATE_FORMAT.c_str()) <<
                      std::setw(14) <<

 certificate.GetExpiration().ToGmtString(DATE_FORMAT.c_str()) <<
                      std::endl;
        }

        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, ListServerCertificatesconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Per elencare i certificati del server presenti nel tuo AWS account

Il `list-server-certificates` comando seguente elenca tutti i certificati server archiviati e disponibili per l'uso nell' AWS account.

```
aws iam list-server-certificates
```

Output:

```
{
    "ServerCertificateMetadataList": [
        {
            "Path": "/",
            "ServerCertificateName": "myUpdatedServerCertificate",
            "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/
myUpdatedServerCertificate",
            "UploadDate": "2019-04-22T21:13:44+00:00",
            "Expiration": "2019-10-15T22:23:16+00:00"
        },
        {
            "Path": "/cloudfront/",
            "ServerCertificateName": "MyTestCert",
            "ServerCertificateId": "ASCAEXAMPLE456EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyTestCert",
            "UploadDate": "2015-04-21T18:14:16+00:00",
            "Expiration": "2018-01-14T17:52:36+00:00"
        }
    ]
}
```

Per ulteriori informazioni, consulta Gestione dei certificati server in IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListServerCertificates AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca i certificati.

```
import { ListServerCertificatesCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 */
export async function* listServerCertificates() {
  const command = new ListServerCertificatesCommand({});
  let response = await client.send(command);

  while (response.ServerCertificateMetadataList?.length) {
    for await (const cert of response.ServerCertificateMetadataList) {
      yield cert;
    }

    if (response.IsTruncated) {
      response = await client.send(new ListServerCertificatesCommand({}));
    } else {
      break;
    }
  }
}
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, ListServerCertificatesconsulta AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.listServerCertificates({}, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, ListServerCertificatesconsulta AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera l'elenco dei certificati del server che sono stati caricati nell' Account AWS corrente.

```
Get-IAMServerCertificateList
```

Output:

```
Arn                   : arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyServerCertificate
Expiration            : 1/14/2018 9:52:36 AM
Path                  : /Org1/Org2/
ServerCertificateId   : ASCAJIFEXAMPLE17HQZYW
ServerCertificateName : MyServerCertificate
UploadDate            : 4/21/2015 11:14:16 AM
```

- Per i dettagli sull'API, vedere ListServerCertificatesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera l'elenco dei certificati del server che sono stati caricati nell' Account AWS corrente.

```
Get-IAMServerCertificateList
```

Output:

```
Arn                   : arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyServerCertificate
Expiration            : 1/14/2018 9:52:36 AM
Path                  : /Org1/Org2/
ServerCertificateId   : ASCAJIFEXAMPLE17HQZYW
ServerCertificateName : MyServerCertificate
UploadDate            : 4/21/2015 11:14:16 AM
```

- Per i dettagli sull'API, vedere ListServerCertificatesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca, aggiorna ed elimina i certificati server.

```
class ServerCertificateManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'ServerCertificateManager'
  end
```

```
  # Creates a new server certificate.
  # @param name [String] the name of the server certificate
  # @param certificate_body [String] the contents of the certificate
  # @param private_key [String] the private key contents
  # @return [Boolean] returns true if the certificate was successfully created
  def create_server_certificate(name, certificate_body, private_key)
    @iam_client.upload_server_certificate({
                                            server_certificate_name: name,
                                            certificate_body: certificate_body,
                                            private_key: private_key
                                          })
    true
  rescue Aws::IAM::Errors::ServiceError => e
    puts "Failed to create server certificate: #{e.message}"
    false
  end

  # Lists available server certificate names.
  def list_server_certificate_names
    response = @iam_client.list_server_certificates

    if response.server_certificate_metadata_list.empty?
      @logger.info('No server certificates found.')
      return
    end

    response.server_certificate_metadata_list.each do |certificate_metadata|
      @logger.info("Certificate Name:
#{certificate_metadata.server_certificate_name}")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing server certificates: #{e.message}")
  end

  # Updates the name of a server certificate.
  def update_server_certificate_name(current_name, new_name)
    @iam_client.update_server_certificate(
      server_certificate_name: current_name,
      new_server_certificate_name: new_name
    )
    @logger.info("Server certificate name updated from '#{current_name}' to
'#{new_name}'.")
    true
```

```
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error updating server certificate name: #{e.message}")
    false
  end

  # Deletes a server certificate.
  def delete_server_certificate(name)
    @iam_client.delete_server_certificate(server_certificate_name: name)
    @logger.info("Server certificate '#{name}' deleted.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting server certificate: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, ListServerCertificatesconsulta AWS SDK per Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListSigningCertificates`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListSigningCertificates`.

CLI

AWS CLI

Per elencare i certificati di firma per un utente IAM

Il comando `list-signing-certificates` seguente elenca i certificati di firma per l'utente IAM denominato Bob.

```
aws iam list-signing-certificates \
    --user-name Bob
```

Output:

```
{
```

```
    "Certificates": [
        {
            "UserName": "Bob",
            "Status": "Inactive",
            "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-
body>-----END CERTIFICATE-----",
            "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
            "UploadDate": "2013-06-06T21:40:08Z"
        }
    ]
}
```

Per ulteriori informazioni, consulta Gestione dei certificati di firma nella Guida per l'utente di Amazon EC2.

- Per i dettagli sull'API, consulta ListSigningCertificates AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio recupera i dettagli del certificato di firma associato all'utente denominato **Bob**.

```
Get-IAMSigningCertificate -UserName Bob
```

Output:

```
CertificateBody : -----BEGIN CERTIFICATE-----

 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC

 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6

 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd

 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN

 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD

 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
```

```
 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
                  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn
+a4GmWIWJ
                  21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9T
                  rDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE

 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
                  nUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb

 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
                  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
                  -----END CERTIFICATE-----
CertificateId   : Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU
Status          : Active
UploadDate      : 4/20/2015 1:26:01 PM
UserName        : Bob
```

- Per i dettagli sull'API, vedere ListSigningCertificatesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio recupera i dettagli del certificato di firma associato all'utente denominato **Bob**.

```
Get-IAMSigningCertificate -UserName Bob
```

Output:

```
CertificateBody : -----BEGIN CERTIFICATE-----

 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC

 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6

 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd

 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
```

```
 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD

 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z

 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
                  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn
+a4GmWIWJ
                  21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9T
                  rDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE

 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
                  nUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb

 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
                  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
                  -----END CERTIFICATE-----
CertificateId   : Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU
Status          : Active
UploadDate      : 4/20/2015 1:26:01 PM
UserName        : Bob
```

- Per i dettagli sull'API, vedere ListSigningCertificatesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`ListUserPolicies`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListUserPolicies`.

CLI

AWS CLI

Come elencare le policy per un utente IAM

Il comando `list-user-policies` seguente elenca le policy collegate all'utente IAM denominato Bob.

```
aws iam list-user-policies \
    --user-name Bob
```

Output:

```
{
    "PolicyNames": [
        "ExamplePolicy",
        "TestPolicy"
    ]
}
```

Per ulteriori informazioni, consulta Creating an IAM user in your AWS account nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta ListUserPolicies AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
```

```
 "github.com/aws/smithy-go"
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListUserPolicies lists the inline policies for the specified user.
func (wrapper UserWrapper) ListUserPolicies(ctx context.Context, userName string)
 ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListUserPolicies(ctx,
 &iam.ListUserPoliciesInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list policies for user %v. Here's why: %v\n", userName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}
```

- Per i dettagli sull'API, consulta la ListUserPoliciessezione AWS SDK per Go API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera l'elenco dei nomi delle policy in linea incorporate nell'utente IAM denominato **`David`**.

```
Get-IAMUserPolicyList -UserName David
```

Output:

```
Davids_IAM_Admin_Policy
```

- Per i dettagli sull'API, vedere ListUserPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera l'elenco dei nomi delle policy in linea incorporate nell'utente IAM denominato **`David`**.

```
Get-IAMUserPolicyList -UserName David
```

Output:

```
Davids_IAM_Admin_Policy
```

- Per i dettagli sull'API, vedere ListUserPoliciesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListUserTags`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListUserTags`.

CLI

AWS CLI

Elencare i tag collegati a un utente

Il comando `list-user-tags` seguente recupera l'elenco dei tag associati all'utente IAM specificato.

```
aws iam list-user-tags \
    --user-name alice
```

Output:

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Applicazione di tag a risorse IAM nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta ListUserTags AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera il tag associato all'utente.

```
Get-IAMUserTagList -UserName joe
```

- Per i dettagli sull'API, vedere ListUserTagsin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera il tag associato all'utente.

```
Get-IAMUserTagList -UserName joe
```

- Per i dettagli sull'API, vedere ListUserTagsin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`ListUsers`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListUsers`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di utenti di sola lettura e di lettura e scrittura

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// List IAM users.
    /// </summary>
    /// <returns>A list of IAM users.</returns>
    public async Task<List<User>> ListUsersAsync()
    {
        var listUsersPaginator = _IAMService.Paginators.ListUsers(new
ListUsersRequest());
        var users = new List<User>();

        await foreach (var response in listUsersPaginator.Responses)
        {
            users.AddRange(response.Users);
        }

        return users;
    }
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_list_users
#
# List the IAM users in the account.
#
# Returns:
#       The list of users names
#    And:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
###############################################################################
function iam_list_users() {
  local option OPTARG # Required to use getopts command in a function.
  local error_code
  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_list_users"
    echo "Lists the AWS Identity and Access Management (IAM) user in the
 account."
```

```
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "h" option; do
    case "${option}" in
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response

  response=$(aws iam list-users \
    --output text \
    --query "Users[].UserName")
  error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-users operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta ListUsers AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::listUsers(const Aws::Client::ClientConfiguration &clientConfig)
 {
    const Aws::String DATE_FORMAT = "%Y-%m-%d";
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListUsersRequest request;

    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListUsers(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list iam users:" <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        if (!header) {
            std::cout << std::left << std::setw(32) << "Name" <<
                      std::setw(30) << "ID" << std::setw(64) << "Arn" <<
                      std::setw(20) << "CreateDate" << std::endl;
            header = true;
        }

        const auto &users = outcome.GetResult().GetUsers();
        for (const auto &user: users) {
            std::cout << std::left << std::setw(32) << user.GetUserName() <<
                      std::setw(30) << user.GetUserId() << std::setw(64) <<
                      user.GetArn() << std::setw(20) <<
                      user.GetCreateDate().ToGmtString(DATE_FORMAT.c_str())
                      << std::endl;
        }
```

```
        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come elencare gli utenti IAM

Il comando `list-users` seguente elenca gli utenti IAM nell'account corrente.

```
aws iam list-users
```

Output:

```
{
    "Users": [
        {
            "UserName": "Adele",
            "Path": "/",
            "CreateDate": "2013-03-07T05:14:48Z",
            "UserId": "AKIAI44QH8DHBEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Adele"
        },
        {
            "UserName": "Bob",
            "Path": "/",
            "CreateDate": "2012-09-21T23:03:13Z",
            "UserId": "AKIAIOSFODNN7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Bob"
        }
    ]
```

```
}
```

Per ulteriori informazioni, consulta Elencazione degli utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListUsers AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListUsers gets up to maxUsers number of users.
func (wrapper UserWrapper) ListUsers(ctx context.Context, maxUsers int32)
 ([]types.User, error) {
 var users []types.User
```

```
 result, err := wrapper.IamClient.ListUsers(ctx, &iam.ListUsersInput{
  MaxItems: aws.Int32(maxUsers),
 })
 if err != nil {
  log.Printf("Couldn't list users. Here's why: %v\n", err)
 } else {
  users = result.Users
 }
 return users, err
}
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK per Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.AttachedPermissionsBoundary;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.ListUsersRequest;
import software.amazon.awssdk.services.iam.model.ListUsersResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.User;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class ListUsers {
    public static void main(String[] args) {
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        listAllUsers(iam);
        System.out.println("Done");
        iam.close();
    }

    public static void listAllUsers(IamClient iam) {
        try {
            boolean done = false;
            String newMarker = null;
            while (!done) {
                ListUsersResponse response;
                if (newMarker == null) {
                    ListUsersRequest request =
 ListUsersRequest.builder().build();
                    response = iam.listUsers(request);
                } else {
                    ListUsersRequest request = ListUsersRequest.builder()
                            .marker(newMarker)
                            .build();

                    response = iam.listUsers(request);
                }

                for (User user : response.users()) {
                    System.out.format("\n Retrieved user %s", user.userName());
                    AttachedPermissionsBoundary permissionsBoundary =
 user.permissionsBoundary();
                    if (permissionsBoundary != null)
                        System.out.format("\n Permissions boundary details %s",

 permissionsBoundary.permissionsBoundaryTypeAsString());
                }

                if (!response.isTruncated()) {
                    done = true;
                } else {
                    newMarker = response.marker();
```

```
                }
            }

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca gli utenti.

```
import { ListUsersCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

export const listUsers = async () => {
  const command = new ListUsersCommand({ MaxItems: 10 });

  const response = await client.send(command);

  for (const { UserName, CreateDate } of response.Users) {
    console.log(`${UserName} created on: ${CreateDate}`);
  }
  return response;
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  MaxItems: 10,
};

iam.listUsers(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    var users = data.Users || [];
    users.forEach(function (user) {
      console.log("User " + user.UserName + " created", user.CreateDate);
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun listAllUsers() {
    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listUsers(ListUsersRequest { })
        response.users?.forEach { user ->
            println("Retrieved user ${user.userName}")
            val permissionsBoundary = user.permissionsBoundary
            if (permissionsBoundary != null) {
                println("Permissions boundary details
 ${permissionsBoundary.permissionsBoundaryType}")
            }
        }
    }
}
```

- Per i dettagli sull'API, ListUsersconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();
```

```
    public function listUsers($pathPrefix = "", $marker = "", $maxItems = 0)
    {
        $listUsersArguments = [];
        if ($pathPrefix) {
            $listUsersArguments["PathPrefix"] = $pathPrefix;
        }
        if ($marker) {
            $listUsersArguments["Marker"] = $marker;
        }
        if ($maxItems) {
            $listUsersArguments["MaxItems"] = $maxItems;
        }

        return $this->iamClient->listUsers($listUsersArguments);
    }
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio recupera una raccolta di utenti nella cartella corrente. Account AWS

```
Get-IAMUserList
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Administrator
CreateDate       : 10/16/2014 9:03:09 AM
PasswordLastUsed : 3/4/2015 12:12:33 PM
Path             : /
UserId           : 7K3GJEANSKZF2EXAMPLE1
UserName         : Administrator

Arn              : arn:aws:iam::123456789012:user/Bob
CreateDate       : 4/6/2015 12:54:42 PM
PasswordLastUsed : 1/1/0001 12:00:00 AM
Path             : /
UserId           : L3EWNONDOM3YUEXAMPLE2
```

```
UserName         : bab

Arn              : arn:aws:iam::123456789012:user/David
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 3/19/2015 8:44:04 AM
Path             : /
UserId           : Y4FKWQCXTA52QEXAMPLE3
UserName         : David
```

- Per i dettagli sull'API, vedere ListUsersin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio recupera una raccolta di utenti nella cartella corrente. Account AWS

```
Get-IAMUserList
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Administrator
CreateDate       : 10/16/2014 9:03:09 AM
PasswordLastUsed : 3/4/2015 12:12:33 PM
Path             : /
UserId           : 7K3GJEANSKZF2EXAMPLE1
UserName         : Administrator

Arn              : arn:aws:iam::123456789012:user/Bob
CreateDate       : 4/6/2015 12:54:42 PM
PasswordLastUsed : 1/1/0001 12:00:00 AM
Path             : /
UserId           : L3EWNONDOM3YUEXAMPLE2
UserName         : bab

Arn              : arn:aws:iam::123456789012:user/David
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 3/19/2015 8:44:04 AM
Path             : /
UserId           : Y4FKWQCXTA52QEXAMPLE3
UserName         : David
```

- Per i dettagli sull'API, vedere ListUsersin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def list_users():
    """
    Lists the users in the current account.

    :return: The list of users.
    """
    try:
        users = list(iam.users.all())
        logger.info("Got %s users.", len(users))
    except ClientError:
        logger.exception("Couldn't get users.")
        raise
    else:
        return users
```

- Per i dettagli sull'API, consulta ListUsers AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
# Lists all users in the AWS account
#
# @return [Array<Aws::IAM::Types::User>] An array of user objects
def list_users
  users = []
  @iam_client.list_users.each_page do |page|
    page.users.each do |user|
      users << user
    end
  end
  users
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing users: #{e.message}")
  []
end
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
pub async fn list_users(
```

```
    client: &iamClient,
    path_prefix: Option<String>,
    marker: Option<String>,
    max_items: Option<i32>,
) -> Result<ListUsersOutput, SdkError<ListUsersError>> {
    let response = client
        .list_users()
        .set_path_prefix(path_prefix)
        .set_marker(marker)
        .set_max_items(max_items)
        .send()
        .await?;
    Ok(response)
}
```

- Per i dettagli sulle API, consulta la ListUsersguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_iam->listusers( ).
        MESSAGE 'Retrieved user list.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when listing users.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, ListUsersconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


    public func listUsers() async throws -> [MyUserRecord] {
        var userList: [MyUserRecord] = []

        // Use "Paginated" to get all the users.
        // This lets the SDK handle the 'isTruncated' in "ListUsersOutput".
        let input = ListUsersInput()
        let output = client.listUsersPaginated(input: input)

        do {
            for try await page in output {
                guard let users = page.users else {
                    continue
                }
                for user in users {
                    if let id = user.userId, let name = user.userName {
                        userList.append(MyUserRecord(id: id, name: name))
                    }
                }
            }
        }
        catch {
            print("ERROR: listUsers:", dump(error))
            throw error
        }
       return userList
    }
```

- Per i dettagli sull'API, consulta la ListUsersguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`ListVirtualMfaDevices`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ListVirtualMfaDevices`.

CLI

AWS CLI

Per elencare i dispositivi MFA virtuale

Il comando `list-virtual-mfa-devices` seguente elenca i dispositivi MFA virtuali che sono stati configurati per l'account corrente.

```
aws iam list-virtual-mfa-devices
```

Output:

```
{
    "VirtualMFADevices": [
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/ExampleMFADevice"
        },
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/Fred"
        }
    ]
}
```

Per ulteriori informazioni, consulta Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta ListVirtualMfaDevices AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio recupera una raccolta di dispositivi MFA virtuali assegnati agli utenti dell'account. AWS La proprietà **`User`** di ciascuno è un oggetto con i dettagli dell'utente IAM a cui è assegnato il dispositivo.

```
Get-IAMVirtualMFADevice -AssignmentStatus Assigned
```

Output:

```
Base32StringSeed :
EnableDate       : 4/13/2015 12:03:42 PM
QRCodePNG        :
SerialNumber     : arn:aws:iam::123456789012:mfa/David
User             : Amazon.IdentityManagement.Model.User

Base32StringSeed :
EnableDate       : 4/13/2015 12:06:41 PM
QRCodePNG        :
SerialNumber     : arn:aws:iam::123456789012:mfa/root-account-mfa-device
User             : Amazon.IdentityManagement.Model.User
```

- Per i dettagli sull'API, vedere ListVirtualMfaDevicesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio recupera una raccolta di dispositivi MFA virtuali assegnati agli utenti dell'account. AWS La proprietà **`User`** di ciascuno è un oggetto con i dettagli dell'utente IAM a cui è assegnato il dispositivo.

```
Get-IAMVirtualMFADevice -AssignmentStatus Assigned
```

Output:

```
Base32StringSeed :
EnableDate       : 4/13/2015 12:03:42 PM
QRCodePNG        :
SerialNumber     : arn:aws:iam::123456789012:mfa/David
```

```
User             : Amazon.IdentityManagement.Model.User

Base32StringSeed :
EnableDate       : 4/13/2015 12:06:41 PM
QRCodePNG        :
SerialNumber     : arn:aws:iam::123456789012:mfa/root-account-mfa-device
User             : Amazon.IdentityManagement.Model.User
```

- Per i dettagli sull'API, vedere ListVirtualMfaDevicesin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`PutGroupPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`PutGroupPolicy`.

CLI

AWS CLI

Come aggiungere una policy a un gruppo

Il comando `put-group-policy` seguente aggiunge una policy al gruppo IAM denominato `Admins`.

```
aws iam put-group-policy \
    --group-name Admins \
    --policy-document file://AdminPolicy.json \
    --policy-name AdminRoot
```

Questo comando non produce alcun output.

La policy è definita come un documento JSON nel AdminPolicy.jsonfile. (Il nome e l'estensione del file non hanno importanza.)

Per ulteriori informazioni, consulta Gestione delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta PutGroupPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea una policy in linea denominata **`AppTesterPolicy`** e la incorpora nel gruppo IAM **`AppTesters`**. Se esiste già una policy in linea con lo stesso nome, quest'ultima sarà sovrascritta. Il contenuto della policy JSON viene fornito nel file **`apptesterpolicy.json`**. Si noti che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro **`-Raw`**.

```
Write-IAMGroupPolicy -GroupName AppTesters -PolicyName AppTesterPolicy -
PolicyDocument (Get-Content -Raw apptesterpolicy.json)
```

- Per i dettagli sull'API, vedere PutGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea una policy in linea denominata **`AppTesterPolicy`** e la incorpora nel gruppo IAM **`AppTesters`**. Se esiste già una policy in linea con lo stesso nome, quest'ultima sarà sovrascritta. Il contenuto della policy JSON viene fornito nel file **`apptesterpolicy.json`**. Si noti che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro **`-Raw`**.

```
Write-IAMGroupPolicy -GroupName AppTesters -PolicyName AppTesterPolicy -
PolicyDocument (Get-Content -Raw apptesterpolicy.json)
```

- Per i dettagli sull'API, vedere PutGroupPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`PutRolePermissionsBoundary`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`PutRolePermissionsBoundary`.

CLI

AWS CLI

Esempio 1: applicare un limite delle autorizzazioni basato su una policy personalizzata a un ruolo IAM

L'esempio `put-role-permissions-boundary` seguente applica la policy personalizzata denominata `intern-boundary` come limite delle autorizzazioni per il ruolo IAM specificato.

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --role-name lambda-application-role
```

Questo comando non produce alcun output.

Esempio 2: applicare un limite di autorizzazioni basato su una policy AWS gestita a un ruolo IAM

L'`put-role-permissions-boundary`esempio seguente applica la `PowerUserAccess` policy AWS gestita come limite di autorizzazioni per il ruolo IAM specificato.

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --role-name x-account-admin
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta AWS CLI Command PutRolePermissionsBoundaryReference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio mostra come impostare il limite delle autorizzazioni per un ruolo IAM. È possibile impostare politiche AWS gestite o politiche personalizzate come limite di autorizzazione.

```
Set-IAMRolePermissionsBoundary -RoleName MyRoleName -PermissionsBoundary
 arn:aws:iam::123456789012:policy/intern-boundary
```

- Per i dettagli sull'API, vedere PutRolePermissionsBoundaryin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio mostra come impostare il limite delle autorizzazioni per un ruolo IAM. È possibile impostare politiche AWS gestite o politiche personalizzate come limite di autorizzazione.

```
Set-IAMRolePermissionsBoundary -RoleName MyRoleName -PermissionsBoundary
 arn:aws:iam::123456789012:policy/intern-boundary
```

- Per i dettagli sull'API, vedere PutRolePermissionsBoundaryin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`PutRolePolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`PutRolePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Nozioni di base sui data warehouse serverless
- Guida introduttiva alla gestione della configurazione
- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
- Guida introduttiva ai cluster di data warehouse con provisioning
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    /// <summary>
    /// Update the inline policy document embedded in a role.
    /// </summary>
    /// <param name="policyName">The name of the policy to embed.</param>
    /// <param name="roleName">The name of the role to update.</param>
    /// <param name="policyDocument">The policy document that defines the role.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutRolePolicyAsync(string policyName, string
 roleName, string policyDocument)
    {
        var request = new PutRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutRolePolicyAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, PutRolePolicyconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::putRolePolicy(
        const Aws::String &roleName,
        const Aws::String &policyName,
        const Aws::String &policyDocument,
        const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::IAM::IAMClient iamClient(clientConfig);
    Aws::IAM::Model::PutRolePolicyRequest request;

    request.SetRoleName(roleName);
    request.SetPolicyName(policyName);
    request.SetPolicyDocument(policyDocument);

    Aws::IAM::Model::PutRolePolicyOutcome outcome =
 iamClient.PutRolePolicy(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error putting policy on role. " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully put the role policy." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, PutRolePolicyconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come collegare una policy di autorizzazioni a un ruolo IAM

Il comando `put-role-policy` seguente aggiunge una policy di autorizzazioni al ruolo denominato `Test-Role`.

```
aws iam put-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

Questo comando non produce alcun output.

La policy è definita come documento JSON nel AdminPolicy.jsonfile. (Il nome e l'estensione del file non hanno importanza.)

Per collegare una policy di attendibilità a un ruolo, usa il comando `update-assume-role-policy`.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta PutRolePolicy AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { PutRolePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const examplePolicyDocument = JSON.stringify({
  Version: "2012-10-17",
  Statement: [
    {
```

```
      Sid: "VisualEditor0",
      Effect: "Allow",
      Action: [
        "s3:ListBucketMultipartUploads",
        "s3:ListBucketVersions",
        "s3:ListBucket",
        "s3:ListMultipartUploadParts",
      ],
      Resource: "arn:aws:s3:::amzn-s3-demo-bucket",
    },
    {
      Sid: "VisualEditor1",
      Effect: "Allow",
      Action: [
        "s3:ListStorageLensConfigurations",
        "s3:ListAccessPointsForObjectLambda",
        "s3:ListAllMyBuckets",
        "s3:ListAccessPoints",
        "s3:ListJobs",
        "s3:ListMultiRegionAccessPoints",
      ],
      Resource: "*",
    },
  ],
});

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 * @param {string} policyName
 * @param {string} policyDocument
 */
export const putRolePolicy = async (roleName, policyName, policyDocument) => {
  const command = new PutRolePolicyCommand({
    RoleName: roleName,
    PolicyName: policyName,
    PolicyDocument: policyDocument,
  });

  const response = await client.send(command);
  console.log(response);
  return response;
```

```
};
```

- Per i dettagli sull'API, PutRolePolicyconsulta AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea una policy in linea denominata **`FedTesterRolePolicy`** e la incorpora nel ruolo IAM **`FedTesterRole`**. Se esiste già una policy in linea con lo stesso nome, quest'ultima sarà sovrascritta. Il contenuto della policy JSON proviene dal file **`FedTesterPolicy.json`**. Si noti che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro **`-Raw`**.

```
Write-IAMRolePolicy -RoleName FedTesterRole -PolicyName FedTesterRolePolicy -
PolicyDocument (Get-Content -Raw FedTesterPolicy.json)
```

- Per i dettagli sull'API, vedere PutRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea una policy in linea denominata **`FedTesterRolePolicy`** e la incorpora nel ruolo IAM **`FedTesterRole`**. Se esiste già una policy in linea con lo stesso nome, quest'ultima sarà sovrascritta. Il contenuto della policy JSON proviene dal file **`FedTesterPolicy.json`**. Si noti che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro **`-Raw`**.

```
Write-IAMRolePolicy -RoleName FedTesterRole -PolicyName FedTesterRolePolicy -
PolicyDocument (Get-Content -Raw FedTesterPolicy.json)
```

- Per i dettagli sull'API, vedere PutRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **PutUserPermissionsBoundary** con una CLI

I seguenti esempi di codice mostrano come utilizzare`PutUserPermissionsBoundary`.

CLI

AWS CLI

Esempio 1: applicare un limite delle autorizzazioni basato su una policy personalizzata a un utente IAM

L'esempio `put-user-permissions-boundary` seguente applica una policy personalizzata denominata `intern-boundary` come limite delle autorizzazioni per l'utente IAM specificato.

```
aws iam put-user-permissions-boundary \
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --user-name intern
```

Questo comando non produce alcun output.

Esempio 2: applicare un limite di autorizzazioni basato su una policy AWS gestita a un utente IAM

L'`put-user-permissions-boundary`esempio seguente applica la policy AWS gestita `PowerUserAccess` denominata limite delle autorizzazioni per l'utente IAM specificato.

```
aws iam put-user-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --user-name developer
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta Command PutUserPermissionsBoundaryReference AWS CLI .

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio mostra come impostare il limite delle autorizzazioni per l'utente. È possibile impostare politiche AWS gestite o politiche personalizzate come limite di autorizzazione.

```
Set-IAMUserPermissionsBoundary -UserName joe -PermissionsBoundary
 arn:aws:iam::123456789012:policy/intern-boundary
```

- Per i dettagli sull'API, vedere PutUserPermissionsBoundaryin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio mostra come impostare il limite delle autorizzazioni per l'utente. È possibile impostare politiche AWS gestite o politiche personalizzate come limite di autorizzazione.

```
Set-IAMUserPermissionsBoundary -UserName joe -PermissionsBoundary
 arn:aws:iam::123456789012:policy/intern-boundary
```

- Per i dettagli sull'API, vedere PutUserPermissionsBoundaryin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`PutUserPolicy`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`PutUserPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

CLI

AWS CLI

Come collegare una policy a un utente IAM

Il comando `put-user-policy` seguente collega una policy all'utente IAM denominato Bob.

```
aws iam put-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

Questo comando non produce alcun output.

La policy è definita come un documento JSON nel AdminPolicy.jsonfile. (Il nome e l'estensione del file non hanno importanza.)

Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta PutUserPolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import (
 "context"
 "encoding/json"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/iam"
```

```
 "github.com/aws/aws-sdk-go-v2/service/iam/types"
 "github.com/aws/smithy-go"
)

// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// CreateUserPolicy adds an inline policy to a user. This example creates a
 policy that
// grants a list of actions on a specified role.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper UserWrapper) CreateUserPolicy(ctx context.Context, userName string,
 policyName string, actions []string,
 roleArn string) error {
 policyDoc := PolicyDocument{
  Version: "2012-10-17",
  Statement: []PolicyStatement{{
   Effect:   "Allow",
   Action:   actions,
   Resource: aws.String(roleArn),
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n", roleArn,
 err)
  return err
 }
 _, err = wrapper.IamClient.PutUserPolicy(ctx, &iam.PutUserPolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
  UserName:       aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create policy for user %v. Here's why: %v\n", userName,
 err)
 }
```

```
 return err
}
```

- Per i dettagli sull'API, consulta la PutUserPolicysezione AWS SDK per Go API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio crea una policy in linea denominata **EC2AccessPolicy** e la incorpora nell'utente IAM **Bob**. Se esiste già una policy in linea con lo stesso nome, quest'ultima sarà sovrascritta. Il contenuto della policy JSON proviene dal file **EC2AccessPolicy.json**. Si noti che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro **-Raw**.

```
Write-IAMUserPolicy -UserName Bob -PolicyName EC2AccessPolicy -PolicyDocument
 (Get-Content -Raw EC2AccessPolicy.json)
```

- Per i dettagli sull'API, vedere PutUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio crea una policy in linea denominata **EC2AccessPolicy** e la incorpora nell'utente IAM **Bob**. Se esiste già una policy in linea con lo stesso nome, quest'ultima sarà sovrascritta. Il contenuto della policy JSON proviene dal file **EC2AccessPolicy.json**. Si noti che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro **-Raw**.

```
Write-IAMUserPolicy -UserName Bob -PolicyName EC2AccessPolicy -PolicyDocument
 (Get-Content -Raw EC2AccessPolicy.json)
```

- Per i dettagli sull'API, vedere PutUserPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
# Creates an inline policy for a specified user.
# @param username [String] The name of the IAM user.
# @param policy_name [String] The name of the policy to create.
# @param policy_document [String] The JSON policy document.
# @return [Boolean]
def create_user_policy(username, policy_name, policy_document)
  @iam_client.put_user_policy({
                                user_name: username,
                                policy_name: policy_name,
                                policy_document: policy_document
                              })
  @logger.info("Policy #{policy_name} created for user #{username}.")
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Couldn't create policy #{policy_name} for user #{username}.
Here's why:")
  @logger.error("\t#{e.code}: #{e.message}")
  false
end
```

- Per i dettagli sull'API, consulta la PutUserPolicysezione AWS SDK per Ruby API Reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSIAM
import AWSS3


   func putUserPolicy(policyDocument: String, policyName: String, user:
 IAMClientTypes.User) async throws {
        let input = PutUserPolicyInput(
            policyDocument: policyDocument,
            policyName: policyName,
            userName: user.userName
        )
        do {
            _ = try await iamClient.putUserPolicy(input: input)
        } catch {
            print("ERROR: putUserPolicy:", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la PutUserPolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`RemoveClientIdFromOpenIdConnectProvider`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`RemoveClientIdFromOpenIdConnectProvider`.

CLI

AWS CLI

Per rimuovere l'ID client specificato dall'elenco degli ID client registrati per il provider OpenID Connect specificato per IAM

Questo esempio rimuove l'ID client `My-TestApp-3` dall'elenco degli ID client associati al provider OIDC per IAM il cui ARN è `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com`.

```
aws iam remove-client-id-from-open-id-connect-provider
    --client-id My-TestApp-3 \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Creazione di provider di identità OpenID Connect (OIDC) nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta RemoveClientIdFromOpenIdConnectProvider AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio rimuove l'ID client **`My-TestApp-3`** dall'elenco degli ID client associati al provider OIDC per IAM il cui ARN è **`arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com`**.

```
Remove-IAMClientIDFromOpenIDConnectProvider -ClientID My-TestApp-3
 -OpenIDConnectProviderArn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

- Per i dettagli sull'API, vedere RemoveClientIdFromOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio rimuove l'ID client **My-TestApp-3** dall'elenco degli ID client associati al provider OIDC per IAM il cui ARN è **arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com**.

```
Remove-IAMClientIDFromOpenIDConnectProvider -ClientID My-TestApp-3
 -OpenIDConnectProviderArn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

- Per i dettagli sull'API, vedere RemoveClientIdFromOpenIdConnectProviderin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **RemoveRoleFromInstanceProfile** con una CLI

I seguenti esempi di codice mostrano come utilizzare`RemoveRoleFromInstanceProfile`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Guida introduttiva allo streaming gestito
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

CLI

AWS CLI

Come rimuovere un ruolo da un profilo dell'istanza

Il comando `remove-role-from-instance-profile` seguente rimuove il ruolo denominato `Test-Role` dal profilo dell'istanza denominato `ExampleInstanceProfile`.

```
aws iam remove-role-from-instance-profile \
```

```
    --instance-profile-name ExampleInstanceProfile \
    --role-name Test-Role
```

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta RemoveRoleFromInstanceProfile AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio elimina il ruolo denominato **MyNewRole** dal profilo di istanza EC2 denominato **MyNewRole**. Un profilo dell'istanza creato nella console IAM ha sempre lo stesso nome del ruolo, come in questo esempio. Se li crei nell'API o nella CLI, possono avere nomi diversi.

```
Remove-IAMRoleFromInstanceProfile -InstanceProfileName MyNewRole -RoleName
 MyNewRole -Force
```

- Per i dettagli sull'API, vedere RemoveRoleFromInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio elimina il ruolo denominato **MyNewRole** dal profilo di istanza EC2 denominato **MyNewRole**. Un profilo dell'istanza creato nella console IAM ha sempre lo stesso nome del ruolo, come in questo esempio. Se li crei nell'API o nella CLI, possono avere nomi diversi.

```
Remove-IAMRoleFromInstanceProfile -InstanceProfileName MyNewRole -RoleName
 MyNewRole -Force
```

- Per i dettagli sull'API, vedere RemoveRoleFromInstanceProfilein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **RemoveUserFromGroup** con una CLI

I seguenti esempi di codice mostrano come utilizzare`RemoveUserFromGroup`.

CLI

AWS CLI

Come rimuovere un utente da un gruppo IAM

Il comando `remove-user-from-group` seguente rimuove l'utente denominato `Bob` dal gruppo IAM denominato `Admins`.

```
aws iam remove-user-from-group \
    --user-name Bob \
    --group-name Admins
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Aggiunta e rimozione di utenti in un gruppo di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta RemoveUserFromGroup AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio rimuove l'utente IAM **Bob** dal gruppo **Testers**.

```
Remove-IAMUserFromGroup -GroupName Testers -UserName Bob
```

Esempio 2: questo esempio trova tutti i gruppi di cui l'utente IAM **Theresa** è membro e quindi rimuove **Theresa** da tali gruppi.

```
$groups = Get-IAMGroupForUser -UserName Theresa
foreach ($group in $groups) { Remove-IAMUserFromGroup -GroupName $group.GroupName
 -UserName Theresa -Force }
```

Esempio 3: questo esempio mostra un modo alternativo per rimuovere l'utente IAM **Bob** dal gruppo **Testers**.

```
Get-IAMGroupForUser -UserName Bob | Remove-IAMUserFromGroup -UserName Bob -
GroupName Testers -Force
```

- Per i dettagli sull'API, vedere RemoveUserFromGroupin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio rimuove l'utente IAM **Bob** dal gruppo **Testers**.

```
Remove-IAMUserFromGroup -GroupName Testers -UserName Bob
```

Esempio 2: questo esempio trova tutti i gruppi di cui l'utente IAM **Theresa** è membro e quindi rimuove **Theresa** da tali gruppi.

```
$groups = Get-IAMGroupForUser -UserName Theresa
foreach ($group in $groups) { Remove-IAMUserFromGroup -GroupName $group.GroupName
 -UserName Theresa -Force }
```

Esempio 3: questo esempio mostra un modo alternativo per rimuovere l'utente IAM **Bob** dal gruppo **Testers**.

```
Get-IAMGroupForUser -UserName Bob | Remove-IAMUserFromGroup -UserName Bob -
GroupName Testers -Force
```

- Per i dettagli sull'API, vedere RemoveUserFromGroupin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **ResyncMfaDevice** con una CLI

I seguenti esempi di codice mostrano come utilizzare`ResyncMfaDevice`.

CLI

AWS CLI

Per sincronizzare un dispositivo MFA

L'esempio `resync-mfa-device` seguente sincronizza il dispositivo MFA associato all'utente IAM Bob e il cui ARN è `arn:aws:iam::123456789012:mfa/BobsMFADevice` con un programma di autenticazione che ha fornito i due codici di autenticazione.

```
aws iam resync-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
    --authentication-code2 987654
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella AWS Guida per l'utente IAM.

- Per i dettagli sull'API, consulta ResyncMfaDevice AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio sincronizza il dispositivo MFA associato all'utente IAM **Bob** e il cui ARN è **arn:aws:iam::123456789012:mfa/bob** con un programma di autenticazione che ha fornito i due codici di autenticazione.

```
Sync-IAMMFADevice -SerialNumber arn:aws:iam::123456789012:mfa/theresa -
AuthenticationCode1 123456 -AuthenticationCode2 987654 -UserName Bob
```

Esempio 2: questo esempio sincronizza il dispositivo MFA IAM associato all'utente IAM **Theresa** con un dispositivo fisico che ha il numero di serie **ABCD12345678** e che ha fornito i due codici di autenticazione.

```
Sync-IAMMFADevice -SerialNumber ABCD12345678 -AuthenticationCode1 123456 -
AuthenticationCode2 987654 -UserName Theresa
```

- Per i dettagli sull'API, vedere ResyncMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio sincronizza il dispositivo MFA associato all'utente IAM **Bob** e il cui ARN è **arn:aws:iam::123456789012:mfa/bob** con un programma di autenticazione che ha fornito i due codici di autenticazione.

```
Sync-IAMMFADevice -SerialNumber arn:aws:iam::123456789012:mfa/theresa -
AuthenticationCode1 123456 -AuthenticationCode2 987654 -UserName Bob
```

Esempio 2: questo esempio sincronizza il dispositivo MFA IAM associato all'utente IAM **Theresa** con un dispositivo fisico che ha il numero di serie **ABCD12345678** e che ha fornito i due codici di autenticazione.

```
Sync-IAMMFADevice -SerialNumber ABCD12345678 -AuthenticationCode1 123456 -
AuthenticationCode2 987654 -UserName Theresa
```

- Per i dettagli sull'API, vedere ResyncMfaDevicein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **SetDefaultPolicyVersion**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`SetDefaultPolicyVersion`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Gestione delle policy
- Rollback di una versione della policy

CLI

AWS CLI

Come impostare la versione indicata della policy specificata come versione predefinita della policy.

Questo esempio imposta la versione v2 della policy il cui ARN è `arn:aws:iam::123456789012:policy/MyPolicy` come versione attiva predefinita.

```
aws iam set-default-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta SetDefaultPolicyVersion AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio imposta la versione **v2** della policy il cui ARN è **arn:aws:iam::123456789012:policy/MyPolicy** come versione attiva predefinita.

```
Set-IAMDefaultPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/MyPolicy
 -VersionId v2
```

- Per i dettagli sull'API, vedere SetDefaultPolicyVersionin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio imposta la versione **v2** della policy il cui ARN è **arn:aws:iam::123456789012:policy/MyPolicy** come versione attiva predefinita.

```
Set-IAMDefaultPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/MyPolicy
 -VersionId v2
```

- Per i dettagli sull'API, vedere SetDefaultPolicyVersionin AWS Strumenti per PowerShell Cmdlet Reference (V5).

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
TRY.
    lo_iam->setdefaultpolicyversion(
      iv_policyarn = iv_policy_arn
      iv_versionid = iv_version_id ).
    MESSAGE 'Default policy version set successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy or version does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iaminvalidinputex.
    MESSAGE 'Invalid input provided.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, SetDefaultPolicyVersionconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`TagRole`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`TagRole`.

CLI

AWS CLI

Per aggiungere un tag a un ruolo

Il comando `tag-role` seguente aggiunge un tag con un nome di reparto al ruolo specificato.

```
aws iam tag-role --role-name my-role \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Applicazione di tag a risorse IAM nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta TagRole AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio aggiunge un tag a un ruolo nel servizio di gestione identità

```
Add-IAMRoleTag -RoleName AdminRoleacess -Tag @{ Key = 'abac'; Value = 'testing'}
```

- Per i dettagli sull'API, vedere TagRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio aggiunge un tag a un ruolo nel servizio di gestione identità

```
Add-IAMRoleTag -RoleName AdminRoleacess -Tag @{ Key = 'abac'; Value = 'testing'}
```

- Per i dettagli sull'API, vedere TagRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`TagUser`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`TagUser`.

CLI

AWS CLI

Per aggiungere un tag a un utente

Il comando `tag-user` seguente aggiunge un tag con il reparto associato all'utente specificato.

```
aws iam tag-user \
    --user-name alice \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Applicazione di tag a risorse IAM nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta TagUser AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio aggiunge un tag a un utente nel servizio di gestione identità

```
Add-IAMUserTag -UserName joe -Tag @{ Key = 'abac'; Value = 'testing'}
```

- Per i dettagli sull'API, vedere TagUserin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio aggiunge un tag a un utente nel servizio di gestione identità

```
Add-IAMUserTag -UserName joe -Tag @{ Key = 'abac'; Value = 'testing'}
```

- Per i dettagli sull'API, vedere TagUserin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UntagRole`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UntagRole`.

CLI

AWS CLI

Per rimuovere un tag da un ruolo

Il comando `untag-role` seguente rimuove qualsiasi tag con il nome chiave 'Department' dal ruolo specificato.

```
aws iam untag-role \
    --role-name my-role \
    --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Applicazione di tag a risorse IAM nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta UntagRole AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio rimuove il tag dal ruolo denominato "MyRoleName" con la chiave del tag come «abac». Per rimuovere più tag, fornisci un elenco di chiavi di tag separate da virgole.

```
Remove-IAMRoleTag -RoleName MyRoleName -TagKey "abac","xyzw"
```

- Per i dettagli sull'API, vedere UntagRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio rimuove il tag dal ruolo denominato "MyRoleName" con la chiave del tag come «abac». Per rimuovere più tag, fornisci un elenco di chiavi di tag separate da virgole.

```
Remove-IAMRoleTag -RoleName MyRoleName -TagKey "abac","xyzw"
```

- Per i dettagli sull'API, vedere UntagRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UntagUser`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UntagUser`.

CLI

AWS CLI

Per rimuovere un tag da un utente

Il comando `untag-user` seguente rimuove qualsiasi tag con il nome chiave 'Department' dall'utente specificato.

```
aws iam untag-user \
    --user-name alice \
    --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Applicazione di tag a risorse IAM nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta UntagUser AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio rimuove il tag dall'utente denominato "joe" con la chiave di tag "abac" e "xyzw". Per rimuovere più tag, fornisci un elenco di chiavi di tag separate da virgole.

```
Remove-IAMUserTag -UserName joe -TagKey "abac","xyzw"
```

- Per i dettagli sull'API, vedere UntagUserin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio rimuove il tag dall'utente denominato "joe" con la chiave di tag "abac" e "xyzw". Per rimuovere più tag, fornisci un elenco di chiavi di tag separate da virgole.

```
Remove-IAMUserTag -UserName joe -TagKey "abac","xyzw"
```

- Per i dettagli sull'API, vedere UntagUserin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`UpdateAccessKey`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateAccessKey`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestione delle chiavi di accesso

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
#############################################################################
# function iam_update_access_key
#
```

```
# This function can activate or deactivate an IAM access key for the specified
 IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to update.
#       -a            -- Activate the selected access key.
#       -d            -- Deactivate the selected access key.
#
# Example:
#       # To deactivate the selected access key for IAM user Bob
#       iam_update_access_key -u Bob -k AKIAIOSFODNN7EXAMPLE -d
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_update_access_key() {
  local user_name access_key status response
  local option OPTARG # Required to use getopts command in a function.
  local activate_flag=false deactivate_flag=false

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_update_access_key"
    echo "Updates the status of an AWS Identity and Access Management (IAM)
 access key for the specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to update."
    echo "  -a              Activate the access key."
    echo "  -d              Deactivate the access key."
    echo ""
  }

  # Retrieve the calling parameters.
    while getopts "u:k:adh" option; do
      case "${option}" in
        u) user_name="${OPTARG}" ;;
        k) access_key="${OPTARG}" ;;
        a) activate_flag=true ;;
        d) deactivate_flag=true ;;
        h)
          usage
          return 0
```

```
          ;;
        \?)
          echo "Invalid parameter"
          usage
          return 1
          ;;
      esac
    done
    export OPTIND=1

  # Validate input parameters
    if [[ -z "$user_name" ]]; then
      errecho "ERROR: You must provide a username with the -u parameter."
      usage
      return 1
    fi

    if [[ -z "$access_key" ]]; then
      errecho "ERROR: You must provide an access key with the -k parameter."
      usage
      return 1
    fi

    # Ensure that only -a or -d is specified
    if [[ "$activate_flag" == true && "$deactivate_flag" == true ]]; then
      errecho "ERROR: You cannot specify both -a (activate) and -d (deactivate)
at the same time."
      usage
      return 1
    fi

    # If neither -a nor -d is provided, return an error
    if [[ "$activate_flag" == false && "$deactivate_flag" == false ]]; then
      errecho "ERROR: You must specify either -a (activate) or -d (deactivate)."
      usage
      return 1
    fi

    # Determine the status based on the flag
    if [[ "$activate_flag" == true ]]; then
      status="Active"
    elif [[ "$deactivate_flag" == true ]]; then
      status="Inactive"
    fi
```

```
    iecho "Parameters:\n"
    iecho "    Username:   $user_name"
    iecho "    Access key: $access_key"
    iecho "    New status: $status"
    iecho ""

    # Update the access key status
    response=$(aws iam update-access-key \
      --user-name "$user_name" \
      --access-key-id "$access_key" \
      --status "$status" 2>&1)

    local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
      aws_cli_error_log $error_code
      errecho "ERROR: AWS reports update-access-key operation failed.\n$response"
      return 1
    fi

    iecho "update-access-key response: $response"
    iecho

    return 0
}
```

- Per i dettagli sull'API, consulta UpdateAccessKey AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::updateAccessKey(const Aws::String &userName,
                                  const Aws::String &accessKeyID,
```

```
                                    Aws::IAM::Model::StatusType status,
                                    const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::UpdateAccessKeyRequest request;
    request.SetUserName(userName);
    request.SetAccessKeyId(accessKeyID);
    request.SetStatus(status);

    auto outcome = iam.UpdateAccessKey(request);
    if (outcome.IsSuccess()) {
        std::cout << "Successfully updated status of access key "
                  << accessKeyID << " for user " << userName << std::endl;
    }
    else {
        std::cerr << "Error updated status of access key " << accessKeyID <<
                  " for user " << userName << ": " <<
                  outcome.GetError().GetMessage() << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, UpdateAccessKeyconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come attivare o disattivare una chiave di accesso per un utente IAM

Il comando `update-access-key` seguente disattiva la chiave di accesso specificata (ID chiave di accesso e chiave di accesso segreta) per l'utente IAM denominato Bob.

```
aws iam update-access-key \
    --access-key-id AKIAIOSFODNN7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

Questo comando non produce alcun output.

La disattivazione della chiave significa che non può essere utilizzata per l'accesso programmatico a. AWS La chiave, tuttavia, rimane disponibile e può essere riattivata.

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta Command UpdateAccessKeyReference AWS CLI .

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.StatusType;
import software.amazon.awssdk.services.iam.model.UpdateAccessKeyRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateAccessKey {

    private static StatusType statusType;

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <username> <accessId> <status>\s
```

```
                Where:
                    username - The name of the user whose key you want to update.
\s
                    accessId - The access key ID of the secret access key you
 want to update.\s
                    status - The status you want to assign to the secret access
 key.\s
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String username = args[0];
        String accessId = args[1];
        String status = args[2];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        updateKey(iam, username, accessId, status);
        System.out.println("Done");
        iam.close();
    }

    public static void updateKey(IamClient iam, String username, String accessId,
 String status) {
        try {
            if (status.toLowerCase().equalsIgnoreCase("active")) {
                statusType = StatusType.ACTIVE;
            } else if (status.toLowerCase().equalsIgnoreCase("inactive")) {
                statusType = StatusType.INACTIVE;
            } else {
                statusType = StatusType.UNKNOWN_TO_SDK_VERSION;
            }

            UpdateAccessKeyRequest request = UpdateAccessKeyRequest.builder()
                    .accessKeyId(accessId)
                    .userName(username)
                    .status(statusType)
                    .build();
```

```
            iam.updateAccessKey(request);
            System.out.printf("Successfully updated the status of access key %s
 to" +
                    "status %s for user %s", accessId, status, username);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, UpdateAccessKeyconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Aggiorna la chiave di accesso.

```
import {
  UpdateAccessKeyCommand,
  IAMClient,
  StatusType,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} userName
 * @param {string} accessKeyId
 */
export const updateAccessKey = (userName, accessKeyId) => {
```

```
  const command = new UpdateAccessKeyCommand({
    AccessKeyId: accessKeyId,
    Status: StatusType.Inactive,
    UserName: userName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, UpdateAccessKeyconsulta AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  AccessKeyId: "ACCESS_KEY_ID",
  Status: "Active",
  UserName: "USER_NAME",
};

iam.updateAccessKey(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

```
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, UpdateAccessKeyconsulta AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio modifica lo stato della chiave di accesso **`AKIAIOSFODNN7EXAMPLE`** per l'utente IAM denominato **`Bob`** in **`Inactive`**.

```
Update-IAMAccessKey -UserName Bob -AccessKeyId AKIAIOSFODNN7EXAMPLE -Status
 Inactive
```

- Per i dettagli sull'API, vedere UpdateAccessKeyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio modifica lo stato della chiave di accesso **`AKIAIOSFODNN7EXAMPLE`** per l'utente IAM denominato **`Bob`** in **`Inactive`**.

```
Update-IAMAccessKey -UserName Bob -AccessKeyId AKIAIOSFODNN7EXAMPLE -Status
 Inactive
```

- Per i dettagli sull'API, vedere UpdateAccessKeyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def update_key(user_name, key_id, activate):
```

```
    """
    Updates the status of a key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to update.
    :param activate: When True, the key is activated. Otherwise, the key is
deactivated.
    """

    try:
        key = iam.User(user_name).AccessKey(key_id)
        if activate:
            key.activate()
        else:
            key.deactivate()
        logger.info("%s key %s.", "Activated" if activate else "Deactivated",
key_id)
    except ClientError:
        logger.exception(
            "Couldn't %s key %s.", "Activate" if activate else "Deactivate",
key_id
        )
        raise
```

- Per i dettagli sull'API, consulta UpdateAccessKey AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
```

```
        lo_iam->updateaccesskey(
          iv_accesskeyid = iv_access_key_id
          iv_status = iv_status
          iv_username = iv_user_name ).
        MESSAGE 'Access key updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Access key or user does not exist.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, UpdateAccessKeyconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UpdateAccountPasswordPolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateAccountPasswordPolicy`.

CLI

AWS CLI

Per impostare o modificare la policy delle password dell'account corrente

Il comando `update-account-password-policy` seguente imposta la policy delle password in modo che richieda una lunghezza minima di otto caratteri e uno o più numeri nella password.

```
aws iam update-account-password-policy \
    --minimum-password-length 8 \
    --require-numbers
```

Questo comando non produce alcun output.

Le modifiche alla policy delle password di un account influiscono su tutte le nuove password create per gli utenti IAM nell'account. Le modifiche alle policy delle password non influiscono sulle password esistenti.

Per ulteriori informazioni, consulta Impostazione di una policy delle password dell'account per utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta UpdateAccountPasswordPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio aggiorna la policy delle password per l'account con le impostazioni specificate. Tenere presente che tutti i parametri non inclusi nel comando non vengono lasciati invariati. Vengono invece ripristinati ai valori predefiniti.

```
Update-IAMAccountPasswordPolicy -AllowUsersToChangePasswords $true -HardExpiry
 $false -MaxPasswordAge 90 -MinimumPasswordLength 8 -PasswordReusePrevention 20
 -RequireLowercaseCharacters $true -RequireNumbers $true -RequireSymbols $true -
RequireUppercaseCharacters $true
```

- Per i dettagli sull'API, vedere UpdateAccountPasswordPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio aggiorna la policy delle password per l'account con le impostazioni specificate. Tenere presente che tutti i parametri non inclusi nel comando non vengono lasciati invariati. Vengono invece ripristinati ai valori predefiniti.

```
Update-IAMAccountPasswordPolicy -AllowUsersToChangePasswords $true -HardExpiry
 $false -MaxPasswordAge 90 -MinimumPasswordLength 8 -PasswordReusePrevention 20
 -RequireLowercaseCharacters $true -RequireNumbers $true -RequireSymbols $true -
RequireUppercaseCharacters $true
```

- Per i dettagli sull'API, vedere UpdateAccountPasswordPolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UpdateAssumeRolePolicy`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateAssumeRolePolicy`.

CLI

AWS CLI

Per aggiornare la policy di attendibilità di un ruolo IAM

Il comando `update-assume-role-policy` seguente aggiorna la policy di attendibilità per il ruolo denominato `Test-Role`.

```
aws iam update-assume-role-policy \
    --role-name Test-Role \
    --policy-document file://Test-Role-Trust-Policy.json
```

Questo comando non produce alcun output.

La politica di fiducia è definita come un documento JSON nel Test-Role-Trust-Policy.jsonfile. (Il nome e l'estensione del file non hanno importanza.) La policy di attendibilità deve specificare un principale.

Per aggiornare una policy di autorizzazioni per un ruolo, usa il comando `put-role-policy`.

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta UpdateAssumeRolePolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio aggiorna il ruolo IAM denominato **`ClientRole`** con una nuova policy di attendibilità, il cui contenuto proviene dal file **`ClientRolePolicy.json`**. Tenere presente che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro di cambio **`-Raw`**.

```
Update-IAMAssumeRolePolicy -RoleName ClientRole -PolicyDocument (Get-Content -raw
 ClientRolePolicy.json)
```

- Per i dettagli sull'API, vedere UpdateAssumeRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio aggiorna il ruolo IAM denominato **`ClientRole`** con una nuova policy di attendibilità, il cui contenuto proviene dal file **`ClientRolePolicy.json`**. Tenere presente che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro di cambio **`-Raw`**.

```
Update-IAMAssumeRolePolicy -RoleName ClientRole -PolicyDocument (Get-Content -raw
 ClientRolePolicy.json)
```

- Per i dettagli sull'API, vedere UpdateAssumeRolePolicyin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UpdateGroup`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateGroup`.

CLI

AWS CLI

Per rinominare un gruppo IAM

Il comando `update-group` seguente modifica il nome del gruppo IAM da `Test` a `Test-1`.

```
aws iam update-group \
    --group-name Test \
    --new-group-name Test-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Ridenominazione di un gruppo di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta UpdateGroup AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio rinomina il gruppo IAM **`Testers`** in **`AppTesters`**.

```
Update-IAMGroup -GroupName Testers -NewGroupName AppTesters
```

Esempio 2: questo esempio modifica il percorso del gruppo IAM **`AppTesters`** in **`/Org1/Org2/`**. Questo modifica l'ARN del gruppo in **`arn:aws:iam::123456789012:group/Org1/Org2/AppTesters`**.

```
Update-IAMGroup -GroupName AppTesters -NewPath /Org1/Org2/
```

- Per i dettagli sull'API, vedere UpdateGroupin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio rinomina il gruppo IAM **`Testers`** in **`AppTesters`**.

```
Update-IAMGroup -GroupName Testers -NewGroupName AppTesters
```

Esempio 2: questo esempio modifica il percorso del gruppo IAM **`AppTesters`** in **`/Org1/Org2/`**. Questo modifica l'ARN del gruppo in **`arn:aws:iam::123456789012:group/Org1/Org2/AppTesters`**.

```
Update-IAMGroup -GroupName AppTesters -NewPath /Org1/Org2/
```

- Per i dettagli sull'API, vedere UpdateGroupin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UpdateLoginProfile`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateLoginProfile`.

CLI

AWS CLI

Per aggiornare la password per un utente IAM

Il comando `update-login-profile` seguente crea una nuova password per l'utente IAM denominato Bob.

```
aws iam update-login-profile \
    --user-name Bob \
    --password <password>
```

Questo comando non produce alcun output.

Per impostare una policy delle password per l'account, usa il comando `update-account-password-policy`. Se la nuova password viola la policy delle password dell'account, il comando restituisce un errore `PasswordPolicyViolation`.

Se la policy delle password dell'account lo consente, gli utenti IAM possono modificare le proprie password utilizzando il comando `change-password`.

Conserva la nuova password in un luogo sicuro. Se la password viene persa, non può essere recuperata e dovrai crearne una nuova utilizzando il comando `create-login-profile`.

Per ulteriori informazioni, consulta Gestione delle password per gli utenti IAM nella Guida per l'utente di AWS .

- Per i dettagli sull'API, consulta UpdateLoginProfile AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio imposta una nuova password temporanea per l'utente IAM **Bob** e richiede all'utente di modificare la password al successivo accesso.

```
Update-IAMLoginProfile -UserName Bob -Password "P@ssw0rd1234" -
PasswordResetRequired $true
```

- Per i dettagli sull'API, vedere UpdateLoginProfilein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio imposta una nuova password temporanea per l'utente IAM **Bob** e richiede all'utente di modificare la password al successivo accesso.

```
Update-IAMLoginProfile -UserName Bob -Password "P@ssw0rd1234" -
PasswordResetRequired $true
```

- Per i dettagli sull'API, vedere UpdateLoginProfilein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UpdateOpenIdConnectProviderThumbprint`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateOpenIdConnectProviderThumbprint`.

CLI

AWS CLI

Per sostituire l'elenco esistente di impronte digitali dei certificati server con un nuovo elenco

Questo esempio aggiorna l'elenco delle impronte digitali dei certificati per il provider OIDC il cui ARN è `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com` per utilizzare una nuova impronta digitale.

```
aws iam update-open-id-connect-provider-thumbprint \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com \
    --thumbprint-list 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

Se gli endpoint discovery e JWKS del tuo provider OIDC utilizzano certificati diversi, includi le impronte digitali per entrambi gli endpoint nel valore. `--thumbprint-list`

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Creazione di provider di identità OpenID Connect (OIDC) nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta Command Reference. UpdateOpenIdConnectProviderThumbprintAWS CLI

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio aggiorna l'elenco delle impronte digitali dei certificati per il provider OIDC il cui ARN è **arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com** per utilizzare una nuova impronta digitale. Il provider OIDC condivide il nuovo valore quando il certificato associato al provider cambia.

```
Update-IAMOpenIDConnectProviderThumbprint -OpenIDConnectProviderArn
 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com -ThumbprintList
 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

- Per i dettagli sull'API, vedere UpdateOpenIdConnectProviderThumbprintin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio aggiorna l'elenco delle impronte digitali dei certificati per il provider OIDC il cui ARN è **arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com** per utilizzare una nuova impronta digitale. Il provider OIDC condivide il nuovo valore quando il certificato associato al provider cambia.

```
Update-IAMOpenIDConnectProviderThumbprint -OpenIDConnectProviderArn
 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com -ThumbprintList
 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

- Per i dettagli sull'API, vedere UpdateOpenIdConnectProviderThumbprintin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UpdateRole`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateRole`.

CLI

AWS CLI

Per modificare la descrizione o la durata della sessione di un ruolo IAM

Il comando `update-role` seguente modifica la descrizione del ruolo IAM `production-role` in `Main production role` e imposta la durata massima della sessione su 12 ore.

```
aws iam update-role \
    --role-name production-role \
    --description 'Main production role' \
    --max-session-duration 43200
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta UpdateRole AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio aggiorna la descrizione del ruolo e il valore della durata massima della sessione (in secondi) per cui è possibile richiedere la sessione di un ruolo.

```
Update-IAMRole -RoleName MyRoleName -Description "My testing role" -
MaxSessionDuration 43200
```

- Per i dettagli sull'API, vedere UpdateRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio aggiorna la descrizione del ruolo e il valore della durata massima della sessione (in secondi) per cui è possibile richiedere la sessione di un ruolo.

```
Update-IAMRole -RoleName MyRoleName -Description "My testing role" -
MaxSessionDuration 43200
```

- Per i dettagli sull'API, vedere UpdateRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UpdateRoleDescription`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateRoleDescription`.

CLI

AWS CLI

Per modificare la descrizione di un ruolo IAM

Il comando `update-role` seguente modifica la descrizione del ruolo IAM da `production-role` a `Main production role`.

```
aws iam update-role-description \
    --role-name production-role \
    --description 'Main production role'
```

Output:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "production-role",
        "RoleId": "AROA1234567890EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/production-role",
        "CreateDate": "2017-12-06T17:16:37+00:00",
        "AssumeRolePolicyDocument": {
            "Version":"2012-10-17",
            "Statement": [
                {
```

```
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
                    "Action": "sts:AssumeRole",
                    "Condition": {}
                }
            ]
        },
        "Description": "Main production role"
    }
}
```

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta UpdateRoleDescription AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio aggiorna la descrizione di un ruolo IAM nel tuo account.

```
Update-IAMRoleDescription -RoleName MyRoleName -Description "My testing role"
```

- Per i dettagli sull'API, vedere UpdateRoleDescriptionin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio aggiorna la descrizione di un ruolo IAM nel tuo account.

```
Update-IAMRoleDescription -RoleName MyRoleName -Description "My testing role"
```

- Per i dettagli sull'API, vedere UpdateRoleDescriptionin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UpdateSamlProvider`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateSamlProvider`.

CLI

AWS CLI

Per aggiornare il documento di metadati per un provider SAML esistente

Questo esempio aggiorna il provider SAML in IAM il cui ARN è `arn:aws:iam::123456789012:saml-provider/SAMLADFS` con un nuovo documento di metadati SAML dal file `SAMLMetaData.xml`.

```
aws iam update-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Output:

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/SAMLADFS"
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta UpdateSamlProvider AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio aggiorna il provider SAML in IAM il cui ARN è **`arn:aws:iam::123456789012:saml-provider/SAMLADFS`** con un nuovo documento di metadati SAML dal file **`SAMLMetaData.xml`**. Tenere presente che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro di cambio **`-Raw`**.

```
Update-IAMSAMLProvider -SAMLProviderArn arn:aws:iam::123456789012:saml-provider/
SAMLADFS -SAMLMetadataDocument (Get-Content -Raw SAMLMetaData.xml)
```

- Per i dettagli sull'API, vedere UpdateSamlProviderin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio aggiorna il provider SAML in IAM il cui ARN è **`arn:aws:iam::123456789012:saml-provider/SAMLADFS`** con un nuovo documento di metadati SAML dal file **`SAMLMetaData.xml`**. Tenere presente che per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro di cambio **`-Raw`**.

```
Update-IAMSAMLProvider -SAMLProviderArn arn:aws:iam::123456789012:saml-provider/
SAMLADFS -SAMLMetadataDocument (Get-Content -Raw SAMLMetaData.xml)
```

- Per i dettagli sull'API, vedere UpdateSamlProviderin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`UpdateServerCertificate`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateServerCertificate`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::updateServerCertificate(const Aws::String
 &currentCertificateName,
                                           const Aws::String &newCertificateName,
                                           const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
```

```
    Aws::IAM::Model::UpdateServerCertificateRequest request;
    request.SetServerCertificateName(currentCertificateName);
    request.SetNewServerCertificateName(newCertificateName);

    auto outcome = iam.UpdateServerCertificate(request);
    bool result = true;
    if (outcome.IsSuccess()) {
        std::cout << "Server certificate " << currentCertificateName
                  << " successfully renamed as " << newCertificateName
                  << std::endl;
    }
    else {
        if (outcome.GetError().GetErrorType() !=
 Aws::IAM::IAMErrors::NO_SUCH_ENTITY) {
            std::cerr << "Error changing name of server certificate " <<
                      currentCertificateName << " to " << newCertificateName <<
 ":" <<
                      outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Certificate '" << currentCertificateName
                      << "' not found." << std::endl;
        }
    }

    return result;
}
```

- Per i dettagli sull'API, consulta la UpdateServerCertificatesezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Per modificare il percorso o il nome di un certificato del server nel tuo AWS account

Il comando `update-server-certificate` seguente modifica il nome del certificato da `myServerCertificate` a `myUpdatedServerCertificate`. Cambia anche il percorso `/cloudfront/` in modo che sia accessibile dal CloudFront servizio Amazon. Questo comando

non produce alcun output. Puoi visualizzare i risultati dell'aggiornamento eseguendo il comando `list-server-certificates`.

```
aws-iam update-server-certificate \
    --server-certificate-name myServerCertificate \
    --new-server-certificate-name myUpdatedServerCertificate \
    --new-path /cloudfront/
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Gestione dei certificati server in IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta UpdateServerCertificate AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Aggiorna un certificato del server.

```
import { UpdateServerCertificateCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} currentName
 * @param {string} newName
 */
export const updateServerCertificate = (currentName, newName) => {
  const command = new UpdateServerCertificateCommand({
    ServerCertificateName: currentName,
    NewServerCertificateName: newName,
  });
```

```
  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la UpdateServerCertificatesezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  ServerCertificateName: "CERTIFICATE_NAME",
  NewServerCertificateName: "NEW_CERTIFICATE_NAME",
};

iam.updateServerCertificate(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la UpdateServerCertificatesezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio rinomina il certificato denominato **MyServerCertificate** in **MyRenamedServerCertificate**.

```
Update-IAMServerCertificate -ServerCertificateName MyServerCertificate -
NewServerCertificateName MyRenamedServerCertificate
```

Esempio 2: Questo esempio sposta il certificato denominato **MyServerCertificate** nel percorso/Org1/Org2/. Questo modifica l'ARN della risorsa in **arn:aws:iam::123456789012:server-certificate/Org1/Org2/MyServerCertificate**.

```
Update-IAMServerCertificate -ServerCertificateName MyServerCertificate -NewPath /
Org1/Org2/
```

- Per i dettagli sull'API, vedere UpdateServerCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio rinomina il certificato denominato **MyServerCertificate** in **MyRenamedServerCertificate**.

```
Update-IAMServerCertificate -ServerCertificateName MyServerCertificate -
NewServerCertificateName MyRenamedServerCertificate
```

Esempio 2: Questo esempio sposta il certificato denominato **MyServerCertificate** nel percorso/Org1/Org2/. Questo modifica l'ARN della risorsa in **arn:aws:iam::123456789012:server-certificate/Org1/Org2/MyServerCertificate**.

```
Update-IAMServerCertificate -ServerCertificateName MyServerCertificate -NewPath /
Org1/Org2/
```

- Per i dettagli sull'API, vedere UpdateServerCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Elenca, aggiorna ed elimina i certificati server.

```
class ServerCertificateManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = 'ServerCertificateManager'
  end

  # Creates a new server certificate.
  # @param name [String] the name of the server certificate
  # @param certificate_body [String] the contents of the certificate
  # @param private_key [String] the private key contents
  # @return [Boolean] returns true if the certificate was successfully created
  def create_server_certificate(name, certificate_body, private_key)
    @iam_client.upload_server_certificate({
                                            server_certificate_name: name,
                                            certificate_body: certificate_body,
                                            private_key: private_key
                                          })
    true
  rescue Aws::IAM::Errors::ServiceError => e
    puts "Failed to create server certificate: #{e.message}"
    false
  end

  # Lists available server certificate names.
  def list_server_certificate_names
    response = @iam_client.list_server_certificates

    if response.server_certificate_metadata_list.empty?
      @logger.info('No server certificates found.')
      return
```

```
    end

    response.server_certificate_metadata_list.each do |certificate_metadata|
      @logger.info("Certificate Name:
 #{certificate_metadata.server_certificate_name}")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing server certificates: #{e.message}")
  end

  # Updates the name of a server certificate.
  def update_server_certificate_name(current_name, new_name)
    @iam_client.update_server_certificate(
      server_certificate_name: current_name,
      new_server_certificate_name: new_name
    )
    @logger.info("Server certificate name updated from '#{current_name}' to
 '#{new_name}'.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error updating server certificate name: #{e.message}")
    false
  end

  # Deletes a server certificate.
  def delete_server_certificate(name)
    @iam_client.delete_server_certificate(server_certificate_name: name)
    @logger.info("Server certificate '#{name}' deleted.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting server certificate: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta la UpdateServerCertificatesezione AWS SDK per Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UpdateSigningCertificate`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateSigningCertificate`.

CLI

AWS CLI

Per attivare o disattivare un certificato di firma per un utente IAM

Il comando `update-signing-certificate` seguente disattiva il certificato di firma specificato per l'utente IAM denominato Bob.

```
aws iam update-signing-certificate \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

Per ottenere l'ID per un certificato di firma, utilizza il comando `list-signing-certificates`.

Per ulteriori informazioni, consulta Gestione dei certificati di firma nella Guida per l'utente di Amazon EC2.

- Per i dettagli sull'API, consulta UpdateSigningCertificate AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio aggiorna il certificato associato all'utente IAM denominato **Bob** e il cui ID certificato per contrassegnarlo come inattivo è **Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU**.

```
Update-IAMSigningCertificate -CertificateId Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU -
UserName Bob -Status Inactive
```

- Per i dettagli sull'API, vedere UpdateSigningCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio aggiorna il certificato associato all'utente IAM denominato **Bob** e il cui ID certificato per contrassegnarlo come inattivo è **Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU**.

```
Update-IAMSigningCertificate -CertificateId Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU -
UserName Bob -Status Inactive
```

- Per i dettagli sull'API, vedere UpdateSigningCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`UpdateUser`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateUser`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Creazione di utenti di sola lettura e di lettura e scrittura

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::IAM::updateUser(const Aws::String &currentUserName,
                             const Aws::String &newUserName,
                             const Aws::Client::ClientConfiguration
 &clientConfig) {
```

```
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::UpdateUserRequest request;
    request.SetUserName(currentUserName);
    request.SetNewUserName(newUserName);

    auto outcome = iam.UpdateUser(request);
    if (outcome.IsSuccess()) {
        std::cout << "IAM user " << currentUserName <<
                  " successfully updated with new user name " << newUserName <<
                  std::endl;
    }
    else {
        std::cerr << "Error updating user name for IAM user " << currentUserName
 <<
                  ":" << outcome.GetError().GetMessage() << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come modificare il nome di un utente IAM

Il comando `update-user` seguente modifica il nome di un utente IAM da `Bob` a `Robert`.

```
aws iam update-user \
    --user-name Bob \
    --new-user-name Robert
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Ridenominazione di un gruppo di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta UpdateUser AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.UpdateUserRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateUser {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <curName> <newName>\s

                Where:
                    curName - The current user name.\s
                    newName - An updated user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String curName = args[0];
```

```
        String newName = args[1];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        updateIAMUser(iam, curName, newName);
        System.out.println("Done");
        iam.close();
    }

    public static void updateIAMUser(IamClient iam, String curName, String
 newName) {
        try {
            UpdateUserRequest request = UpdateUserRequest.builder()
                    .userName(curName)
                    .newUserName(newName)
                    .build();

            iam.updateUser(request);
            System.out.printf("Successfully updated user to username %s",
 newName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Aggiorna l'utente.

```
import { UpdateUserCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} currentUserName
 * @param {string} newUserName
 */
export const updateUser = (currentUserName, newUserName) => {
  const command = new UpdateUserCommand({
    UserName: currentUserName,
    NewUserName: newUserName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });
```

```
var params = {
  UserName: process.argv[2],
  NewUserName: process.argv[3],
};

iam.updateUser(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
suspend fun updateIAMUser(
    curName: String?,
    newName: String?,
) {
    val request =
        UpdateUserRequest {
            userName = curName
            newUserName = newName
        }

    IamClient.fromEnvironment { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.updateUser(request)
        println("Successfully updated user to $newName")
```

```
    }
}
```

- Per i dettagli sull'API, UpdateUserconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per V4 PowerShell

Esempio 1: questo esempio rinomina l'utente IAM **Bob** in **Robert**.

```
Update-IAMUser -UserName Bob -NewUserName Robert
```

Esempio 2: questo esempio modifica il percorso dell'utente IAM **Bob** in **/Org1/Org2/**, il che modifica effettivamente l'ARN dell'utente in **arn:aws:iam::123456789012:user/Org1/Org2/bob**.

```
Update-IAMUser -UserName Bob -NewPath /Org1/Org2/
```

- Per i dettagli sull'API, vedere UpdateUserin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio rinomina l'utente IAM **Bob** in **Robert**.

```
Update-IAMUser -UserName Bob -NewUserName Robert
```

Esempio 2: questo esempio modifica il percorso dell'utente IAM **Bob** in **/Org1/Org2/**, il che modifica effettivamente l'ARN dell'utente in **arn:aws:iam::123456789012:user/Org1/Org2/bob**.

```
Update-IAMUser -UserName Bob -NewPath /Org1/Org2/
```

- Per i dettagli sull'API, vedere UpdateUserin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def update_user(user_name, new_user_name):
    """
    Updates a user's name.

    :param user_name: The current name of the user to update.
    :param new_user_name: The new name to assign to the user.
    :return: The updated user.
    """
    try:
        user = iam.User(user_name)
        user.update(NewUserName=new_user_name)
        logger.info("Renamed %s to %s.", user_name, new_user_name)
    except ClientError:
        logger.exception("Couldn't update name for user %s.", user_name)
        raise
    return user
```

- Per i dettagli sull'API, consulta UpdateUser AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
  # Updates an IAM user's name
  #
  # @param current_name [String] The current name of the user
  # @param new_name [String] The new name of the user
  def update_user_name(current_name, new_name)
    @iam_client.update_user(user_name: current_name, new_user_name: new_name)
    true
  rescue StandardError => e
    @logger.error("Error updating user name from '#{current_name}' to
'#{new_name}': #{e.message}")
    false
  end
```

- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK per Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        lo_iam->updateuser(
          iv_username = iv_user_name
          iv_newusername = iv_new_user_name ).
        MESSAGE 'User updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'User does not exist.' TYPE 'E'.
      CATCH /aws1/cx_iamentityalrdyexex.
        MESSAGE 'New user name already exists.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, UpdateUserconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`UploadServerCertificate`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`UploadServerCertificate`.

CLI

AWS CLI

Per caricare un certificato server sul tuo account AWS

Il seguente comando upload-server-certificate carica un certificato server sul tuo account. AWS In questo esempio, il certificato è nel file `public_key_cert_file.pem`, la chiave privata associata è nel file `my_private_key.pem` e la catena di certificati fornita dall'autorità di certificazione (CA) è nel file `my_certificate_chain_file.pem`. Al termine del caricamento, il file è disponibile con il nome my. ServerCertificate I parametri che iniziano con `file://` indicano al comando di leggere il contenuto del file e di utilizzarlo come valore del parametro in luogo del nome del file.

```
aws iam upload-server-certificate \
    --server-certificate-name myServerCertificate \
    --certificate-body file://public_key_cert_file.pem \
    --private-key file://my_private_key.pem \
    --certificate-chain file://my_certificate_chain_file.pem
```

Output:

```
{
    "ServerCertificateMetadata": {
        "Path": "/",
        "ServerCertificateName": "myServerCertificate",
        "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
        "Arn": "arn:aws:iam::1234567989012:server-certificate/
myServerCertificate",
        "UploadDate": "2019-04-22T21:13:44+00:00",
        "Expiration": "2019-10-15T22:23:16+00:00"
    }
}
```

Per ulteriori informazioni, consulta Creazione, caricamento ed eliminazione di certificati server nella guida Utilizzo di IAM.

- Per i dettagli sull'API, consulta UploadServerCertificate AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { UploadServerCertificateCommand, IAMClient } from "@aws-sdk/client-iam";
import { readFileSync } from "node:fs";
import { dirnameFromMetaUrl } from "@aws-doc-sdk-examples/lib/utils/util-fs.js";
import * as path from "node:path";

const client = new IAMClient({});

const certMessage = `Generate a certificate and key with the following command,
 or the equivalent for your system.

openssl req -x509 -newkey rsa:4096 -sha256 -days 3650 -nodes \
-keyout example.key -out example.crt -subj "/CN=example.com" \
-addext "subjectAltName=DNS:example.com,DNS:www.example.net,IP:10.0.0.1"
`;

const getCertAndKey = () => {
  try {
    const cert = readFileSync(
      path.join(dirnameFromMetaUrl(import.meta.url), "./example.crt"),
    );
    const key = readFileSync(
      path.join(dirnameFromMetaUrl(import.meta.url), "./example.key"),
    );
    return { cert, key };
  } catch (err) {
    if (err.code === "ENOENT") {
      throw new Error(
```

```
        `Certificate and/or private key not found. ${certMessage}`,
      );
    }

    throw err;
  }
};

/**
 *
 * @param {string} certificateName
 */
export const uploadServerCertificate = (certificateName) => {
  const { cert, key } = getCertAndKey();
  const command = new UploadServerCertificateCommand({
    ServerCertificateName: certificateName,
    CertificateBody: cert.toString(),
    PrivateKey: key.toString(),
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, UploadServerCertificateconsulta AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio carica un nuovo certificato server sull'account IAM. I file contenenti il corpo del certificato, la chiave privata e (facoltativamente) la catena di certificati devono tutti essere codificati con PEM. Tieni presente che i parametri richiedono il contenuto effettivo dei file e non i nomi dei file. Per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro di cambio **`-Raw`**.

```
Publish-IAMServerCertificate -ServerCertificateName MyTestCert -CertificateBody
 (Get-Content -Raw server.crt) -PrivateKey (Get-Content -Raw server.key)
```

Output:

```
Arn                  : arn:aws:iam::123456789012:server-certificate/MyTestCert
Expiration           : 1/14/2018 9:52:36 AM
Path                 : /
ServerCertificateId  : ASCAJIEXAMPLE7J7HQZYW
ServerCertificateName : MyTestCert
UploadDate           : 4/21/2015 11:14:16 AM
```

- Per i dettagli sull'API, vedere UploadServerCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio carica un nuovo certificato server sull'account IAM. I file contenenti il corpo del certificato, la chiave privata e (facoltativamente) la catena di certificati devono tutti essere codificati con PEM. Tieni presente che i parametri richiedono il contenuto effettivo dei file e non i nomi dei file. Per elaborare correttamente il contenuto del file JSON è necessario utilizzare il parametro di cambio **`-Raw`**.

```
Publish-IAMServerCertificate -ServerCertificateName MyTestCert -CertificateBody
 (Get-Content -Raw server.crt) -PrivateKey (Get-Content -Raw server.key)
```

Output:

```
Arn                  : arn:aws:iam::123456789012:server-certificate/MyTestCert
Expiration           : 1/14/2018 9:52:36 AM
Path                 : /
ServerCertificateId  : ASCAJIEXAMPLE7J7HQZYW
ServerCertificateName : MyTestCert
UploadDate           : 4/21/2015 11:14:16 AM
```

- Per i dettagli sull'API, vedere UploadServerCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`UploadSigningCertificate`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`UploadSigningCertificate`.

CLI

AWS CLI

Per caricare un certificato di firma per un utente IAM

Il comando `upload-signing-certificate` seguente carica il certificato di firma per l'utente IAM denominato Bob.

```
aws iam upload-signing-certificate \
    --user-name Bob \
    --certificate-body file://certificate.pem
```

Output:

```
{
    "Certificate": {
        "UserName": "Bob",
        "Status": "Active",
        "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----END
 CERTIFICATE-----",
        "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
        "UploadDate": "2013-06-06T21:40:08.121Z"
    }
}
```

Il certificato si trova in un file denominato certificate.pem in formato PEM.

Per ulteriori informazioni, consulta Creazione e caricamento di un certificato di firma dell'utente nella guida Utilizzo di IAM.

- Per i dettagli sull'API, consulta UploadSigningCertificate AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: questo esempio carica un nuovo certificato di X.509 firma e lo associa all'utente IAM denominato. **`Bob`** Il file contenente il corpo del certificato è codificato in PEM. Il parametro **`CertificateBody`** richiede il contenuto effettivo del file e non il nome del file. Per elaborare correttamente il file è necessario utilizzare il parametro di cambio **`-Raw`**.

```
Publish-IAMSigningCertificate -UserName Bob -CertificateBody (Get-Content -Raw
 SampleSigningCert.pem)
```

Output:

```
CertificateBody : -----BEGIN CERTIFICATE-----

 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC

 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6

 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd

 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN

 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD

 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z

 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
                  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn
+a4GmWIWJ
                  21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9T
                  rDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE

 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
                  nUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb

 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
                  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
                  -----END CERTIFICATE-----
CertificateId   : Y3EK7RMEXAMPLESV33FCEXAMPLEHMJLU
Status          : Active
UploadDate      : 4/20/2015 1:26:01 PM
UserName        : Bob
```

- Per i dettagli sull'API, vedere UploadSigningCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: questo esempio carica un nuovo certificato di X.509 firma e lo associa all'utente IAM denominato. **Bob** Il file contenente il corpo del certificato è codificato in PEM. Il parametro **CertificateBody** richiede il contenuto effettivo del file e non il nome del file. Per elaborare correttamente il file è necessario utilizzare il parametro di cambio **-Raw**.

```
Publish-IAMSigningCertificate -UserName Bob -CertificateBody (Get-Content -Raw
 SampleSigningCert.pem)
```

Output:

```
CertificateBody : -----BEGIN CERTIFICATE-----

 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC

 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6

 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd

 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN

 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD

 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z

 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
                  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn
+a4GmWIWJ
                  21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9T
                  rDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE

 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
                  nUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb

 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
                  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
                  -----END CERTIFICATE-----
CertificateId   : Y3EK7RMEXAMPLESV33FCEXAMPLEHMJLU
Status          : Active
```

```
UploadDate      : 4/20/2015 1:26:01 PM
UserName        : Bob
```

- Per i dettagli sull'API, vedere UploadSigningCertificatein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per l'utilizzo di IAM AWS SDK

I seguenti esempi di codice mostrano come implementare scenari comuni in IAM con AWS SDK. Questi scenari illustrano come eseguire attività specifiche richiamando più funzioni all'interno di IAM o combinate con altri Servizi AWS. Ogni scenario include un collegamento al codice sorgente completo, dove è possibile trovare le istruzioni su come configurare ed eseguire il codice.

Gli scenari sono relativi a un livello intermedio di esperienza per aiutarti a comprendere le azioni di servizio nel contesto.

Esempi

- Crea e gestisci un servizio resiliente utilizzando un AWS SDK
- Configurare la connettività dei servizi container
- Crea un'attività contenitore per il tipo di avvio serverless
- Crea un'API REST con integrazione proxy di funzioni
- Crea utenti IAM di sola lettura e lettura-scrittura utilizzando un AWS SDK
- Creazione di un servizio container per le istanze di macchine virtuali
- Creazione di un'area di lavoro di monitoraggio gestita
- Creazione di un dashboard di monitoraggio con il nome della funzione come variabile
- Creazione della tua prima funzione serverless
- Inizia a usare i data warehouse serverless utilizzando la CLI
- Guida introduttiva alla gestione della configurazione
- Guida introduttiva alla protezione dei dispositivi tramite Internet of Things
- Guida introduttiva ai feature store di machine learning

- Guida introduttiva ai cluster Kubernetes gestiti
- Guida introduttiva allo streaming gestito
- Guida introduttiva ai cluster di data warehouse con provisioning
- Guida introduttiva all'orchestrazione del flusso di lavoro
- Gestisci le chiavi di accesso IAM utilizzando un AWS SDK
- Gestisci le policy IAM utilizzando un AWS SDK
- Gestisci i ruoli IAM utilizzando un AWS SDK
- Gestisci il tuo account IAM utilizzando un AWS SDK
- Spostamento di segreti codificati per proteggere l'archiviazione segreta
- Permette AWS Compute Optimizer Funzionalità di automazione per applicare le azioni consigliate
- Politica per abilitare l'automazione in tutta l'organizzazione
- Politica per abilitare l'automazione per il tuo account
- Politica per concedere l'accesso completo a Compute Optimizer Automation per un account di gestione di un'organizzazione
- Politica per concedere l'accesso completo a Compute Optimizer Automation per la versione standalone AWS account
- Politica per concedere l'accesso in sola lettura a Compute Optimizer Automation per un account di gestione di un'organizzazione
- Politica per concedere l'accesso in sola lettura a Compute Optimizer Automation per uso autonomo AWS account
- Politica per la concessione di autorizzazioni di ruolo collegate ai servizi per Compute Optimization Automation
- Ripristina una versione della policy IAM utilizzando un AWS SDK
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori
- Configurare il controllo Attribute-Based degli accessi per DynamoDB utilizzando AWS Command Line Interface v2
- Configurazione della gestione dei sistemi
- Utilizzo delle variabili di proprietà nei dashboard di monitoraggio per monitorare più funzioni serverless
- Lavora con DynamoDB Streams e utilizza Time-to-Live AWS Command Line Interface v2
- Lavora con l'API IAM Policy Builder utilizzando un AWS SDK

Crea e gestisci un servizio resiliente utilizzando un AWS SDK

I seguenti esempi di codice mostrano come creare un servizio web con bilanciamento del carico che restituisca consigli su libri, film e canzoni. L'esempio mostra come il servizio risponde ai guasti e spiega come ristrutturarlo per una maggiore resilienza in caso di guasti.

- Utilizza un gruppo Amazon EC2 Auto Scaling per creare istanze Amazon Elastic Compute Cloud (Amazon EC2) basate su un modello di avvio e per mantenere il numero di istanze entro un intervallo specificato.
- Gestisci e distribuisci le richieste HTTP con Elastic Load Balancing.
- Monitora lo stato delle istanze in un gruppo Auto Scaling e inoltra le richieste soltanto alle istanze integre.
- Esegui un server Web Python su ogni istanza EC2 per gestire le richieste HTTP. Il server Web risponde con consigli e controlli dell'integrità.
- Simula un servizio di raccomandazione con una tabella Amazon DynamoDB.
- Controlla la risposta del server web alle richieste e ai controlli di integrità aggiornando AWS Systems Manager i parametri.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Esegui lo scenario interattivo al prompt dei comandi.

```
static async Task Main(string[] args)
{
    _configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally, load local settings.
        .Build();
```

```
        // Set up dependency injection for the AWS services.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonIdentityManagementService>()
                    .AddAWSService<IAmazonDynamoDB>()
                    .AddAWSService<IAmazonElasticLoadBalancingV2>()
                    .AddAWSService<IAmazonSimpleSystemsManagement>()
                    .AddAWSService<IAmazonAutoScaling>()
                    .AddAWSService<IAmazonEC2>()
                    .AddTransient<AutoScalerWrapper>()
                    .AddTransient<ElasticLoadBalancerWrapper>()
                    .AddTransient<SmParameterWrapper>()
                    .AddTransient<Recommendations>()
                    .AddSingleton<IConfiguration>(_configuration)
            )
            .Build();

        ServicesSetup(host);
        ResourcesSetup();

        try
        {
            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Welcome to the Resilient Architecture Example
Scenario.");
            Console.WriteLine(new string('-', 80));
            await Deploy(true);

            Console.WriteLine("Now let's begin the scenario.");
            Console.WriteLine(new string('-', 80));
            await Demo(true);

            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Finally, let's clean up our resources.");
            Console.WriteLine(new string('-', 80));
```

```
            await DestroyResources(true);

            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Resilient Architecture Example Scenario is
complete.");
            Console.WriteLine(new string('-', 80));
        }
        catch (Exception ex)
        {
            Console.WriteLine(new string('-', 80));
            Console.WriteLine($"There was a problem running the scenario:
{ex.Message}");
            await DestroyResources(true);
            Console.WriteLine(new string('-', 80));
        }
    }

    /// <summary>
    /// Setup any common resources, also used for integration testing.
    /// </summary>
    public static void ResourcesSetup()
    {
        _httpClient = new HttpClient();
    }

    /// <summary>
    /// Populate the services for use within the console application.
    /// </summary>
    /// <param name="host">The services host.</param>
    private static void ServicesSetup(IHost host)
    {
        _elasticLoadBalancerWrapper =
host.Services.GetRequiredService<ElasticLoadBalancerWrapper>();
        _iamClient =
host.Services.GetRequiredService<IAmazonIdentityManagementService>();
        _recommendations = host.Services.GetRequiredService<Recommendations>();
        _autoScalerWrapper =
host.Services.GetRequiredService<AutoScalerWrapper>();
        _smParameterWrapper =
host.Services.GetRequiredService<SmParameterWrapper>();
    }

    /// <summary>
    /// Deploy necessary resources for the scenario.
```

```
    /// </summary>
    /// <param name="interactive">True to run as interactive.</param>
    /// <returns>True if successful.</returns>
    public static async Task<bool> Deploy(bool interactive)
    {
        var protocol = "HTTP";
        var port = 80;
        var sshPort = 22;

        Console.WriteLine(
            "\nFor this demo, we'll use the AWS SDK for .NET to create several
AWS resources\n" +
            "to set up a load-balanced web service endpoint and explore some ways
to make it resilient\n" +
            "against various kinds of failures.\n\n" +
            "Some of the resources create by this demo are:\n");

        Console.WriteLine(
            "\t* A DynamoDB table that the web service depends on to provide
book, movie, and song recommendations.");
        Console.WriteLine(
            "\t* An EC2 launch template that defines EC2 instances that each
contain a Python web server.");
        Console.WriteLine(
            "\t* An EC2 Auto Scaling group that manages EC2 instances across
several Availability Zones.");
        Console.WriteLine(
            "\t* An Elastic Load Balancing (ELB) load balancer that targets the
Auto Scaling group to distribute requests.");
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you're ready to start deploying
resources.");
        if (interactive)
            Console.ReadLine();

        // Create and populate the DynamoDB table.
        var databaseTableName = _configuration["databaseName"];
        var recommendationsPath = Path.Join(_configuration["resourcePath"],
            "recommendations_objects.json");
        Console.WriteLine($"Creating and populating a DynamoDB table named
{databaseTableName}.");
        await _recommendations.CreateDatabaseWithName(databaseTableName);
        await _recommendations.PopulateDatabase(databaseTableName,
recommendationsPath);
```

```
        Console.WriteLine(new string('-', 80));

        // Create the EC2 Launch Template.

        Console.WriteLine(
            $"Creating an EC2 launch template that runs
'server_startup_script.sh' when an instance starts.\n"
            + "\nThis script starts a Python web server defined in the
`server.py` script. The web server\n"
            + "listens to HTTP requests on port 80 and responds to requests to
'/' and to '/healthcheck'.\n"
            + "For demo purposes, this server is run as the root user. In
production, the best practice is to\n"
            + "run a web server, such as Apache, with least-privileged
credentials.");
        Console.WriteLine(
            "\nThe template also defines an IAM policy that each instance uses to
assume a role that grants\n"
            + "permissions to access the DynamoDB recommendation table and
Systems Manager parameters\n"
            + "that control the flow of the demo.");

        var startupScriptPath = Path.Join(_configuration["resourcePath"],
            "server_startup_script.sh");
        var instancePolicyPath = Path.Join(_configuration["resourcePath"],
            "instance_policy.json");
        await _autoScalerWrapper.CreateTemplate(startupScriptPath,
instancePolicyPath);
        Console.WriteLine(new string('-', 80));

        Console.WriteLine(
            "Creating an EC2 Auto Scaling group that maintains three EC2
instances, each in a different\n"
            + "Availability Zone.\n");
        var zones = await _autoScalerWrapper.DescribeAvailabilityZones();
        await _autoScalerWrapper.CreateGroupOfSize(3,
_autoScalerWrapper.GroupName, zones);
        Console.WriteLine(new string('-', 80));

        Console.WriteLine(
            "At this point, you have EC2 instances created. Once each instance
starts, it listens for\n"
            + "HTTP requests. You can see these instances in the console or
continue with the demo.\n");
```

```
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you're ready to continue.");
        if (interactive)
            Console.ReadLine();

        Console.WriteLine("Creating variables that control the flow of the
demo.");
        await _smParameterWrapper.Reset();

        Console.WriteLine(
            "\nCreating an Elastic Load Balancing target group and load balancer.
The target group\n"
            + "defines how the load balancer connects to instances. The load
balancer provides a\n"
            + "single endpoint where clients connect and dispatches requests to
instances in the group.");

        var defaultVpc = await _autoScalerWrapper.GetDefaultVpc();
        var subnets = await
_autoScalerWrapper.GetAllVpcSubnetsForZones(defaultVpc.VpcId, zones);
        var subnetIds = subnets.Select(s => s.SubnetId).ToList();
        var targetGroup = await
_elasticLoadBalancerWrapper.CreateTargetGroupOnVpc(_elasticLoadBalancerWrapper.TargetGro
protocol, port, defaultVpc.VpcId);

        await
_elasticLoadBalancerWrapper.CreateLoadBalancerAndListener(_elasticLoadBalancerWrapper.Lo
subnetIds, targetGroup);
        await
_autoScalerWrapper.AttachLoadBalancerToGroup(_autoScalerWrapper.GroupName,
targetGroup.TargetGroupArn);
        Console.WriteLine("\nVerifying access to the load balancer endpoint...");
        var endPoint = await
_elasticLoadBalancerWrapper.GetEndpointForLoadBalancerByName(_elasticLoadBalancerWrapper
        var loadBalancerAccess = await
_elasticLoadBalancerWrapper.VerifyLoadBalancerEndpoint(endPoint);

        if (!loadBalancerAccess)
        {
            Console.WriteLine("\nCouldn't connect to the load balancer, verifying
that the port is open...");
```

```
            var ipString = await _httpClient.GetStringAsync("https://
checkip.amazonaws.com");
            ipString = ipString.Trim();

            var defaultSecurityGroup = await
 _autoScalerWrapper.GetDefaultSecurityGroupForVpc(defaultVpc);
            var portIsOpen =
 _autoScalerWrapper.VerifyInboundPortForGroup(defaultSecurityGroup, port,
 ipString);
            var sshPortIsOpen =
 _autoScalerWrapper.VerifyInboundPortForGroup(defaultSecurityGroup, sshPort,
 ipString);

            if (!portIsOpen)
            {
                Console.WriteLine(
                    "\nFor this example to work, the default security group for
 your default VPC must\n"
                    + "allows access from this computer. You can either add it
 automatically from this\n"
                    + "example or add it yourself using the AWS Management
 Console.\n");

                if (!interactive || GetYesNoResponse(
                        "Do you want to add a rule to the security group to allow
 inbound traffic from your computer's IP address?"))
                {
                    await
 _autoScalerWrapper.OpenInboundPort(defaultSecurityGroup.GroupId, port,
 ipString);
                }
            }

            if (!sshPortIsOpen)
            {
                if (!interactive || GetYesNoResponse(
                        "Do you want to add a rule to the security group to allow
 inbound SSH traffic for debugging from your computer's IP address?"))
                {
                    await
 _autoScalerWrapper.OpenInboundPort(defaultSecurityGroup.GroupId, sshPort,
 ipString);
                }
            }
```

```
            loadBalancerAccess = await
_elasticLoadBalancerWrapper.VerifyLoadBalancerEndpoint(endPoint);
        }

        if (loadBalancerAccess)
        {
            Console.WriteLine("Your load balancer is ready. You can access it by
browsing to:");
            Console.WriteLine($"\thttp://{endPoint}\n");
        }
        else
        {
            Console.WriteLine(
                "\nCouldn't get a successful response from the load balancer
endpoint. Troubleshoot by\n"
                + "manually verifying that your VPC and security group are
configured correctly and that\n"
                + "you can successfully make a GET request to the load balancer
endpoint:\n");
            Console.WriteLine($"\thttp://{endPoint}\n");
        }
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you're ready to continue with the
demo.");
        if (interactive)
            Console.ReadLine();
        return true;
    }

    /// <summary>
    /// Demonstrate the steps of the scenario.
    /// </summary>
    /// <param name="interactive">True to run as an interactive scenario.</param>
    /// <returns>Async task.</returns>
    public static async Task<bool> Demo(bool interactive)
    {
        var ssmOnlyPolicy = Path.Join(_configuration["resourcePath"],
            "ssm_only_policy.json");

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Resetting parameters to starting values for demo.");
        await _smParameterWrapper.Reset();
```

```
        Console.WriteLine("\nThis part of the demonstration shows how to toggle
different parts of the system\n" +
                          "to create situations where the web service fails, and
shows how using a resilient\n" +
                          "architecture can keep the web service running in spite
of these failures.");
        Console.WriteLine(new string('-', 88));
        Console.WriteLine("At the start, the load balancer endpoint returns
recommendations and reports that all targets are healthy.");
        if (interactive)
            await DemoActionChoices();

        Console.WriteLine($"The web service running on the EC2 instances gets
recommendations by querying a DynamoDB table.\n" +
                          $"The table name is contained in a Systems Manager
parameter named '{_smParameterWrapper.TableParameter}'.\n" +
                          $"To simulate a failure of the recommendation service,
let's set this parameter to name a non-existent table.\n");
        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.TableParameter,
"this-is-not-a-table");
        Console.WriteLine("\nNow, sending a GET request to the load balancer
endpoint returns a failure code. But, the service reports as\n" +
                          "healthy to the load balancer because shallow health
checks don't check for failure of the recommendation service.");
        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("Instead of failing when the recommendation service
fails, the web service can return a static response.");
        Console.WriteLine("While this is not a perfect solution, it presents the
customer with a somewhat better experience than failure.");

        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.FailureResponseParameter,
"static");

        Console.WriteLine("\nNow, sending a GET request to the load balancer
endpoint returns a static response.");
        Console.WriteLine("The service still reports as healthy because health
checks are still shallow.");
        if (interactive)
            await DemoActionChoices();
```

```
        Console.WriteLine("Let's reinstate the recommendation service.\n");
        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.TableParameter,
_smParameterWrapper.TableName);
        Console.WriteLine(
            "\nLet's also substitute bad credentials for one of the instances in
the target group so that it can't\n" +
            "access the DynamoDB recommendation table.\n"
        );
        await _autoScalerWrapper.CreateInstanceProfileWithName(
            _autoScalerWrapper.BadCredsPolicyName,
            _autoScalerWrapper.BadCredsRoleName,
            _autoScalerWrapper.BadCredsProfileName,
            ssmOnlyPolicy,
            new List<string> { "AmazonSSMManagedInstanceCore" }
        );
        var instances = await
_autoScalerWrapper.GetInstancesByGroupName(_autoScalerWrapper.GroupName);
        var badInstanceId = instances.First();
        var instanceProfile = await
_autoScalerWrapper.GetInstanceProfile(badInstanceId);
        Console.WriteLine(
            $"Replacing the profile for instance {badInstanceId} with a profile
that contains\n" +
            "bad credentials...\n"
        );
        await _autoScalerWrapper.ReplaceInstanceProfile(
            badInstanceId,
            _autoScalerWrapper.BadCredsProfileName,
            instanceProfile.AssociationId
        );
        Console.WriteLine(
            "Now, sending a GET request to the load balancer endpoint returns
either a recommendation or a static response,\n" +
            "depending on which instance is selected by the load balancer.\n"
        );
        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("\nLet's implement a deep health check. For this demo,
a deep health check tests whether");
        Console.WriteLine("the web service can access the DynamoDB table that it
depends on for recommendations. Note that");
```

```
        Console.WriteLine("the deep health check is only for ELB routing and not
for Auto Scaling instance health.");
        Console.WriteLine("This kind of deep health check is not recommended for
Auto Scaling instance health, because it");
        Console.WriteLine("risks accidental termination of all instances in the
Auto Scaling group when a dependent service fails.");

        Console.WriteLine("\nBy implementing deep health checks, the load
balancer can detect when one of the instances is failing");
        Console.WriteLine("and take that instance out of rotation.");

        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.HealthCheckParameter,
"deep");

        Console.WriteLine($"\nNow, checking target health indicates that the
instance with bad credentials ({badInstanceId})");
        Console.WriteLine("is unhealthy. Note that it might take a minute or two
for the load balancer to detect the unhealthy");
        Console.WriteLine("instance. Sending a GET request to the load balancer
endpoint always returns a recommendation, because");
        Console.WriteLine("the load balancer takes unhealthy instances out of its
rotation.");

        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("\nBecause the instances in this demo are controlled by
an auto scaler, the simplest way to fix an unhealthy");
        Console.WriteLine("instance is to terminate it and let the auto scaler
start a new instance to replace it.");

        await _autoScalerWrapper.TryTerminateInstanceById(badInstanceId);

        Console.WriteLine($"\nEven while the instance is terminating and the new
instance is starting, sending a GET");
        Console.WriteLine("request to the web service continues to get a
successful recommendation response because");
        Console.WriteLine("starts and reports as healthy, it is included in the
load balancing rotation.");
        Console.WriteLine("Note that terminating and replacing an instance
typically takes several minutes, during which time you");
        Console.WriteLine("can see the changing health check status until the new
instance is running and healthy.");
```

```
        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("\nIf the recommendation service fails now, deep health
checks mean all instances report as unhealthy.");

        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.TableParameter,
"this-is-not-a-table");

        Console.WriteLine($"\nWhen all instances are unhealthy, the load balancer
continues to route requests even to");
        Console.WriteLine("unhealthy instances, allowing them to fail open and
return a static response rather than fail");
        Console.WriteLine("closed and report failure to the customer.");

        if (interactive)
            await DemoActionChoices();
        await _smParameterWrapper.Reset();

        Console.WriteLine(new string('-', 80));
        return true;
    }

    /// <summary>
    /// Clean up the resources from the scenario.
    /// </summary>
    /// <param name="interactive">True to ask the user for cleanup.</param>
    /// <returns>Async task.</returns>
    public static async Task<bool> DestroyResources(bool interactive)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine(
            "To keep things tidy and to avoid unwanted charges on your account,
we can clean up all AWS resources\n" +
            "that were created for this demo."
        );

        if (!interactive || GetYesNoResponse("Do you want to clean up all demo
resources? (y/n) "))
        {
            await
_elasticLoadBalancerWrapper.DeleteLoadBalancerByName(_elasticLoadBalancerWrapper.LoadBal
```

```
            await
_elasticLoadBalancerWrapper.DeleteTargetGroupByName(_elasticLoadBalancerWrapper.TargetGr
            await
_autoScalerWrapper.TerminateAndDeleteAutoScalingGroupWithName(_autoScalerWrapper.GroupNa
            await
_autoScalerWrapper.DeleteKeyPairByName(_autoScalerWrapper.KeyPairName);
            await
_autoScalerWrapper.DeleteTemplateByName(_autoScalerWrapper.LaunchTemplateName);
            await _autoScalerWrapper.DeleteInstanceProfile(
                _autoScalerWrapper.BadCredsProfileName,
                _autoScalerWrapper.BadCredsRoleName
            );
            await
_recommendations.DestroyDatabaseByName(_recommendations.TableName);
        }
        else
        {
            Console.WriteLine(
                "Ok, we'll leave the resources intact.\n" +
                "Don't forget to delete them when you're done with them or you
might incur unexpected charges."
            );
        }

        Console.WriteLine(new string('-', 80));
        return true;
    }
```

Crea una classe che racchiuda le operazioni di dimensionamento automatico e Amazon EC2.

```
/// <summary>
/// Encapsulates Amazon EC2 Auto Scaling and EC2 management methods.
/// </summary>
public class AutoScalerWrapper
{
    private readonly IAmazonAutoScaling _amazonAutoScaling;
    private readonly IAmazonEC2 _amazonEc2;
    private readonly IAmazonSimpleSystemsManagement _amazonSsm;
    private readonly IAmazonIdentityManagementService _amazonIam;
    private readonly ILogger<AutoScalerWrapper> _logger;

    private readonly string _instanceType = "";
```

```
    private readonly string _amiParam = "";
    private readonly string _launchTemplateName = "";
    private readonly string _groupName = "";
    private readonly string _instancePolicyName = "";
    private readonly string _instanceRoleName = "";
    private readonly string _instanceProfileName = "";
    private readonly string _badCredsProfileName = "";
    private readonly string _badCredsRoleName = "";
    private readonly string _badCredsPolicyName = "";
    private readonly string _keyPairName = "";

    public string GroupName => _groupName;
    public string KeyPairName => _keyPairName;
    public string LaunchTemplateName => _launchTemplateName;
    public string InstancePolicyName => _instancePolicyName;
    public string BadCredsProfileName => _badCredsProfileName;
    public string BadCredsRoleName => _badCredsRoleName;
    public string BadCredsPolicyName => _badCredsPolicyName;

    /// <summary>
    /// Constructor for the AutoScalerWrapper.
    /// </summary>
    /// <param name="amazonAutoScaling">The injected AutoScaling client.</param>
    /// <param name="amazonEc2">The injected EC2 client.</param>
    /// <param name="amazonIam">The injected IAM client.</param>
    /// <param name="amazonSsm">The injected SSM client.</param>
    public AutoScalerWrapper(
        IAmazonAutoScaling amazonAutoScaling,
        IAmazonEC2 amazonEc2,
        IAmazonSimpleSystemsManagement amazonSsm,
        IAmazonIdentityManagementService amazonIam,
        IConfiguration configuration,
        ILogger<AutoScalerWrapper> logger)
    {
        _amazonAutoScaling = amazonAutoScaling;
        _amazonEc2 = amazonEc2;
        _amazonSsm = amazonSsm;
        _amazonIam = amazonIam;
        _logger = logger;

        var prefix = configuration["resourcePrefix"];
        _instanceType = configuration["instanceType"];
        _amiParam = configuration["amiParam"];
```

```
        _launchTemplateName = prefix + "-template";
        _groupName = prefix + "-group";
        _instancePolicyName = prefix + "-pol";
        _instanceRoleName = prefix + "-role";
        _instanceProfileName = prefix + "-prof";
        _badCredsPolicyName = prefix + "-bc-pol";
        _badCredsRoleName = prefix + "-bc-role";
        _badCredsProfileName = prefix + "-bc-prof";
        _keyPairName = prefix + "-key-pair";
    }

    /// <summary>
    /// Create a policy, role, and profile that is associated with instances with
a specified name.
    /// An instance's associated profile defines a role that is assumed by the
    /// instance.The role has attached policies that specify the AWS permissions
granted to
    /// clients that run on the instance.
    /// </summary>
    /// <param name="policyName">Name to use for the policy.</param>
    /// <param name="roleName">Name to use for the role.</param>
    /// <param name="profileName">Name to use for the profile.</param>
    /// <param name="ssmOnlyPolicyFile">Path to a policy file for SSM.</param>
    /// <param name="awsManagedPolicies">AWS Managed policies to be attached to
the role.</param>
    /// <returns>The Arn of the profile.</returns>
    public async Task<string> CreateInstanceProfileWithName(
        string policyName,
        string roleName,
        string profileName,
        string ssmOnlyPolicyFile,
        List<string>? awsManagedPolicies = null)
    {

        var assumeRoleDoc = "{" +
                                    "\"Version\": \"2012-10-17\"," +
                                    "\"Statement\": [{" +
                                         "\"Effect\": \"Allow\"," +
                                         "\"Principal\": {" +
                                         "\"Service\": [" +
                                             "\"ec2.amazonaws.com\"" +
                                         "]" +
                                         "}," +
                                    "\"Action\": \"sts:AssumeRole\"" +
```

```
                                "}]" +
                            "}";

        var policyDocument = await File.ReadAllTextAsync(ssmOnlyPolicyFile);

        var policyArn = "";

        try
        {
            var createPolicyResult = await _amazonIam.CreatePolicyAsync(
                new CreatePolicyRequest
                {
                    PolicyName = policyName,
                    PolicyDocument = policyDocument
                });
            policyArn = createPolicyResult.Policy.Arn;
        }
        catch (EntityAlreadyExistsException)
        {
            // The policy already exists, so we look it up to get the Arn.
            var policiesPaginator = _amazonIam.Paginators.ListPolicies(
                new ListPoliciesRequest()
                {
                    Scope = PolicyScopeType.Local
                });
            // Get the entire list using the paginator.
            await foreach (var policy in policiesPaginator.Policies)
            {
                if (policy.PolicyName.Equals(policyName))
                {
                    policyArn = policy.Arn;
                }
            }

            if (policyArn == null)
            {
                throw new InvalidOperationException("Policy not found");
            }
        }

        try
        {
            await _amazonIam.CreateRoleAsync(new CreateRoleRequest()
            {
```

```
                RoleName = roleName,
                AssumeRolePolicyDocument = assumeRoleDoc,
            });
            await _amazonIam.AttachRolePolicyAsync(new AttachRolePolicyRequest()
            {
                RoleName = roleName,
                PolicyArn = policyArn
            });
            if (awsManagedPolicies != null)
            {
                foreach (var awsPolicy in awsManagedPolicies)
                {
                    await _amazonIam.AttachRolePolicyAsync(new
AttachRolePolicyRequest()
                    {
                        PolicyArn = $"arn:aws:iam::aws:policy/{awsPolicy}",
                        RoleName = roleName
                    });
                }
            }
        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine("Role already exists.");
        }

        string profileArn = "";
        try
        {
            var profileCreateResponse = await
_amazonIam.CreateInstanceProfileAsync(
                new CreateInstanceProfileRequest()
                {
                    InstanceProfileName = profileName
                });
            // Allow time for the profile to be ready.
            profileArn = profileCreateResponse.InstanceProfile.Arn;
            Thread.Sleep(10000);
            await _amazonIam.AddRoleToInstanceProfileAsync(
                new AddRoleToInstanceProfileRequest()
                {
                    InstanceProfileName = profileName,
                    RoleName = roleName
                });
```

```
        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine("Policy already exists.");
            var profileGetResponse = await _amazonIam.GetInstanceProfileAsync(
                new GetInstanceProfileRequest()
                {
                    InstanceProfileName = profileName
                });
            profileArn = profileGetResponse.InstanceProfile.Arn;
        }
        return profileArn;
    }

    /// <summary>
    /// Create a new key pair and save the file.
    /// </summary>
    /// <param name="newKeyPairName">The name of the new key pair.</param>
    /// <returns>Async task.</returns>
    public async Task CreateKeyPair(string newKeyPairName)
    {
        try
        {
            var keyResponse = await _amazonEc2.CreateKeyPairAsync(
                new CreateKeyPairRequest() { KeyName = newKeyPairName });
            await File.WriteAllTextAsync($"{newKeyPairName}.pem",
                keyResponse.KeyPair.KeyMaterial);
            Console.WriteLine($"Created key pair {newKeyPairName}.");
        }
        catch (AlreadyExistsException)
        {
            Console.WriteLine("Key pair already exists.");
        }
    }

    /// <summary>
    /// Delete the key pair and file by name.
    /// </summary>
    /// <param name="deleteKeyPairName">The key pair to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteKeyPairByName(string deleteKeyPairName)
    {
        try
```

```
        {
            await _amazonEc2.DeleteKeyPairAsync(
                new DeleteKeyPairRequest() { KeyName = deleteKeyPairName });
            File.Delete($"{deleteKeyPairName}.pem");
        }
        catch (FileNotFoundException)
        {
            Console.WriteLine($"Key pair {deleteKeyPairName} not found.");
        }
    }

    /// <summary>
    /// Creates an Amazon EC2 launch template to use with Amazon EC2 Auto
Scaling.
    /// The launch template specifies a Bash script in its user data field that
runs after
    /// the instance is started. This script installs the Python packages and
starts a Python
    /// web server on the instance.
    /// </summary>
    /// <param name="startupScriptPath">The path to a Bash script file that is
run.</param>
    /// <param name="instancePolicyPath">The path to a permissions policy to
create and attach to the profile.</param>
    /// <returns>The template object.</returns>
    public async Task<Amazon.EC2.Model.LaunchTemplate> CreateTemplate(string
startupScriptPath, string instancePolicyPath)
    {
        try
        {
            await CreateKeyPair(_keyPairName);
            await CreateInstanceProfileWithName(_instancePolicyName,
_instanceRoleName,
                _instanceProfileName, instancePolicyPath);

            var startServerText = await File.ReadAllTextAsync(startupScriptPath);
            var plainTextBytes =
System.Text.Encoding.UTF8.GetBytes(startServerText);

            var amiLatest = await _amazonSsm.GetParameterAsync(
                new GetParameterRequest() { Name = _amiParam });
            var amiId = amiLatest.Parameter.Value;
            var launchTemplateResponse = await
_amazonEc2.CreateLaunchTemplateAsync(
```

```
                new CreateLaunchTemplateRequest()
                {
                    LaunchTemplateName = _launchTemplateName,
                    LaunchTemplateData = new RequestLaunchTemplateData()
                    {
                        InstanceType = _instanceType,
                        ImageId = amiId,
                        IamInstanceProfile =
                            new

LaunchTemplateIamInstanceProfileSpecificationRequest()
                            {
                                Name = _instanceProfileName
                            },
                        KeyName = _keyPairName,
                        UserData = System.Convert.ToBase64String(plainTextBytes)
                    }
                });
            return launchTemplateResponse.LaunchTemplate;
        }
        catch (AmazonEC2Exception ec2Exception)
        {
            if (ec2Exception.ErrorCode ==
"InvalidLaunchTemplateName.AlreadyExistsException")
            {
                _logger.LogError($"Could not create the template, the name
{_launchTemplateName} already exists. " +
                                 $"Please try again with a unique name.");
            }

            throw;
        }
        catch (Exception ex)
        {
            _logger.LogError($"An error occurred while creating the template.:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Get a list of Availability Zones in the AWS Region of the Amazon EC2
Client.
    /// </summary>
```

```
    /// <returns>A list of availability zones.</returns>
    public async Task<List<string>> DescribeAvailabilityZones()
    {
        try
        {
            var zoneResponse = await _amazonEc2.DescribeAvailabilityZonesAsync(
                new DescribeAvailabilityZonesRequest());
            return zoneResponse.AvailabilityZones.Select(z =>
 z.ZoneName).ToList();
        }
        catch (AmazonEC2Exception ec2Exception)
        {
            _logger.LogError($"An Amazon EC2 error occurred while listing
 availability zones.: {ec2Exception.Message}");
            throw;
        }
        catch (Exception ex)
        {
            _logger.LogError($"An error occurred while listing availability
 zones.: {ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Create an EC2 Auto Scaling group of a specified size and name.
    /// </summary>
    /// <param name="groupSize">The size for the group.</param>
    /// <param name="groupName">The name for the group.</param>
    /// <param name="availabilityZones">The availability zones for the group.</
param>
    /// <returns>Async task.</returns>
    public async Task CreateGroupOfSize(int groupSize, string groupName,
 List<string> availabilityZones)
    {
        try
        {
            await _amazonAutoScaling.CreateAutoScalingGroupAsync(
                new CreateAutoScalingGroupRequest()
                {
                    AutoScalingGroupName = groupName,
                    AvailabilityZones = availabilityZones,
                    LaunchTemplate =
```

```
                        new
Amazon.AutoScaling.Model.LaunchTemplateSpecification()
                        {
                            LaunchTemplateName = _launchTemplateName,
                            Version = "$Default"
                        },
                    MaxSize = groupSize,
                    MinSize = groupSize
                });
            Console.WriteLine($"Created EC2 Auto Scaling group {groupName} with
size {groupSize}.");
        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine($"EC2 Auto Scaling group {groupName} already
exists.");
        }
    }

    /// <summary>
    /// Get the default VPC for the account.
    /// </summary>
    /// <returns>The default VPC object.</returns>
    public async Task<Vpc> GetDefaultVpc()
    {
        try
        {
            var vpcResponse = await _amazonEc2.DescribeVpcsAsync(
                new DescribeVpcsRequest()
                {
                    Filters = new List<Amazon.EC2.Model.Filter>()
                    {
                        new("is-default", new List<string>() { "true" })
                    }
                });
            return vpcResponse.Vpcs[0];
        }
        catch (AmazonEC2Exception ec2Exception)
        {
            if (ec2Exception.ErrorCode == "UnauthorizedOperation")
            {
                _logger.LogError(ec2Exception, $"You do not have the necessary
permissions to describe VPCs.");
            }
```

```
            throw;
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"An error occurred while describing the vpcs.:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Get all the subnets for a Vpc in a set of availability zones.
    /// </summary>
    /// <param name="vpcId">The Id of the Vpc.</param>
    /// <param name="availabilityZones">The list of availability zones.</param>
    /// <returns>The collection of subnet objects.</returns>
    public async Task<List<Subnet>> GetAllVpcSubnetsForZones(string vpcId,
List<string> availabilityZones)
    {
        try
        {
            var subnets = new List<Subnet>();
            var subnetPaginator = _amazonEc2.Paginators.DescribeSubnets(
                new DescribeSubnetsRequest()
                {
                    Filters = new List<Amazon.EC2.Model.Filter>()
                    {
                        new("vpc-id", new List<string>() { vpcId }),
                        new("availability-zone", availabilityZones),
                        new("default-for-az", new List<string>() { "true" })
                    }
                });

            // Get the entire list using the paginator.
            await foreach (var subnet in subnetPaginator.Subnets)
            {
                subnets.Add(subnet);
            }

            return subnets;
        }
        catch (AmazonEC2Exception ec2Exception)
        {
```

```
            if (ec2Exception.ErrorCode == "InvalidVpcID.NotFound")
            {
                _logger.LogError(ec2Exception, $"The specified VPC ID {vpcId}
does not exist.");
            }

            throw;
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"An error occurred while describing the
subnets.: {ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Delete a launch template by name.
    /// </summary>
    /// <param name="templateName">The name of the template to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteTemplateByName(string templateName)
    {
        try
        {
            await _amazonEc2.DeleteLaunchTemplateAsync(
                new DeleteLaunchTemplateRequest()
                {
                    LaunchTemplateName = templateName
                });
        }
        catch (AmazonEC2Exception ec2Exception)
        {
            if (ec2Exception.ErrorCode ==
"InvalidLaunchTemplateName.NotFoundException")
            {
                _logger.LogError(
                    $"Could not delete the template, the name
{_launchTemplateName} was not found.");
            }

            throw;
        }
        catch (Exception ex)
```

```
        {
            _logger.LogError($"An error occurred while deleting the template.:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Detaches a role from an instance profile, detaches policies from the
role,
    /// and deletes all the resources.
    /// </summary>
    /// <param name="profileName">The name of the profile to delete.</param>
    /// <param name="roleName">The name of the role to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteInstanceProfile(string profileName, string roleName)
    {
        try
        {
            await _amazonIam.RemoveRoleFromInstanceProfileAsync(
                new RemoveRoleFromInstanceProfileRequest()
                {
                    InstanceProfileName = profileName,
                    RoleName = roleName
                });
            await _amazonIam.DeleteInstanceProfileAsync(
                new DeleteInstanceProfileRequest() { InstanceProfileName =
profileName });
            var attachedPolicies = await
_amazonIam.ListAttachedRolePoliciesAsync(
                new ListAttachedRolePoliciesRequest() { RoleName = roleName });
            foreach (var policy in attachedPolicies.AttachedPolicies)
            {
                await _amazonIam.DetachRolePolicyAsync(
                    new DetachRolePolicyRequest()
                    {
                        RoleName = roleName,
                        PolicyArn = policy.PolicyArn
                    });
                // Delete the custom policies only.
                if (!policy.PolicyArn.StartsWith("arn:aws:iam::aws"))
                {
                    await _amazonIam.DeletePolicyAsync(
                        new Amazon.IdentityManagement.Model.DeletePolicyRequest()
```

```
                        {
                            PolicyArn = policy.PolicyArn
                        });
                }
            }

            await _amazonIam.DeleteRoleAsync(
                new DeleteRoleRequest() { RoleName = roleName });
        }
        catch (NoSuchEntityException)
        {
            Console.WriteLine($"Instance profile {profileName} does not exist.");
        }
    }

    /// <summary>
    /// Gets data about the instances in an EC2 Auto Scaling group by its group
name.
    /// </summary>
    /// <param name="group">The name of the auto scaling group.</param>
    /// <returns>A collection of instance Ids.</returns>
    public async Task<IEnumerable<string>> GetInstancesByGroupName(string group)
    {
        var instanceResponse = await
_amazonAutoScaling.DescribeAutoScalingGroupsAsync(
            new DescribeAutoScalingGroupsRequest()
            {
                AutoScalingGroupNames = new List<string>() { group }
            });
        var instanceIds = instanceResponse.AutoScalingGroups.SelectMany(
            g => g.Instances.Select(i => i.InstanceId));
        return instanceIds;
    }

    /// <summary>
    /// Get the instance profile association data for an instance.
    /// </summary>
    /// <param name="instanceId">The Id of the instance.</param>
    /// <returns>Instance profile associations data.</returns>
    public async Task<IamInstanceProfileAssociation> GetInstanceProfile(string
instanceId)
    {
        try
        {
```

```
            var response = await
_amazonEc2.DescribeIamInstanceProfileAssociationsAsync(
                new DescribeIamInstanceProfileAssociationsRequest()
                {
                    Filters = new List<Amazon.EC2.Model.Filter>()
                    {
                        new("instance-id", new List<string>() { instanceId })
                    },
                });
            return response.IamInstanceProfileAssociations[0];
        }
        catch (AmazonEC2Exception ec2Exception)
        {
            if (ec2Exception.ErrorCode == "InvalidInstanceID.NotFound")
            {
                _logger.LogError(ec2Exception, $"Instance {instanceId} not
found");
            }

            throw;
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"An error occurred while creating the
template.: {ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Replace the profile associated with a running instance. After the profile
is replaced, the instance
    /// is rebooted to ensure that it uses the new profile. When the instance is
ready, Systems Manager is
    /// used to restart the Python web server.
    /// </summary>
    /// <param name="instanceId">The Id of the instance to update.</param>
    /// <param name="credsProfileName">The name of the new profile to associate
with the specified instance.</param>
    /// <param name="associationId">The Id of the existing profile association
for the instance.</param>
    /// <returns>Async task.</returns>
    public async Task ReplaceInstanceProfile(string instanceId, string
credsProfileName, string associationId)
```

```
    {
        try
        {
            await _amazonEc2.ReplaceIamInstanceProfileAssociationAsync(
                new ReplaceIamInstanceProfileAssociationRequest()
                {
                    AssociationId = associationId,
                    IamInstanceProfile = new IamInstanceProfileSpecification()
                    {
                        Name = credsProfileName
                    }
                });
            // Allow time before resetting.
            Thread.Sleep(25000);

            await _amazonEc2.RebootInstancesAsync(
                new RebootInstancesRequest(new List<string>() { instanceId }));
            Thread.Sleep(25000);
            var instanceReady = false;
            var retries = 5;
            while (retries-- > 0 && !instanceReady)
            {
                var instancesPaginator =
                    _amazonSsm.Paginators.DescribeInstanceInformation(
                        new DescribeInstanceInformationRequest());
                // Get the entire list using the paginator.
                await foreach (var instance in
instancesPaginator.InstanceInformationList)
                {
                    instanceReady = instance.InstanceId == instanceId;
                    if (instanceReady)
                    {
                        break;
                    }
                }
            }
            Console.WriteLine("Waiting for instance to be running.");
            await WaitForInstanceState(instanceId, InstanceStateName.Running);
            Console.WriteLine("Instance ready.");
            Console.WriteLine($"Sending restart command to instance
{instanceId}");
            await _amazonSsm.SendCommandAsync(
                new SendCommandRequest()
                {
```

```
                    InstanceIds = new List<string>() { instanceId },
                    DocumentName = "AWS-RunShellScript",
                    Parameters = new Dictionary<string, List<string>>()
                    {
                        {
                            "commands",
                            new List<string>() { "cd / && sudo python3 server.py
80" }
                        }
                    }
                });
            Console.WriteLine($"Restarted the web server on instance
{instanceId}");
        }
        catch (AmazonEC2Exception ec2Exception)
        {
            if (ec2Exception.ErrorCode == "InvalidInstanceID.NotFound")
            {
                _logger.LogError(ec2Exception, $"Instance {instanceId} not
found");
            }

            throw;
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"An error occurred while replacing the
template.: {ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Try to terminate an instance by its Id.
    /// </summary>
    /// <param name="instanceId">The Id of the instance to terminate.</param>
    /// <returns>Async task.</returns>
    public async Task TryTerminateInstanceById(string instanceId)
    {
        var stopping = false;
        Console.WriteLine($"Stopping {instanceId}...");
        while (!stopping)
        {
            try
```

```
            {
                await
_amazonAutoScaling.TerminateInstanceInAutoScalingGroupAsync(
                    new TerminateInstanceInAutoScalingGroupRequest()
                    {
                        InstanceId = instanceId,
                        ShouldDecrementDesiredCapacity = false
                    });
                stopping = true;
            }
            catch (ScalingActivityInProgressException)
            {
                Console.WriteLine($"Scaling activity in progress for
{instanceId}. Waiting...");
                Thread.Sleep(10000);
            }
        }
    }

    /// <summary>
    /// Tries to delete the EC2 Auto Scaling group. If the group is in use or in
progress,
    /// waits and retries until the group is successfully deleted.
    /// </summary>
    /// <param name="groupName">The name of the group to try to delete.</param>
    /// <returns>Async task.</returns>
    public async Task TryDeleteGroupByName(string groupName)
    {
        var stopped = false;
        while (!stopped)
        {
            try
            {
                await _amazonAutoScaling.DeleteAutoScalingGroupAsync(
                    new DeleteAutoScalingGroupRequest()
                    {
                        AutoScalingGroupName = groupName
                    });
                stopped = true;
            }
            catch (Exception e)
                when ((e is ScalingActivityInProgressException)
                      || (e is Amazon.AutoScaling.Model.ResourceInUseException))
            {
```

```
                Console.WriteLine($"Some instances are still running.
Waiting...");
                Thread.Sleep(10000);
            }
        }
    }

    /// <summary>
    /// Terminate instances and delete the Auto Scaling group by name.
    /// </summary>
    /// <param name="groupName">The name of the group to delete.</param>
    /// <returns>Async task.</returns>
    public async Task TerminateAndDeleteAutoScalingGroupWithName(string
groupName)
    {
        var describeGroupsResponse = await
_amazonAutoScaling.DescribeAutoScalingGroupsAsync(
            new DescribeAutoScalingGroupsRequest()
            {
                AutoScalingGroupNames = new List<string>() { groupName }
            });
        if (describeGroupsResponse.AutoScalingGroups.Any())
        {
            // Update the size to 0.
            await _amazonAutoScaling.UpdateAutoScalingGroupAsync(
                new UpdateAutoScalingGroupRequest()
                {
                    AutoScalingGroupName = groupName,
                    MinSize = 0
                });
            var group = describeGroupsResponse.AutoScalingGroups[0];
            foreach (var instance in group.Instances)
            {
                await TryTerminateInstanceById(instance.InstanceId);
            }

            await TryDeleteGroupByName(groupName);
        }
        else
        {
            Console.WriteLine($"No groups found with name {groupName}.");
        }
    }
```

```
    /// <summary>
    /// Get the default security group for a specified Vpc.
    /// </summary>
    /// <param name="vpc">The Vpc to search.</param>
    /// <returns>The default security group.</returns>
    public async Task<SecurityGroup> GetDefaultSecurityGroupForVpc(Vpc vpc)
    {
        var groupResponse = await _amazonEc2.DescribeSecurityGroupsAsync(
            new DescribeSecurityGroupsRequest()
            {
                Filters = new List<Amazon.EC2.Model.Filter>()
                {
                    new ("group-name", new List<string>() { "default" }),
                    new ("vpc-id", new List<string>() { vpc.VpcId })
                }
            });
        return groupResponse.SecurityGroups[0];
    }

    /// <summary>
    /// Verify the default security group of a Vpc allows ingress from the
calling computer.
    /// This can be done by allowing ingress from this computer's IP address.
    /// In some situations, such as connecting from a corporate network, you must
instead specify
    /// a prefix list Id. You can also temporarily open the port to any IP
address while running this example.
    /// If you do, be sure to remove public access when you're done.
    /// </summary>
    /// <param name="vpc">The group to check.</param>
    /// <param name="port">The port to verify.</param>
    /// <param name="ipAddress">This computer's IP address.</param>
    /// <returns>True if the ip address is allowed on the group.</returns>
    public bool VerifyInboundPortForGroup(SecurityGroup group, int port, string
ipAddress)
    {
        var portIsOpen = false;
        foreach (var ipPermission in group.IpPermissions)
        {
            if (ipPermission.FromPort == port)
            {
                foreach (var ipRange in ipPermission.Ipv4Ranges)
                {
```

```
                    var cidr = ipRange.CidrIp;
                    if (cidr.StartsWith(ipAddress) || cidr == "0.0.0.0/0")
                    {
                        portIsOpen = true;
                    }
                }

                if (ipPermission.PrefixListIds.Any())
                {
                    portIsOpen = true;
                }

                if (!portIsOpen)
                {
                    Console.WriteLine("The inbound rule does not appear to be
open to either this computer's IP\n" +
                                      "address, to all IP addresses (0.0.0.0/0),
or to a prefix list ID.");
                }
                else
                {
                    break;
                }
            }
        }

        return portIsOpen;
    }

    /// <summary>
    /// Add an ingress rule to the specified security group that allows access on
the
    /// specified port from the specified IP address.
    /// </summary>
    /// <param name="groupId">The Id of the security group to modify.</param>
    /// <param name="port">The port to open.</param>
    /// <param name="ipAddress">The IP address to allow access.</param>
    /// <returns>Async task.</returns>
    public async Task OpenInboundPort(string groupId, int port, string ipAddress)
    {
        await _amazonEc2.AuthorizeSecurityGroupIngressAsync(
            new AuthorizeSecurityGroupIngressRequest()
            {
                GroupId = groupId,
```

```
                IpPermissions = new List<IpPermission>()
                {
                    new IpPermission()
                    {
                        FromPort = port,
                        ToPort = port,
                        IpProtocol = "tcp",
                        Ipv4Ranges = new List<IpRange>()
                        {
                            new IpRange() { CidrIp = $"{ipAddress}/32" }
                        }
                    }
                }
            });
    }

    /// <summary>
    /// Attaches an Elastic Load Balancing (ELB) target group to this EC2 Auto
 Scaling group.
    /// The
    /// </summary>
    /// <param name="autoScalingGroupName">The name of the Auto Scaling group.</
param>
    /// <param name="targetGroupArn">The Arn for the target group.</param>
    /// <returns>Async task.</returns>
    public async Task AttachLoadBalancerToGroup(string autoScalingGroupName,
 string targetGroupArn)
    {
        await _amazonAutoScaling.AttachLoadBalancerTargetGroupsAsync(
            new AttachLoadBalancerTargetGroupsRequest()
            {
                AutoScalingGroupName = autoScalingGroupName,
                TargetGroupARNs = new List<string>() { targetGroupArn }
            });
    }

    /// <summary>
    /// Wait until an EC2 instance is in a specified state.
    /// </summary>
    /// <param name="instanceId">The instance Id.</param>
    /// <param name="stateName">The state to wait for.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> WaitForInstanceState(string instanceId,
 InstanceStateName stateName)
```

```
    {
        var request = new DescribeInstancesRequest
        {
            InstanceIds = new List<string> { instanceId }
        };

        // Wait until the instance is in the specified state.
        var hasState = false;
        do
        {
            // Wait 5 seconds.
            Thread.Sleep(5000);

            // Check for the desired state.
            var response = await _amazonEc2.DescribeInstancesAsync(request);
            var instance = response.Reservations[0].Instances[0];
            hasState = instance.State.Name == stateName;
            Console.Write(". ");
        } while (!hasState);

        return hasState;
    }
}
```

Crea una classe che racchiuda le operazioni di Elastic Load Balancing.

```
/// <summary>
/// Encapsulates Elastic Load Balancer actions.
/// </summary>
public class ElasticLoadBalancerWrapper
{
    private readonly IAmazonElasticLoadBalancingV2 _amazonElasticLoadBalancingV2;
    private string? _endpoint = null;
    private readonly string _targetGroupName = "";
    private readonly string _loadBalancerName = "";
    HttpClient _httpClient = new();

    public string TargetGroupName => _targetGroupName;
    public string LoadBalancerName => _loadBalancerName;

    /// <summary>
```

```
    /// Constructor for the Elastic Load Balancer wrapper.
    /// </summary>
    /// <param name="amazonElasticLoadBalancingV2">The injected load balancing v2
client.</param>
    /// <param name="configuration">The injected configuration.</param>
    public ElasticLoadBalancerWrapper(
        IAmazonElasticLoadBalancingV2 amazonElasticLoadBalancingV2,
        IConfiguration configuration)
    {
        _amazonElasticLoadBalancingV2 = amazonElasticLoadBalancingV2;
        var prefix = configuration["resourcePrefix"];
        _targetGroupName = prefix + "-tg";
        _loadBalancerName = prefix + "-lb";
    }

    /// <summary>
    /// Get the HTTP Endpoint of a load balancer by its name.
    /// </summary>
    /// <param name="loadBalancerName">The name of the load balancer.</param>
    /// <returns>The HTTP endpoint.</returns>
    public async Task<string> GetEndpointForLoadBalancerByName(string
loadBalancerName)
    {
        if (_endpoint == null)
        {
            var endpointResponse =
                await _amazonElasticLoadBalancingV2.DescribeLoadBalancersAsync(
                    new DescribeLoadBalancersRequest()
                    {
                        Names = new List<string>() { loadBalancerName }
                    });
            _endpoint = endpointResponse.LoadBalancers[0].DNSName;
        }

        return _endpoint;
    }

    /// <summary>
    /// Return the GET response for an endpoint as text.
    /// </summary>
    /// <param name="endpoint">The endpoint for the request.</param>
    /// <returns>The request response.</returns>
    public async Task<string> GetEndPointResponse(string endpoint)
    {
```

```
        var endpointResponse = await _httpClient.GetAsync($"http://{endpoint}");
        var textResponse = await endpointResponse.Content.ReadAsStringAsync();
        return textResponse!;
    }

    /// <summary>
    /// Get the target health for a group by name.
    /// </summary>
    /// <param name="groupName">The name of the group.</param>
    /// <returns>The collection of health descriptions.</returns>
    public async Task<List<TargetHealthDescription>>
CheckTargetHealthForGroup(string groupName)
    {
        List<TargetHealthDescription> result = null!;
        try
        {
            var groupResponse =
                await _amazonElasticLoadBalancingV2.DescribeTargetGroupsAsync(
                    new DescribeTargetGroupsRequest()
                    {
                        Names = new List<string>() { groupName }
                    });
            var healthResponse =
                await _amazonElasticLoadBalancingV2.DescribeTargetHealthAsync(
                    new DescribeTargetHealthRequest()
                    {
                        TargetGroupArn =
groupResponse.TargetGroups[0].TargetGroupArn
                    });
            ;
            result = healthResponse.TargetHealthDescriptions;
        }
        catch (TargetGroupNotFoundException)
        {
            Console.WriteLine($"Target group {groupName} not found.");
        }
        return result;
    }

    /// <summary>
    /// Create an Elastic Load Balancing target group. The target group specifies
how the load balancer forwards
    /// requests to instances in the group and how instance health is checked.
    ///
```

```
    /// To speed up this demo, the health check is configured with shortened
 times and lower thresholds. In production,
    /// you might want to decrease the sensitivity of your health checks to avoid
 unwanted failures.
    /// </summary>
    /// <param name="groupName">The name for the group.</param>
    /// <param name="protocol">The protocol, such as HTTP.</param>
    /// <param name="port">The port to use to forward requests, such as 80.</
param>
    /// <param name="vpcId">The Id of the Vpc in which the load balancer
 exists.</param>
    /// <returns>The new TargetGroup object.</returns>
    public async Task<TargetGroup> CreateTargetGroupOnVpc(string groupName,
 ProtocolEnum protocol, int port, string vpcId)
    {
        var createResponse = await
 _amazonElasticLoadBalancingV2.CreateTargetGroupAsync(
            new CreateTargetGroupRequest()
            {
                Name = groupName,
                Protocol = protocol,
                Port = port,
                HealthCheckPath = "/healthcheck",
                HealthCheckIntervalSeconds = 10,
                HealthCheckTimeoutSeconds = 5,
                HealthyThresholdCount = 2,
                UnhealthyThresholdCount = 2,
                VpcId = vpcId
            });
        var targetGroup = createResponse.TargetGroups[0];
        return targetGroup;
    }

    /// <summary>
    /// Create an Elastic Load Balancing load balancer that uses the specified
 subnets
    /// and forwards requests to the specified target group.
    /// </summary>
    /// <param name="name">The name for the new load balancer.</param>
    /// <param name="subnetIds">Subnets for the load balancer.</param>
    /// <param name="targetGroup">Target group for forwarded requests.</param>
    /// <returns>The new LoadBalancer object.</returns>
    public async Task<LoadBalancer> CreateLoadBalancerAndListener(string name,
 List<string> subnetIds, TargetGroup targetGroup)
```

```
    {
        var createLbResponse = await
_amazonElasticLoadBalancingV2.CreateLoadBalancerAsync(
            new CreateLoadBalancerRequest()
            {
                Name = name,
                Subnets = subnetIds
            });
        var loadBalancerArn = createLbResponse.LoadBalancers[0].LoadBalancerArn;

        // Wait for load balancer to be available.
        var loadBalancerReady = false;
        while (!loadBalancerReady)
        {
            try
            {
                var describeResponse =
                    await
_amazonElasticLoadBalancingV2.DescribeLoadBalancersAsync(
                        new DescribeLoadBalancersRequest()
                        {
                            Names = new List<string>() { name }
                        });

                var loadBalancerState =
describeResponse.LoadBalancers[0].State.Code;

                loadBalancerReady = loadBalancerState ==
LoadBalancerStateEnum.Active;
            }
            catch (LoadBalancerNotFoundException)
            {
                loadBalancerReady = false;
            }
            Thread.Sleep(10000);
        }
        // Create the listener.
        await _amazonElasticLoadBalancingV2.CreateListenerAsync(
            new CreateListenerRequest()
            {
                LoadBalancerArn = loadBalancerArn,
                Protocol = targetGroup.Protocol,
                Port = targetGroup.Port,
                DefaultActions = new List<Action>()
```

```
                {
                    new Action()
                    {
                        Type = ActionTypeEnum.Forward,
                        TargetGroupArn = targetGroup.TargetGroupArn
                    }
                }
            });
        return createLbResponse.LoadBalancers[0];
    }

    /// <summary>
    /// Verify this computer can successfully send a GET request to the
    /// load balancer endpoint.
    /// </summary>
    /// <param name="endpoint">The endpoint to check.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> VerifyLoadBalancerEndpoint(string endpoint)
    {
        var success = false;
        var retries = 3;
        while (!success && retries > 0)
        {
            try
            {
                var endpointResponse = await _httpClient.GetAsync($"http://
{endpoint}");
                Console.WriteLine($"Response: {endpointResponse.StatusCode}.");

                if (endpointResponse.IsSuccessStatusCode)
                {
                    success = true;
                }
                else
                {
                    retries = 0;
                }
            }
            catch (HttpRequestException)
            {
                Console.WriteLine("Connection error, retrying...");
                retries--;
                Thread.Sleep(10000);
            }
```

```
        }

        return success;
    }

    /// <summary>
    /// Delete a load balancer by its specified name.
    /// </summary>
    /// <param name="name">The name of the load balancer to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteLoadBalancerByName(string name)
    {
        try
        {
            var describeLoadBalancerResponse =
                await _amazonElasticLoadBalancingV2.DescribeLoadBalancersAsync(
                    new DescribeLoadBalancersRequest()
                    {
                        Names = new List<string>() { name }
                    });
            var lbArn =
describeLoadBalancerResponse.LoadBalancers[0].LoadBalancerArn;
            await _amazonElasticLoadBalancingV2.DeleteLoadBalancerAsync(
                new DeleteLoadBalancerRequest()
                {
                    LoadBalancerArn = lbArn
                }
            );
        }
        catch (LoadBalancerNotFoundException)
        {
            Console.WriteLine($"Load balancer {name} not found.");
        }
    }

    /// <summary>
    /// Delete a TargetGroup by its specified name.
    /// </summary>
    /// <param name="groupName">Name of the group to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteTargetGroupByName(string groupName)
    {
        var done = false;
        while (!done)
```

```
        {
            try
            {
                var groupResponse =
                    await
 _amazonElasticLoadBalancingV2.DescribeTargetGroupsAsync(
                        new DescribeTargetGroupsRequest()
                        {
                            Names = new List<string>() { groupName }
                        });

                var targetArn = groupResponse.TargetGroups[0].TargetGroupArn;
                await _amazonElasticLoadBalancingV2.DeleteTargetGroupAsync(
                    new DeleteTargetGroupRequest() { TargetGroupArn =
 targetArn });
                Console.WriteLine($"Deleted load balancing target group
 {groupName}.");
                done = true;
            }
            catch (TargetGroupNotFoundException)
            {
                Console.WriteLine(
                    $"Target group {groupName} not found, could not delete.");
                done = true;
            }
            catch (ResourceInUseException)
            {
                Console.WriteLine("Target group not yet released, waiting...");
                Thread.Sleep(10000);
            }
        }
    }
}
```

Crea una classe che utilizzi DynamoDB per simulare un servizio di raccomandazione.

```
/// <summary>
/// Encapsulates a DynamoDB table to use as a service that recommends books,
 movies, and songs.
/// </summary>
public class Recommendations
{
```

```
    private readonly IAmazonDynamoDB _amazonDynamoDb;
    private readonly DynamoDBContext _context;
    private readonly string _tableName;

    public string TableName => _tableName;

    /// <summary>
    /// Constructor for the Recommendations service.
    /// </summary>
    /// <param name="amazonDynamoDb">The injected DynamoDb client.</param>
    /// <param name="configuration">The injected configuration.</param>
    public Recommendations(IAmazonDynamoDB amazonDynamoDb, IConfiguration
configuration)
    {
        _amazonDynamoDb = amazonDynamoDb;
        _context = new DynamoDBContext(_amazonDynamoDb);
        _tableName = configuration["databaseName"]!;
    }

    /// <summary>
    /// Create the DynamoDb table with a specified name.
    /// </summary>
    /// <param name="tableName">The name for the table.</param>
    /// <returns>True when ready.</returns>
    public async Task<bool> CreateDatabaseWithName(string tableName)
    {
        try
        {
            Console.Write($"Creating table {tableName}...");
            var createRequest = new CreateTableRequest()
            {
                TableName = tableName,
                AttributeDefinitions = new List<AttributeDefinition>()
                    {
                        new AttributeDefinition()
                        {
                            AttributeName = "MediaType",
                            AttributeType = ScalarAttributeType.S
                        },
                        new AttributeDefinition()
                        {
                            AttributeName = "ItemId",
                            AttributeType = ScalarAttributeType.N
                        }
```

```
                    },
                KeySchema = new List<KeySchemaElement>()
                    {
                        new KeySchemaElement()
                        {
                            AttributeName = "MediaType",
                            KeyType = KeyType.HASH
                        },
                        new KeySchemaElement()
                        {
                            AttributeName = "ItemId",
                            KeyType = KeyType.RANGE
                        }
                    },
                ProvisionedThroughput = new ProvisionedThroughput()
                {
                    ReadCapacityUnits = 5,
                    WriteCapacityUnits = 5
                }
            };
            await _amazonDynamoDb.CreateTableAsync(createRequest);

            // Wait until the table is ACTIVE and then report success.
            Console.Write("\nWaiting for table to become active...");

            var request = new DescribeTableRequest
            {
                TableName = tableName
            };

            TableStatus status;
            do
            {
                Thread.Sleep(2000);

                var describeTableResponse = await
_amazonDynamoDb.DescribeTableAsync(request);
                status = describeTableResponse.Table.TableStatus;

                Console.Write(".");
            }
            while (status != "ACTIVE");

            return status == TableStatus.ACTIVE;
```

```
        }
        catch (ResourceInUseException)
        {
            Console.WriteLine($"Table {tableName} already exists.");
            return false;
        }
    }

    /// <summary>
    /// Populate the database table with data from a specified path.
    /// </summary>
    /// <param name="databaseTableName">The name of the table.</param>
    /// <param name="recommendationsPath">The path of the recommendations data.</
param>
    /// <returns>Async task.</returns>
    public async Task PopulateDatabase(string databaseTableName, string
 recommendationsPath)
    {
        var recommendationsText = await
 File.ReadAllTextAsync(recommendationsPath);
        var records =

 JsonSerializer.Deserialize<RecommendationModel[]>(recommendationsText);
        var batchWrite = _context.CreateBatchWrite<RecommendationModel>();

        foreach (var record in records!)
        {
            batchWrite.AddPutItem(record);
        }

        await batchWrite.ExecuteAsync();
    }

    /// <summary>
    /// Delete the recommendation table by name.
    /// </summary>
    /// <param name="tableName">The name of the recommendation table.</param>
    /// <returns>Async task.</returns>
    public async Task DestroyDatabaseByName(string tableName)
    {
        try
        {
            await _amazonDynamoDb.DeleteTableAsync(
                new DeleteTableRequest() { TableName = tableName });
```

```
            Console.WriteLine($"Table {tableName} was deleted.");
        }
        catch (ResourceNotFoundException)
        {
            Console.WriteLine($"Table {tableName} not found");
        }
    }
}
```

Crea una classe che racchiuda le operazioni di Systems Manager.

```
/// <summary>
/// Encapsulates Systems Manager parameter operations. This example uses these
 parameters
/// to drive the demonstration of resilient architecture, such as failure of a
 dependency or
/// how the service responds to a health check.
/// </summary>
public class SmParameterWrapper
{
    private readonly IAmazonSimpleSystemsManagement
 _amazonSimpleSystemsManagement;

    private readonly string _tableParameter = "doc-example-resilient-
architecture-table";
    private readonly string _failureResponseParameter = "doc-example-resilient-
architecture-failure-response";
    private readonly string _healthCheckParameter = "doc-example-resilient-
architecture-health-check";
    private readonly string _tableName = "";

    public string TableParameter => _tableParameter;
    public string TableName => _tableName;
    public string HealthCheckParameter => _healthCheckParameter;
    public string FailureResponseParameter => _failureResponseParameter;

    /// <summary>
    /// Constructor for the SmParameterWrapper.
    /// </summary>
    /// <param name="amazonSimpleSystemsManagement">The injected Simple Systems
 Management client.</param>
    /// <param name="configuration">The injected configuration.</param>
```

```
    public SmParameterWrapper(IAmazonSimpleSystemsManagement
 amazonSimpleSystemsManagement, IConfiguration configuration)
    {
        _amazonSimpleSystemsManagement = amazonSimpleSystemsManagement;
        _tableName = configuration["databaseName"]!;
    }

    /// <summary>
    /// Reset the Systems Manager parameters to starting values for the demo.
    /// </summary>
    /// <returns>Async task.</returns>
    public async Task Reset()
    {
        await this.PutParameterByName(_tableParameter, _tableName);
        await this.PutParameterByName(_failureResponseParameter, "none");
        await this.PutParameterByName(_healthCheckParameter, "shallow");
    }

    /// <summary>
    /// Set the value of a named Systems Manager parameter.
    /// </summary>
    /// <param name="name">The name of the parameter.</param>
    /// <param name="value">The value to set.</param>
    /// <returns>Async task.</returns>
    public async Task PutParameterByName(string name, string value)
    {
        await _amazonSimpleSystemsManagement.PutParameterAsync(
            new PutParameterRequest() { Name = name, Value = value, Overwrite =
 true });
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per .NET .
  - AttachLoadBalancerTargetGroups
  - CreateAutoScalingGroup
  - CreateInstanceProfile
  - CreateLaunchTemplate
  - CreateListener
  - CreateLoadBalancer

- CreateTargetGroup
- DeleteAutoScalingGroup
- DeleteInstanceProfile
- DeleteLaunchTemplate
- DeleteLoadBalancer
- DeleteTargetGroup
- DescribeAutoScalingGroups
- DescribeAvailabilityZones
- DescribeIamInstanceProfileAssociations
- DescribeInstances
- DescribeLoadBalancers
- DescribeSubnets
- DescribeTargetGroups
- DescribeTargetHealth
- DescribeVpcs
- RebootInstances
- ReplaceIamInstanceProfileAssociation
- TerminateInstanceInAutoScalingGroup
- UpdateAutoScalingGroup

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Esegui lo scenario interattivo al prompt dei comandi.

```
public class Main {
```

```
    public static final String fileName = "C:\\AWS\\resworkflow\
\recommendations.json"; // Modify file location.
    public static final String tableName = "doc-example-recommendation-service";
    public static final String startScript = "C:\\AWS\\resworkflow\
\server_startup_script.sh"; // Modify file location.
    public static final String policyFile = "C:\\AWS\\resworkflow\
\instance_policy.json"; // Modify file location.
    public static final String ssmJSON = "C:\\AWS\\resworkflow\
\ssm_only_policy.json"; // Modify file location.
    public static final String failureResponse = "doc-example-resilient-
architecture-failure-response";
    public static final String healthCheck = "doc-example-resilient-architecture-
health-check";
    public static final String templateName = "doc-example-resilience-template";
    public static final String roleName = "doc-example-resilience-role";
    public static final String policyName = "doc-example-resilience-pol";
    public static final String profileName = "doc-example-resilience-prof";

    public static final String badCredsProfileName = "doc-example-resilience-
prof-bc";

    public static final String targetGroupName = "doc-example-resilience-tg";
    public static final String autoScalingGroupName = "doc-example-resilience-
group";
    public static final String lbName = "doc-example-resilience-lb";
    public static final String protocol = "HTTP";
    public static final int port = 80;

    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws IOException,
 InterruptedException {
        Scanner in = new Scanner(System.in);
        Database database = new Database();
        AutoScaler autoScaler = new AutoScaler();
        LoadBalancer loadBalancer = new LoadBalancer();

        System.out.println(DASHES);
        System.out.println("Welcome to the demonstration of How to Build and
 Manage a Resilient Service!");
        System.out.println(DASHES);

        System.out.println(DASHES);
```

```
        System.out.println("A - SETUP THE RESOURCES");
        System.out.println("Press Enter when you're ready to start deploying
resources.");
        in.nextLine();
        deploy(loadBalancer);
        System.out.println(DASHES);
        System.out.println(DASHES);
        System.out.println("B - DEMO THE RESILIENCE FUNCTIONALITY");
        System.out.println("Press Enter when you're ready.");
        in.nextLine();
        demo(loadBalancer);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("C - DELETE THE RESOURCES");
        System.out.println("""
                This concludes the demo of how to build and manage a resilient
service.
                To keep things tidy and to avoid unwanted charges on your
account, we can clean up all AWS resources
                that were created for this demo.
                """);

        System.out.println("\n Do you want to delete the resources (y/n)? ");
        String userInput = in.nextLine().trim().toLowerCase(); // Capture user
input

        if (userInput.equals("y")) {
            // Delete resources here
            deleteResources(loadBalancer, autoScaler, database);
            System.out.println("Resources deleted.");
        } else {
            System.out.println("""
                    Okay, we'll leave the resources intact.
                    Don't forget to delete them when you're done with them or you
might incur unexpected charges.
                    """);
        }
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The example has completed. ");
        System.out.println("\n Thanks for watching!");
        System.out.println(DASHES);
```

```
   }

   // Deletes the AWS resources used in this example.
   private static void deleteResources(LoadBalancer loadBalancer, AutoScaler
autoScaler, Database database)
           throws IOException, InterruptedException {
       loadBalancer.deleteLoadBalancer(lbName);
       System.out.println("*** Wait 30 secs for resource to be deleted");
       TimeUnit.SECONDS.sleep(30);
       loadBalancer.deleteTargetGroup(targetGroupName);
       autoScaler.deleteAutoScaleGroup(autoScalingGroupName);
       autoScaler.deleteRolesPolicies(policyName, roleName, profileName);
       autoScaler.deleteTemplate(templateName);
       database.deleteTable(tableName);
   }

   private static void deploy(LoadBalancer loadBalancer) throws
InterruptedException, IOException {
       Scanner in = new Scanner(System.in);
       System.out.println(
               """
                       For this demo, we'll use the AWS SDK for Java (v2) to
create several AWS resources
                       to set up a load-balanced web service endpoint and
explore some ways to make it resilient
                       against various kinds of failures.

                       Some of the resources create by this demo are:
                       \t* A DynamoDB table that the web service depends on to
provide book, movie, and song recommendations.
                       \t* An EC2 launch template that defines EC2 instances
that each contain a Python web server.
                       \t* An EC2 Auto Scaling group that manages EC2 instances
across several Availability Zones.
                       \t* An Elastic Load Balancing (ELB) load balancer that
targets the Auto Scaling group to distribute requests.
                       """);

       System.out.println("Press Enter when you're ready.");
       in.nextLine();
       System.out.println(DASHES);

       System.out.println(DASHES);
```

```
        System.out.println("Creating and populating a DynamoDB table named " +
tableName);
        Database database = new Database();
        database.createTable(tableName, fileName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("""
                Creating an EC2 launch template that runs '{startup_script}' when
an instance starts.
                This script starts a Python web server defined in the `server.py`
script. The web server
                listens to HTTP requests on port 80 and responds to requests to
'/' and to '/healthcheck'.
                For demo purposes, this server is run as the root user. In
production, the best practice is to
                run a web server, such as Apache, with least-privileged
credentials.

                The template also defines an IAM policy that each instance uses
to assume a role that grants
                permissions to access the DynamoDB recommendation table and
Systems Manager parameters
                that control the flow of the demo.
                """);

        LaunchTemplateCreator templateCreator = new LaunchTemplateCreator();
        templateCreator.createTemplate(policyFile, policyName, profileName,
startScript, templateName, roleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "Creating an EC2 Auto Scaling group that maintains three EC2
instances, each in a different Availability Zone.");
        System.out.println("*** Wait 30 secs for the VPC to be created");
        TimeUnit.SECONDS.sleep(30);
        AutoScaler autoScaler = new AutoScaler();
        String[] zones = autoScaler.createGroup(3, templateName,
autoScalingGroupName);

        System.out.println("""
                At this point, you have EC2 instances created. Once each instance
starts, it listens for
```

```
                HTTP requests. You can see these instances in the console or
continue with the demo.
                Press Enter when you're ready to continue.
                """);

        in.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Creating variables that control the flow of the
demo.");
        ParameterHelper paramHelper = new ParameterHelper();
        paramHelper.reset();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("""
                Creating an Elastic Load Balancing target group and load
balancer. The target group
                defines how the load balancer connects to instances. The load
balancer provides a
                single endpoint where clients connect and dispatches requests to
instances in the group.
                """);

        String vpcId = autoScaler.getDefaultVPC();
        List<Subnet> subnets = autoScaler.getSubnets(vpcId, zones);
        System.out.println("You have retrieved a list with " + subnets.size() + "
subnets");
        String targetGroupArn = loadBalancer.createTargetGroup(protocol, port,
vpcId, targetGroupName);
        String elbDnsName = loadBalancer.createLoadBalancer(subnets,
targetGroupArn, lbName, port, protocol);
        autoScaler.attachLoadBalancerTargetGroup(autoScalingGroupName,
targetGroupArn);
        System.out.println("Verifying access to the load balancer endpoint...");
        boolean wasSuccessul =
loadBalancer.verifyLoadBalancerEndpoint(elbDnsName);
        if (!wasSuccessul) {
            System.out.println("Couldn't connect to the load balancer, verifying
that the port is open...");
            CloseableHttpClient httpClient = HttpClients.createDefault();

            // Create an HTTP GET request to "http://checkip.amazonaws.com"
```

```
            HttpGet httpGet = new HttpGet("http://checkip.amazonaws.com");
            try {
                // Execute the request and get the response
                HttpResponse response = httpClient.execute(httpGet);

                // Read the response content.
                String ipAddress =
IOUtils.toString(response.getEntity().getContent(),
StandardCharsets.UTF_8).trim();

                // Print the public IP address.
                System.out.println("Public IP Address: " + ipAddress);
                GroupInfo groupInfo = autoScaler.verifyInboundPort(vpcId, port,
ipAddress);
                if (!groupInfo.isPortOpen()) {
                    System.out.println("""
                            For this example to work, the default security group
for your default VPC must
                            allow access from this computer. You can either add
it automatically from this
                            example or add it yourself using the AWS Management
Console.
                            """);

                    System.out.println(
                            "Do you want to add a rule to security group " +
groupInfo.getGroupName() + " to allow");
                    System.out.println("inbound traffic on port " + port + " from
your computer's IP address (y/n) ");
                    String ans = in.nextLine();
                    if ("y".equalsIgnoreCase(ans)) {
                        autoScaler.openInboundPort(groupInfo.getGroupName(),
String.valueOf(port), ipAddress);
                        System.out.println("Security group rule added.");
                    } else {
                        System.out.println("No security group rule added.");
                    }
                }

            } catch (AutoScalingException e) {
                e.printStackTrace();
            }
        } else if (wasSuccessul) {
```

```
            System.out.println("Your load balancer is ready. You can access it by
browsing to:");
            System.out.println("\t http://" + elbDnsName);
        } else {
            System.out.println("Couldn't get a successful response from the load
balancer endpoint. Troubleshoot by");
            System.out.println("manually verifying that your VPC and security
group are configured correctly and that");
            System.out.println("you can successfully make a GET request to the
load balancer.");
        }

        System.out.println("Press Enter when you're ready to continue with the
demo.");
        in.nextLine();
    }

    // A method that controls the demo part of the Java program.
    public static void demo(LoadBalancer loadBalancer) throws IOException,
InterruptedException {
        ParameterHelper paramHelper = new ParameterHelper();
        System.out.println("Read the ssm_only_policy.json file");
        String ssmOnlyPolicy = readFileAsString(ssmJSON);

        System.out.println("Resetting parameters to starting values for demo.");
        paramHelper.reset();

        System.out.println(
                """
                        This part of the demonstration shows how to toggle
different parts of the system
                        to create situations where the web service fails, and
shows how using a resilient
                        architecture can keep the web service running in spite
of these failures.

                        At the start, the load balancer endpoint returns
recommendations and reports that all targets are healthy.
                       """);
        demoChoices(loadBalancer);

        System.out.println(
                """
```

```
                        The web service running on the EC2 instances gets
recommendations by querying a DynamoDB table.
                        The table name is contained in a Systems Manager
parameter named self.param_helper.table.
                        To simulate a failure of the recommendation service,
let's set this parameter to name a non-existent table.
                       """);
       paramHelper.put(paramHelper.tableName, "this-is-not-a-table");

       System.out.println(
               """
                        \nNow, sending a GET request to the load balancer
endpoint returns a failure code. But, the service reports as
                        healthy to the load balancer because shallow health
checks don't check for failure of the recommendation service.
                       """);
       demoChoices(loadBalancer);

       System.out.println(
               """
                       Instead of failing when the recommendation service fails,
the web service can return a static response.
                       While this is not a perfect solution, it presents the
customer with a somewhat better experience than failure.
                       """);
       paramHelper.put(paramHelper.failureResponse, "static");

       System.out.println("""
               Now, sending a GET request to the load balancer endpoint returns
a static response.
               The service still reports as healthy because health checks are
still shallow.
               """);
       demoChoices(loadBalancer);

       System.out.println("Let's reinstate the recommendation service.");
       paramHelper.put(paramHelper.tableName, paramHelper.dyntable);

       System.out.println("""
               Let's also substitute bad credentials for one of the instances in
the target group so that it can't
               access the DynamoDB recommendation table. We will get an instance
id value.
               """);
```

```
        LaunchTemplateCreator templateCreator = new LaunchTemplateCreator();
        AutoScaler autoScaler = new AutoScaler();

        // Create a new instance profile based on badCredsProfileName.
        templateCreator.createInstanceProfile(policyFile, policyName,
badCredsProfileName, roleName);
        String badInstanceId = autoScaler.getBadInstance(autoScalingGroupName);
        System.out.println("The bad instance id values used for this demo is " +
badInstanceId);

        String profileAssociationId =
autoScaler.getInstanceProfile(badInstanceId);
        System.out.println("The association Id value is " +
profileAssociationId);
        System.out.println("Replacing the profile for instance " + badInstanceId
                + " with a profile that contains bad credentials");
        autoScaler.replaceInstanceProfile(badInstanceId, badCredsProfileName,
profileAssociationId);

        System.out.println(
                """
                        Now, sending a GET request to the load balancer endpoint
returns either a recommendation or a static response,
                        depending on which instance is selected by the load
balancer.
                        """);

        demoChoices(loadBalancer);

        System.out.println("""
                Let's implement a deep health check. For this demo, a deep health
check tests whether
                the web service can access the DynamoDB table that it depends on
for recommendations. Note that
                the deep health check is only for ELB routing and not for Auto
Scaling instance health.
                This kind of deep health check is not recommended for Auto
Scaling instance health, because it
                risks accidental termination of all instances in the Auto Scaling
group when a dependent service fails.
                """);

        System.out.println("""
```

```
                By implementing deep health checks, the load balancer can detect
when one of the instances is failing
                and take that instance out of rotation.
                """);

        paramHelper.put(paramHelper.healthCheck, "deep");

        System.out.println("""
                Now, checking target health indicates that the instance with bad
credentials
                is unhealthy. Note that it might take a minute or two for the
load balancer to detect the unhealthy
                instance. Sending a GET request to the load balancer endpoint
always returns a recommendation, because
                the load balancer takes unhealthy instances out of its rotation.
                """);

        demoChoices(loadBalancer);

        System.out.println(
                """
                        Because the instances in this demo are controlled by an
auto scaler, the simplest way to fix an unhealthy
                        instance is to terminate it and let the auto scaler start
a new instance to replace it.
                        """);
        autoScaler.terminateInstance(badInstanceId);

        System.out.println("""
                Even while the instance is terminating and the new instance is
starting, sending a GET
                request to the web service continues to get a successful
recommendation response because
                the load balancer routes requests to the healthy instances. After
the replacement instance
                starts and reports as healthy, it is included in the load
balancing rotation.
                Note that terminating and replacing an instance typically takes
several minutes, during which time you
                can see the changing health check status until the new instance
is running and healthy.
                """);

        demoChoices(loadBalancer);
```

```
        System.out.println(
                "If the recommendation service fails now, deep health checks mean
all instances report as unhealthy.");
        paramHelper.put(paramHelper.tableName, "this-is-not-a-table");

        demoChoices(loadBalancer);
        paramHelper.reset();
    }

    public static void demoChoices(LoadBalancer loadBalancer) throws IOException,
InterruptedException {
        String[] actions = {
                "Send a GET request to the load balancer endpoint.",
                "Check the health of load balancer targets.",
                "Go to the next part of the demo."
        };
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("-".repeat(88));
            System.out.println("See the current state of the service by selecting
one of the following choices:");
            for (int i = 0; i < actions.length; i++) {
                System.out.println(i + ": " + actions[i]);
            }

            try {
                System.out.print("\nWhich action would you like to take? ");
                int choice = scanner.nextInt();
                System.out.println("-".repeat(88));

                switch (choice) {
                    case 0 -> {
                        System.out.println("Request:\n");
                        System.out.println("GET http://" +
loadBalancer.getEndpoint(lbName));
                        CloseableHttpClient httpClient =
HttpClients.createDefault();

                        // Create an HTTP GET request to the ELB.
                        HttpGet httpGet = new HttpGet("http://" +
loadBalancer.getEndpoint(lbName));

                        // Execute the request and get the response.
```

```
                        HttpResponse response = httpClient.execute(httpGet);
                        int statusCode =
response.getStatusLine().getStatusCode();
                        System.out.println("HTTP Status Code: " + statusCode);

                        // Display the JSON response
                        BufferedReader reader = new BufferedReader(
                                new
InputStreamReader(response.getEntity().getContent()));
                        StringBuilder jsonResponse = new StringBuilder();
                        String line;
                        while ((line = reader.readLine()) != null) {
                            jsonResponse.append(line);
                        }
                        reader.close();

                        // Print the formatted JSON response.
                        System.out.println("Full Response:\n");
                        System.out.println(jsonResponse.toString());

                        // Close the HTTP client.
                        httpClient.close();

                    }
                    case 1 -> {
                        System.out.println("\nChecking the health of load
balancer targets:\n");
                        List<TargetHealthDescription> health =
loadBalancer.checkTargetHealth(targetGroupName);
                        for (TargetHealthDescription target : health) {
                            System.out.printf("\tTarget %s on port %d is %s%n",
target.target().id(),
                                    target.target().port(),
target.targetHealth().stateAsString());
                        }
                        System.out.println("""
                                Note that it can take a minute or two for the
health check to update
                                after changes are made.
                                """);
                    }
                    case 2 -> {
                        System.out.println("\nOkay, let's move on.");
                        System.out.println("-".repeat(88));
```

```
                        return; // Exit the method when choice is 2
                    }
                    default -> System.out.println("You must choose a value
 between 0-2. Please select again.");
                }

            } catch (java.util.InputMismatchException e) {
                System.out.println("Invalid input. Please select again.");
                scanner.nextLine(); // Clear the input buffer.
            }
        }
    }

    public static String readFileAsString(String filePath) throws IOException {
        byte[] bytes = Files.readAllBytes(Paths.get(filePath));
        return new String(bytes);
    }
}
```

Crea una classe che racchiuda le operazioni di dimensionamento automatico e Amazon EC2.

```
public class AutoScaler {

    private static Ec2Client ec2Client;
    private static AutoScalingClient autoScalingClient;
    private static IamClient iamClient;

    private static SsmClient ssmClient;

    private IamClient getIAMClient() {
        if (iamClient == null) {
            iamClient = IamClient.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }
        return iamClient;
    }

    private SsmClient getSSMClient() {
        if (ssmClient == null) {
            ssmClient = SsmClient.builder()
                    .region(Region.US_EAST_1)
```

```
                    .build();
        }
        return ssmClient;
    }

    private Ec2Client getEc2Client() {
        if (ec2Client == null) {
            ec2Client = Ec2Client.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }
        return ec2Client;
    }

    private AutoScalingClient getAutoScalingClient() {
        if (autoScalingClient == null) {
            autoScalingClient = AutoScalingClient.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }
        return autoScalingClient;
    }

    /**
     * Terminates and instances in an EC2 Auto Scaling group. After an instance
is
     * terminated, it can no longer be accessed.
     */
    public void terminateInstance(String instanceId) {
        TerminateInstanceInAutoScalingGroupRequest terminateInstanceIRequest =
TerminateInstanceInAutoScalingGroupRequest
                .builder()
                .instanceId(instanceId)
                .shouldDecrementDesiredCapacity(false)
                .build();


getAutoScalingClient().terminateInstanceInAutoScalingGroup(terminateInstanceIRequest);
        System.out.format("Terminated instance %s.", instanceId);
    }

    /**
     * Replaces the profile associated with a running instance. After the profile
is
```

```
     * replaced, the instance is rebooted to ensure that it uses the new profile.
     * When
     * the instance is ready, Systems Manager is used to restart the Python web
     * server.
     */
    public void replaceInstanceProfile(String instanceId, String
newInstanceProfileName, String profileAssociationId)
            throws InterruptedException {
        // Create an IAM instance profile specification.
        software.amazon.awssdk.services.ec2.model.IamInstanceProfileSpecification
iamInstanceProfile =
software.amazon.awssdk.services.ec2.model.IamInstanceProfileSpecification
                .builder()
                .name(newInstanceProfileName) // Make sure
'newInstanceProfileName' is a valid IAM Instance Profile
                                              // name.
                .build();

        // Replace the IAM instance profile association for the EC2 instance.
        ReplaceIamInstanceProfileAssociationRequest replaceRequest =
ReplaceIamInstanceProfileAssociationRequest
                .builder()
                .iamInstanceProfile(iamInstanceProfile)
                .associationId(profileAssociationId) // Make sure
'profileAssociationId' is a valid association ID.
                .build();

        try {
            getEc2Client().replaceIamInstanceProfileAssociation(replaceRequest);
            // Handle the response as needed.
        } catch (Ec2Exception e) {
            // Handle exceptions, log, or report the error.
            System.err.println("Error: " + e.getMessage());
        }
        System.out.format("Replaced instance profile for association %s with
profile %s.", profileAssociationId,
                newInstanceProfileName);
        TimeUnit.SECONDS.sleep(15);
        boolean instReady = false;
        int tries = 0;

        // Reboot after 60 seconds
        while (!instReady) {
            if (tries % 6 == 0) {
```

```
                getEc2Client().rebootInstances(RebootInstancesRequest.builder()
                        .instanceIds(instanceId)
                        .build());
                System.out.println("Rebooting instance " + instanceId + " and
waiting for it to be ready.");
            }
            tries++;
            try {
                TimeUnit.SECONDS.sleep(10);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }

            DescribeInstanceInformationResponse informationResponse =
getSSMClient().describeInstanceInformation();
            List<InstanceInformation> instanceInformationList =
informationResponse.instanceInformationList();
            for (InstanceInformation info : instanceInformationList) {
                if (info.instanceId().equals(instanceId)) {
                    instReady = true;
                    break;
                }
            }
        }

        SendCommandRequest sendCommandRequest = SendCommandRequest.builder()
                .instanceIds(instanceId)
                .documentName("AWS-RunShellScript")
                .parameters(Collections.singletonMap("commands",
                        Collections.singletonList("cd / && sudo python3 server.py
80")))
                .build();

        getSSMClient().sendCommand(sendCommandRequest);
        System.out.println("Restarted the Python web server on instance " +
instanceId + ".");
    }

    public void openInboundPort(String secGroupId, String port, String ipAddress)
{
        AuthorizeSecurityGroupIngressRequest ingressRequest =
AuthorizeSecurityGroupIngressRequest.builder()
                .groupName(secGroupId)
                .cidrIp(ipAddress)
```

```
                .fromPort(Integer.parseInt(port))
                .build();

        getEc2Client().authorizeSecurityGroupIngress(ingressRequest);
        System.out.format("Authorized ingress to %s on port %s from %s.",
secGroupId, port, ipAddress);
    }

    /**
     * Detaches a role from an instance profile, detaches policies from the role,
     * and deletes all the resources.
     */
    public void deleteInstanceProfile(String roleName, String profileName) {
        try {
            software.amazon.awssdk.services.iam.model.GetInstanceProfileRequest
getInstanceProfileRequest =
software.amazon.awssdk.services.iam.model.GetInstanceProfileRequest
                    .builder()
                    .instanceProfileName(profileName)
                    .build();

            GetInstanceProfileResponse response =
getIAMClient().getInstanceProfile(getInstanceProfileRequest);
            String name = response.instanceProfile().instanceProfileName();
            System.out.println(name);

            RemoveRoleFromInstanceProfileRequest profileRequest =
RemoveRoleFromInstanceProfileRequest.builder()
                    .instanceProfileName(profileName)
                    .roleName(roleName)
                    .build();

            getIAMClient().removeRoleFromInstanceProfile(profileRequest);
            DeleteInstanceProfileRequest deleteInstanceProfileRequest =
DeleteInstanceProfileRequest.builder()
                    .instanceProfileName(profileName)
                    .build();

            getIAMClient().deleteInstanceProfile(deleteInstanceProfileRequest);
            System.out.println("Deleted instance profile " + profileName);

            DeleteRoleRequest deleteRoleRequest = DeleteRoleRequest.builder()
                    .roleName(roleName)
                    .build();
```

```
            // List attached role policies.
            ListAttachedRolePoliciesResponse rolesResponse = getIAMClient()
                    .listAttachedRolePolicies(role -> role.roleName(roleName));
            List<AttachedPolicy> attachedPolicies =
rolesResponse.attachedPolicies();
            for (AttachedPolicy attachedPolicy : attachedPolicies) {
                DetachRolePolicyRequest request =
DetachRolePolicyRequest.builder()
                        .roleName(roleName)
                        .policyArn(attachedPolicy.policyArn())
                        .build();

                getIAMClient().detachRolePolicy(request);
                System.out.println("Detached and deleted policy " +
attachedPolicy.policyName());
            }

            getIAMClient().deleteRole(deleteRoleRequest);
            System.out.println("Instance profile and role deleted.");

        } catch (IamException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public void deleteTemplate(String templateName) {
        getEc2Client().deleteLaunchTemplate(name ->
name.launchTemplateName(templateName));
        System.out.format(templateName + " was deleted.");
    }

    public void deleteAutoScaleGroup(String groupName) {
        DeleteAutoScalingGroupRequest deleteAutoScalingGroupRequest =
DeleteAutoScalingGroupRequest.builder()
                .autoScalingGroupName(groupName)
                .forceDelete(true)
                .build();


getAutoScalingClient().deleteAutoScalingGroup(deleteAutoScalingGroupRequest);
        System.out.println(groupName + " was deleted.");
    }
```

```
   /*
    * Verify the default security group of the specified VPC allows ingress from
    * this
    * computer. This can be done by allowing ingress from this computer's IP
    * address. In some situations, such as connecting from a corporate network,
you
    * must instead specify a prefix list ID. You can also temporarily open the
port
    * to
    * any IP address while running this example. If you do, be sure to remove
    * public
    * access when you're done.
    *
    */
   public GroupInfo verifyInboundPort(String VPC, int port, String ipAddress) {
       boolean portIsOpen = false;
       GroupInfo groupInfo = new GroupInfo();
       try {
           Filter filter = Filter.builder()
                   .name("group-name")
                   .values("default")
                   .build();

           Filter filter1 = Filter.builder()
                   .name("vpc-id")
                   .values(VPC)
                   .build();

           DescribeSecurityGroupsRequest securityGroupsRequest =
DescribeSecurityGroupsRequest.builder()
                   .filters(filter, filter1)
                   .build();

           DescribeSecurityGroupsResponse securityGroupsResponse =
getEc2Client()
                   .describeSecurityGroups(securityGroupsRequest);
           String securityGroup =
securityGroupsResponse.securityGroups().get(0).groupName();
           groupInfo.setGroupName(securityGroup);

           for (SecurityGroup secGroup :
securityGroupsResponse.securityGroups()) {
```

```
                System.out.println("Found security group: " +
secGroup.groupId());

                for (IpPermission ipPermission : secGroup.ipPermissions()) {
                    if (ipPermission.fromPort() == port) {
                        System.out.println("Found inbound rule: " +
ipPermission);
                        for (IpRange ipRange : ipPermission.ipRanges()) {
                            String cidrIp = ipRange.cidrIp();
                            if (cidrIp.startsWith(ipAddress) ||
cidrIp.equals("0.0.0.0/0")) {
                                System.out.println(cidrIp + " is applicable");
                                portIsOpen = true;
                            }
                        }

                        if (!ipPermission.prefixListIds().isEmpty()) {
                            System.out.println("Prefix lList is applicable");
                            portIsOpen = true;
                        }

                        if (!portIsOpen) {
                            System.out
                                    .println("The inbound rule does not appear to
be open to either this computer's IP,"
                                            + " all IP addresses (0.0.0.0/0), or
to a prefix list ID.");
                        } else {
                            break;
                        }
                    }
                }
            }

        } catch (AutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }

        groupInfo.setPortOpen(portIsOpen);
        return groupInfo;
    }

    /*
     * Attaches an Elastic Load Balancing (ELB) target group to this EC2 Auto
```

```
    * Scaling group.
    * The target group specifies how the load balancer forward requests to the
    * instances
    * in the group.
    */
   public void attachLoadBalancerTargetGroup(String asGroupName, String
targetGroupARN) {
       try {
           AttachLoadBalancerTargetGroupsRequest targetGroupsRequest =
AttachLoadBalancerTargetGroupsRequest.builder()
                   .autoScalingGroupName(asGroupName)
                   .targetGroupARNs(targetGroupARN)
                   .build();


getAutoScalingClient().attachLoadBalancerTargetGroups(targetGroupsRequest);
           System.out.println("Attached load balancer to " + asGroupName);

       } catch (AutoScalingException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }

   // Creates an EC2 Auto Scaling group with the specified size.
   public String[] createGroup(int groupSize, String templateName, String
autoScalingGroupName) {

       // Get availability zones.

software.amazon.awssdk.services.ec2.model.DescribeAvailabilityZonesRequest
zonesRequest =
software.amazon.awssdk.services.ec2.model.DescribeAvailabilityZonesRequest
               .builder()
               .build();

       DescribeAvailabilityZonesResponse zonesResponse =
getEc2Client().describeAvailabilityZones(zonesRequest);
       List<String> availabilityZoneNames =
zonesResponse.availabilityZones().stream()

.map(software.amazon.awssdk.services.ec2.model.AvailabilityZone::zoneName)
               .collect(Collectors.toList());
```

```
        String availabilityZones = String.join(",", availabilityZoneNames);
        LaunchTemplateSpecification specification =
LaunchTemplateSpecification.builder()
                .launchTemplateName(templateName)
                .version("$Default")
                .build();

        String[] zones = availabilityZones.split(",");
        CreateAutoScalingGroupRequest groupRequest =
CreateAutoScalingGroupRequest.builder()
                .launchTemplate(specification)
                .availabilityZones(zones)
                .maxSize(groupSize)
                .minSize(groupSize)
                .autoScalingGroupName(autoScalingGroupName)
                .build();

        try {
            getAutoScalingClient().createAutoScalingGroup(groupRequest);

        } catch (AutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Created an EC2 Auto Scaling group named " +
autoScalingGroupName);
        return zones;
    }

    public String getDefaultVPC() {
        // Define the filter.
        Filter defaultFilter = Filter.builder()
                .name("is-default")
                .values("true")
                .build();

        software.amazon.awssdk.services.ec2.model.DescribeVpcsRequest request =
software.amazon.awssdk.services.ec2.model.DescribeVpcsRequest
                .builder()
                .filters(defaultFilter)
                .build();

        DescribeVpcsResponse response = getEc2Client().describeVpcs(request);
        return response.vpcs().get(0).vpcId();
```

```
    }

    // Gets the default subnets in a VPC for a specified list of Availability
Zones.
    public List<Subnet> getSubnets(String vpcId, String[] availabilityZones) {
        List<Subnet> subnets = null;
        Filter vpcFilter = Filter.builder()
                .name("vpc-id")
                .values(vpcId)
                .build();

        Filter azFilter = Filter.builder()
                .name("availability-zone")
                .values(availabilityZones)
                .build();

        Filter defaultForAZ = Filter.builder()
                .name("default-for-az")
                .values("true")
                .build();

        DescribeSubnetsRequest request = DescribeSubnetsRequest.builder()
                .filters(vpcFilter, azFilter, defaultForAZ)
                .build();

        DescribeSubnetsResponse response =
getEc2Client().describeSubnets(request);
        subnets = response.subnets();
        return subnets;
    }

    // Gets data about the instances in the EC2 Auto Scaling group.
    public String getBadInstance(String groupName) {
        DescribeAutoScalingGroupsRequest request =
DescribeAutoScalingGroupsRequest.builder()
                .autoScalingGroupNames(groupName)
                .build();

        DescribeAutoScalingGroupsResponse response =
getAutoScalingClient().describeAutoScalingGroups(request);
        AutoScalingGroup autoScalingGroup = response.autoScalingGroups().get(0);
        List<String> instanceIds = autoScalingGroup.instances().stream()
                .map(instance -> instance.instanceId())
                .collect(Collectors.toList());
```

```
        String[] instanceIdArray = instanceIds.toArray(new String[0]);
        for (String instanceId : instanceIdArray) {
            System.out.println("Instance ID: " + instanceId);
            return instanceId;
        }
        return "";
    }

    // Gets data about the profile associated with an instance.
    public String getInstanceProfile(String instanceId) {
        Filter filter = Filter.builder()
                .name("instance-id")
                .values(instanceId)
                .build();

        DescribeIamInstanceProfileAssociationsRequest associationsRequest =
DescribeIamInstanceProfileAssociationsRequest
                .builder()
                .filters(filter)
                .build();

        DescribeIamInstanceProfileAssociationsResponse response = getEc2Client()
                .describeIamInstanceProfileAssociations(associationsRequest);
        return response.iamInstanceProfileAssociations().get(0).associationId();
    }

    public void deleteRolesPolicies(String policyName, String roleName, String
InstanceProfile) {
        ListPoliciesRequest listPoliciesRequest =
ListPoliciesRequest.builder().build();
        ListPoliciesResponse listPoliciesResponse =
getIAMClient().listPolicies(listPoliciesRequest);
        for (Policy policy : listPoliciesResponse.policies()) {
            if (policy.policyName().equals(policyName)) {
                // List the entities (users, groups, roles) that are attached to
the policy.

software.amazon.awssdk.services.iam.model.ListEntitiesForPolicyRequest
listEntitiesRequest =
software.amazon.awssdk.services.iam.model.ListEntitiesForPolicyRequest
                        .builder()
                        .policyArn(policy.arn())
                        .build();
```

```
                ListEntitiesForPolicyResponse listEntitiesResponse = iamClient
                        .listEntitiesForPolicy(listEntitiesRequest);
                if (!listEntitiesResponse.policyGroups().isEmpty() || !
listEntitiesResponse.policyUsers().isEmpty()
                        || !listEntitiesResponse.policyRoles().isEmpty()) {
                    // Detach the policy from any entities it is attached to.
                    DetachRolePolicyRequest detachPolicyRequest =
 DetachRolePolicyRequest.builder()
                            .policyArn(policy.arn())
                            .roleName(roleName) // Specify the name of the IAM
 role
                            .build();

                    getIAMClient().detachRolePolicy(detachPolicyRequest);
                    System.out.println("Policy detached from entities.");
                }

                // Now, you can delete the policy.
                DeletePolicyRequest deletePolicyRequest =
 DeletePolicyRequest.builder()
                        .policyArn(policy.arn())
                        .build();

                getIAMClient().deletePolicy(deletePolicyRequest);
                System.out.println("Policy deleted successfully.");
                break;
            }
        }

        // List the roles associated with the instance profile
        ListInstanceProfilesForRoleRequest listRolesRequest =
 ListInstanceProfilesForRoleRequest.builder()
                .roleName(roleName)
                .build();

        // Detach the roles from the instance profile
        ListInstanceProfilesForRoleResponse listRolesResponse =
 iamClient.listInstanceProfilesForRole(listRolesRequest);
        for (software.amazon.awssdk.services.iam.model.InstanceProfile profile :
 listRolesResponse.instanceProfiles()) {
            RemoveRoleFromInstanceProfileRequest removeRoleRequest =
 RemoveRoleFromInstanceProfileRequest.builder()
                    .instanceProfileName(InstanceProfile)
                    .roleName(roleName) // Remove the extra dot here
```

```
                    .build();

            getIAMClient().removeRoleFromInstanceProfile(removeRoleRequest);
            System.out.println("Role " + roleName + " removed from instance
 profile " + InstanceProfile);
        }

        // Delete the instance profile after removing all roles
        DeleteInstanceProfileRequest deleteInstanceProfileRequest =
 DeleteInstanceProfileRequest.builder()
                .instanceProfileName(InstanceProfile)
                .build();

        getIAMClient().deleteInstanceProfile(r ->
 r.instanceProfileName(InstanceProfile));
        System.out.println(InstanceProfile + " Deleted");
        System.out.println("All roles and policies are deleted.");
    }
}
```

Crea una classe che racchiuda le operazioni di Elastic Load Balancing.

```
public class LoadBalancer {
    public ElasticLoadBalancingV2Client elasticLoadBalancingV2Client;

    public ElasticLoadBalancingV2Client getLoadBalancerClient() {
        if (elasticLoadBalancingV2Client == null) {
            elasticLoadBalancingV2Client = ElasticLoadBalancingV2Client.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }

        return elasticLoadBalancingV2Client;
    }

    // Checks the health of the instances in the target group.
    public List<TargetHealthDescription> checkTargetHealth(String
 targetGroupName) {
        DescribeTargetGroupsRequest targetGroupsRequest =
 DescribeTargetGroupsRequest.builder()
                .names(targetGroupName)
                .build();
```

```
        DescribeTargetGroupsResponse tgResponse =
getLoadBalancerClient().describeTargetGroups(targetGroupsRequest);

        DescribeTargetHealthRequest healthRequest =
DescribeTargetHealthRequest.builder()

.targetGroupArn(tgResponse.targetGroups().get(0).targetGroupArn())
                .build();

        DescribeTargetHealthResponse healthResponse =
getLoadBalancerClient().describeTargetHealth(healthRequest);
        return healthResponse.targetHealthDescriptions();
    }

    // Gets the HTTP endpoint of the load balancer.
    public String getEndpoint(String lbName) {
        DescribeLoadBalancersResponse res = getLoadBalancerClient()
                .describeLoadBalancers(describe -> describe.names(lbName));
        return res.loadBalancers().get(0).dnsName();
    }

    // Deletes a load balancer.
    public void deleteLoadBalancer(String lbName) {
        try {
            // Use a waiter to delete the Load Balancer.
            DescribeLoadBalancersResponse res = getLoadBalancerClient()
                    .describeLoadBalancers(describe -> describe.names(lbName));
            ElasticLoadBalancingV2Waiter loadBalancerWaiter =
getLoadBalancerClient().waiter();
            DescribeLoadBalancersRequest request =
DescribeLoadBalancersRequest.builder()

.loadBalancerArns(res.loadBalancers().get(0).loadBalancerArn())
                    .build();

            getLoadBalancerClient().deleteLoadBalancer(
                    builder ->
builder.loadBalancerArn(res.loadBalancers().get(0).loadBalancerArn()));
            WaiterResponse<DescribeLoadBalancersResponse> waiterResponse =
loadBalancerWaiter
                    .waitUntilLoadBalancersDeleted(request);
            waiterResponse.matched().response().ifPresent(System.out::println);
```

```
        } catch (ElasticLoadBalancingV2Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
        System.out.println(lbName + " was deleted.");
    }

    // Deletes the target group.
    public void deleteTargetGroup(String targetGroupName) {
        try {
            DescribeTargetGroupsResponse res = getLoadBalancerClient()
                    .describeTargetGroups(describe ->
describe.names(targetGroupName));
            getLoadBalancerClient()
                    .deleteTargetGroup(builder ->
builder.targetGroupArn(res.targetGroups().get(0).targetGroupArn()));
        } catch (ElasticLoadBalancingV2Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
        System.out.println(targetGroupName + " was deleted.");
    }

    // Verify this computer can successfully send a GET request to the load
balancer
    // endpoint.
    public boolean verifyLoadBalancerEndpoint(String elbDnsName) throws
IOException, InterruptedException {
        boolean success = false;
        int retries = 3;
        CloseableHttpClient httpClient = HttpClients.createDefault();

        // Create an HTTP GET request to the ELB.
        HttpGet httpGet = new HttpGet("http://" + elbDnsName);
        try {
            while ((!success) && (retries > 0)) {
                // Execute the request and get the response.
                HttpResponse response = httpClient.execute(httpGet);
                int statusCode = response.getStatusLine().getStatusCode();
                System.out.println("HTTP Status Code: " + statusCode);
                if (statusCode == 200) {
                    success = true;
                } else {
                    retries--;
                    System.out.println("Got connection error from load balancer
endpoint, retrying...");
```

```
                    TimeUnit.SECONDS.sleep(15);
                }
            }

        } catch (org.apache.http.conn.HttpHostConnectException e) {
            System.out.println(e.getMessage());
        }

        System.out.println("Status.." + success);
        return success;
    }

    /*
     * Creates an Elastic Load Balancing target group. The target group specifies
     * how
     * the load balancer forward requests to instances in the group and how
instance
     * health is checked.
     */
    public String createTargetGroup(String protocol, int port, String vpcId,
String targetGroupName) {
        CreateTargetGroupRequest targetGroupRequest =
CreateTargetGroupRequest.builder()
                .healthCheckPath("/healthcheck")
                .healthCheckTimeoutSeconds(5)
                .port(port)
                .vpcId(vpcId)
                .name(targetGroupName)
                .protocol(protocol)
                .build();

        CreateTargetGroupResponse targetGroupResponse =
getLoadBalancerClient().createTargetGroup(targetGroupRequest);
        String targetGroupArn =
targetGroupResponse.targetGroups().get(0).targetGroupArn();
        String targetGroup =
targetGroupResponse.targetGroups().get(0).targetGroupName();
        System.out.println("The " + targetGroup + " was created with ARN" +
targetGroupArn);
        return targetGroupArn;
    }

    /*
     * Creates an Elastic Load Balancing load balancer that uses the specified
```

```
    * subnets
    * and forwards requests to the specified target group.
    */
   public String createLoadBalancer(List<Subnet> subnetIds, String
targetGroupARN, String lbName, int port,
           String protocol) {
       try {
           List<String> subnetIdStrings = subnetIds.stream()
                   .map(Subnet::subnetId)
                   .collect(Collectors.toList());

           CreateLoadBalancerRequest balancerRequest =
CreateLoadBalancerRequest.builder()
                   .subnets(subnetIdStrings)
                   .name(lbName)
                   .scheme("internet-facing")
                   .build();

           // Create and wait for the load balancer to become available.
           CreateLoadBalancerResponse lsResponse =
getLoadBalancerClient().createLoadBalancer(balancerRequest);
           String lbARN = lsResponse.loadBalancers().get(0).loadBalancerArn();

           ElasticLoadBalancingV2Waiter loadBalancerWaiter =
getLoadBalancerClient().waiter();
           DescribeLoadBalancersRequest request =
DescribeLoadBalancersRequest.builder()
                   .loadBalancerArns(lbARN)
                   .build();

           System.out.println("Waiting for Load Balancer " + lbName + " to
become available.");
           WaiterResponse<DescribeLoadBalancersResponse> waiterResponse =
loadBalancerWaiter
                   .waitUntilLoadBalancerAvailable(request);
           waiterResponse.matched().response().ifPresent(System.out::println);
           System.out.println("Load Balancer " + lbName + " is available.");

           // Get the DNS name (endpoint) of the load balancer.
           String lbDNSName = lsResponse.loadBalancers().get(0).dnsName();
           System.out.println("*** Load Balancer DNS Name: " + lbDNSName);

           // Create a listener for the load balance.
           Action action = Action.builder()
```

```
                    .targetGroupArn(targetGroupARN)
                    .type("forward")
                    .build();

            CreateListenerRequest listenerRequest =
 CreateListenerRequest.builder()

 .loadBalancerArn(lsResponse.loadBalancers().get(0).loadBalancerArn())
                    .defaultActions(action)
                    .port(port)
                    .protocol(protocol)
                    .build();

            getLoadBalancerClient().createListener(listenerRequest);
            System.out.println("Created listener to forward traffic from load
 balancer " + lbName + " to target group "
                    + targetGroupARN);

            // Return the load balancer DNS name.
            return lbDNSName;

        } catch (ElasticLoadBalancingV2Exception e) {
            e.printStackTrace();
        }
        return "";
    }
}
```

Crea una classe che utilizzi DynamoDB per simulare un servizio di raccomandazione.

```
public class Database {

    private static DynamoDbClient dynamoDbClient;

    public static DynamoDbClient getDynamoDbClient() {
        if (dynamoDbClient == null) {
            dynamoDbClient = DynamoDbClient.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }
        return dynamoDbClient;
    }
```

```
   // Checks to see if the Amazon DynamoDB table exists.
   private boolean doesTableExist(String tableName) {
       try {
           // Describe the table and catch any exceptions.
           DescribeTableRequest describeTableRequest =
DescribeTableRequest.builder()
                   .tableName(tableName)
                   .build();

           getDynamoDbClient().describeTable(describeTableRequest);
           System.out.println("Table '" + tableName + "' exists.");
           return true;

       } catch (ResourceNotFoundException e) {
           System.out.println("Table '" + tableName + "' does not exist.");
       } catch (DynamoDbException e) {
           System.err.println("Error checking table existence: " +
e.getMessage());
       }
       return false;
   }

   /*
    * Creates a DynamoDB table to use a recommendation service. The table has a
    * hash key named 'MediaType' that defines the type of media recommended,
such
    * as
    * Book or Movie, and a range key named 'ItemId' that, combined with the
    * MediaType,
    * forms a unique identifier for the recommended item.
    */
   public void createTable(String tableName, String fileName) throws IOException
{
       // First check to see if the table exists.
       boolean doesExist = doesTableExist(tableName);
       if (!doesExist) {
           DynamoDbWaiter dbWaiter = getDynamoDbClient().waiter();
           CreateTableRequest createTableRequest = CreateTableRequest.builder()
                   .tableName(tableName)
                   .attributeDefinitions(
                           AttributeDefinition.builder()
                                   .attributeName("MediaType")
                                   .attributeType(ScalarAttributeType.S)
```

```
                                .build(),
                        AttributeDefinition.builder()
                                .attributeName("ItemId")
                                .attributeType(ScalarAttributeType.N)
                                .build())
                .keySchema(
                        KeySchemaElement.builder()
                                .attributeName("MediaType")
                                .keyType(KeyType.HASH)
                                .build(),
                        KeySchemaElement.builder()
                                .attributeName("ItemId")
                                .keyType(KeyType.RANGE)
                                .build())
                .provisionedThroughput(
                        ProvisionedThroughput.builder()
                                .readCapacityUnits(5L)
                                .writeCapacityUnits(5L)
                                .build())
                .build();

        getDynamoDbClient().createTable(createTableRequest);
        System.out.println("Creating table " + tableName + "...");

        // Wait until the Amazon DynamoDB table is created.
        DescribeTableRequest tableRequest = DescribeTableRequest.builder()
                .tableName(tableName)
                .build();

        WaiterResponse<DescribeTableResponse> waiterResponse =
dbWaiter.waitUntilTableExists(tableRequest);
        waiterResponse.matched().response().ifPresent(System.out::println);
        System.out.println("Table " + tableName + " created.");

        // Add records to the table.
        populateTable(fileName, tableName);
    }
}

public void deleteTable(String tableName) {
    getDynamoDbClient().deleteTable(table -> table.tableName(tableName));
    System.out.println("Table " + tableName + " deleted.");
}
```

```
    // Populates the table with data located in a JSON file using the DynamoDB
    // enhanced client.
    public void populateTable(String fileName, String tableName) throws
 IOException {
        DynamoDbEnhancedClient enhancedClient = DynamoDbEnhancedClient.builder()
                .dynamoDbClient(getDynamoDbClient())
                .build();
        ObjectMapper objectMapper = new ObjectMapper();
        File jsonFile = new File(fileName);
        JsonNode rootNode = objectMapper.readTree(jsonFile);

        DynamoDbTable<Recommendation> mappedTable =
 enhancedClient.table(tableName,
                TableSchema.fromBean(Recommendation.class));
        for (JsonNode currentNode : rootNode) {
            String mediaType = currentNode.path("MediaType").path("S").asText();
            int itemId = currentNode.path("ItemId").path("N").asInt();
            String title = currentNode.path("Title").path("S").asText();
            String creator = currentNode.path("Creator").path("S").asText();

            // Create a Recommendation object and set its properties.
            Recommendation rec = new Recommendation();
            rec.setMediaType(mediaType);
            rec.setItemId(itemId);
            rec.setTitle(title);
            rec.setCreator(creator);

            // Put the item into the DynamoDB table.
            mappedTable.putItem(rec); // Add the Recommendation to the list.
        }
        System.out.println("Added all records to the " + tableName);
    }
}
```

Crea una classe che racchiuda le operazioni di Systems Manager.

```
public class ParameterHelper {

    String tableName = "doc-example-resilient-architecture-table";
    String dyntable = "doc-example-recommendation-service";
    String failureResponse = "doc-example-resilient-architecture-failure-
response";
```

```
    String healthCheck = "doc-example-resilient-architecture-health-check";

    public void reset() {
        put(dyntable, tableName);
        put(failureResponse, "none");
        put(healthCheck, "shallow");
    }

    public void put(String name, String value) {
        SsmClient ssmClient = SsmClient.builder()
                .region(Region.US_EAST_1)
                .build();

        PutParameterRequest parameterRequest = PutParameterRequest.builder()
                .name(name)
                .value(value)
                .overwrite(true)
                .type("String")
                .build();

        ssmClient.putParameter(parameterRequest);
        System.out.printf("Setting demo parameter %s to '%s'.", name, value);
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK for Java 2.x .
  - AttachLoadBalancerTargetGroups
  - CreateAutoScalingGroup
  - CreateInstanceProfile
  - CreateLaunchTemplate
  - CreateListener
  - CreateLoadBalancer
  - CreateTargetGroup
  - DeleteAutoScalingGroup
  - DeleteInstanceProfile
  - DeleteLaunchTemplate
  - DeleteLoadBalancer

- DeleteTargetGroup
- DescribeAutoScalingGroups
- DescribeAvailabilityZones
- DescribeIamInstanceProfileAssociations
- DescribeInstances
- DescribeLoadBalancers
- DescribeSubnets
- DescribeTargetGroups
- DescribeTargetHealth
- DescribeVpcs
- RebootInstances
- ReplaceIamInstanceProfileAssociation
- TerminateInstanceInAutoScalingGroup
- UpdateAutoScalingGroup

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Esegui lo scenario interattivo al prompt dei comandi.

```
#!/usr/bin/env node
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

import {
  Scenario,
  parseScenarioArgs,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";
```

```
/**
 * The workflow steps are split into three stages:
 *   - deploy
 *   - demo
 *   - destroy
 *
 * Each of these stages has a corresponding file prefixed with steps-*.
 */
import { deploySteps } from "./steps-deploy.js";
import { demoSteps } from "./steps-demo.js";
import { destroySteps } from "./steps-destroy.js";

/**
 * The context is passed to every scenario. Scenario steps
 * will modify the context.
 */
const context = {};

/**
 * Three Scenarios are created for the workflow. A Scenario is an orchestration
 class
 * that simplifies running a series of steps.
 */
export const scenarios = {
  // Deploys all resources necessary for the workflow.
  deploy: new Scenario("Resilient Workflow - Deploy", deploySteps, context),
  // Demonstrates how a fragile web service can be made more resilient.
  demo: new Scenario("Resilient Workflow - Demo", demoSteps, context),
  // Destroys the resources created for the workflow.
  destroy: new Scenario("Resilient Workflow - Destroy", destroySteps, context),
};

// Call function if run directly
import { fileURLToPath } from "node:url";

if (process.argv[1] === fileURLToPath(import.meta.url)) {
  parseScenarioArgs(scenarios, {
    name: "Resilient Workflow",
    synopsis:
      "node index.js --scenario <deploy | demo | destroy> [-h|--help] [-y|--yes]
 [-v|--verbose]",
    description: "Deploy and interact with scalable EC2 instances.",
  });
}
```

Crea passaggi per distribuire tutte le risorse.

```
import { join } from "node:path";
import { readFileSync, writeFileSync } from "node:fs";
import axios from "axios";

import {
  BatchWriteItemCommand,
  CreateTableCommand,
  DynamoDBClient,
  waitUntilTableExists,
} from "@aws-sdk/client-dynamodb";
import {
  EC2Client,
  CreateKeyPairCommand,
  CreateLaunchTemplateCommand,
  DescribeAvailabilityZonesCommand,
  DescribeVpcsCommand,
  DescribeSubnetsCommand,
  DescribeSecurityGroupsCommand,
  AuthorizeSecurityGroupIngressCommand,
} from "@aws-sdk/client-ec2";
import {
  IAMClient,
  CreatePolicyCommand,
  CreateRoleCommand,
  CreateInstanceProfileCommand,
  AddRoleToInstanceProfileCommand,
  AttachRolePolicyCommand,
  waitUntilInstanceProfileExists,
} from "@aws-sdk/client-iam";
import { SSMClient, GetParameterCommand } from "@aws-sdk/client-ssm";
import {
  CreateAutoScalingGroupCommand,
  AutoScalingClient,
  AttachLoadBalancerTargetGroupsCommand,
} from "@aws-sdk/client-auto-scaling";
import {
  CreateListenerCommand,
  CreateLoadBalancerCommand,
  CreateTargetGroupCommand,
```

```
  ElasticLoadBalancingV2Client,
  waitUntilLoadBalancerAvailable,
} from "@aws-sdk/client-elastic-load-balancing-v2";

import {
  ScenarioOutput,
  ScenarioInput,
  ScenarioAction,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";
import { saveState } from "@aws-doc-sdk-examples/lib/scenario/steps-common.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

import { MESSAGES, NAMES, RESOURCES_PATH, ROOT } from "./constants.js";
import { initParamsSteps } from "./steps-reset-params.js";

/**
 * @type {import('@aws-doc-sdk-examples/lib/scenario.js').Step[]}
 */
export const deploySteps = [
  new ScenarioOutput("introduction", MESSAGES.introduction, { header: true }),
  new ScenarioInput("confirmDeployment", MESSAGES.confirmDeployment, {
    type: "confirm",
  }),
  new ScenarioAction(
    "handleConfirmDeployment",
    (c) => c.confirmDeployment === false && process.exit(),
  ),
  new ScenarioOutput(
    "creatingTable",
    MESSAGES.creatingTable.replace("${TABLE_NAME}", NAMES.tableName),
  ),
  new ScenarioAction("createTable", async () => {
    const client = new DynamoDBClient({});
    await client.send(
      new CreateTableCommand({
        TableName: NAMES.tableName,
        ProvisionedThroughput: {
          ReadCapacityUnits: 5,
          WriteCapacityUnits: 5,
        },
        AttributeDefinitions: [
          {
            AttributeName: "MediaType",
            AttributeType: "S",
```

```
          },
          {
            AttributeName: "ItemId",
            AttributeType: "N",
          },
        ],
        KeySchema: [
          {
            AttributeName: "MediaType",
            KeyType: "HASH",
          },
          {
            AttributeName: "ItemId",
            KeyType: "RANGE",
          },
        ],
      }),
    );
    await waitUntilTableExists({ client }, { TableName: NAMES.tableName });
  }),
  new ScenarioOutput(
    "createdTable",
    MESSAGES.createdTable.replace("${TABLE_NAME}", NAMES.tableName),
  ),
  new ScenarioOutput(
    "populatingTable",
    MESSAGES.populatingTable.replace("${TABLE_NAME}", NAMES.tableName),
  ),
  new ScenarioAction("populateTable", () => {
    const client = new DynamoDBClient({});
    /**
     * @type {{ default: import("@aws-sdk/client-dynamodb").PutRequest['Item']
[] }}
     */
    const recommendations = JSON.parse(
      readFileSync(join(RESOURCES_PATH, "recommendations.json")),
    );

    return client.send(
      new BatchWriteItemCommand({
        RequestItems: {
          [NAMES.tableName]: recommendations.map((item) => ({
            PutRequest: { Item: item },
          })),
```

```
      },
    }),
  );
}),
new ScenarioOutput(
  "populatedTable",
  MESSAGES.populatedTable.replace("${TABLE_NAME}", NAMES.tableName),
),
new ScenarioOutput(
  "creatingKeyPair",
  MESSAGES.creatingKeyPair.replace("${KEY_PAIR_NAME}", NAMES.keyPairName),
),
new ScenarioAction("createKeyPair", async () => {
  const client = new EC2Client({});
  const { KeyMaterial } = await client.send(
    new CreateKeyPairCommand({
      KeyName: NAMES.keyPairName,
    }),
  );

  writeFileSync(`${NAMES.keyPairName}.pem`, KeyMaterial, { mode: 0o600 });
}),
new ScenarioOutput(
  "createdKeyPair",
  MESSAGES.createdKeyPair.replace("${KEY_PAIR_NAME}", NAMES.keyPairName),
),
new ScenarioOutput(
  "creatingInstancePolicy",
  MESSAGES.creatingInstancePolicy.replace(
    "${INSTANCE_POLICY_NAME}",
    NAMES.instancePolicyName,
  ),
),
new ScenarioAction("createInstancePolicy", async (state) => {
  const client = new IAMClient({});
  const {
    Policy: { Arn },
  } = await client.send(
    new CreatePolicyCommand({
      PolicyName: NAMES.instancePolicyName,
      PolicyDocument: readFileSync(
        join(RESOURCES_PATH, "instance_policy.json"),
      ),
    }),
```

```
    );
    state.instancePolicyArn = Arn;
  }),
  new ScenarioOutput("createdInstancePolicy", (state) =>
    MESSAGES.createdInstancePolicy
      .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
      .replace("${INSTANCE_POLICY_ARN}", state.instancePolicyArn),
  ),
  new ScenarioOutput(
    "creatingInstanceRole",
    MESSAGES.creatingInstanceRole.replace(
      "${INSTANCE_ROLE_NAME}",
      NAMES.instanceRoleName,
    ),
  ),
  new ScenarioAction("createInstanceRole", () => {
    const client = new IAMClient({});
    return client.send(
      new CreateRoleCommand({
        RoleName: NAMES.instanceRoleName,
        AssumeRolePolicyDocument: readFileSync(
          join(ROOT, "assume-role-policy.json"),
        ),
      }),
    );
  }),
  new ScenarioOutput(
    "createdInstanceRole",
    MESSAGES.createdInstanceRole.replace(
      "${INSTANCE_ROLE_NAME}",
      NAMES.instanceRoleName,
    ),
  ),
  new ScenarioOutput(
    "attachingPolicyToRole",
    MESSAGES.attachingPolicyToRole
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName)
      .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName),
  ),
  new ScenarioAction("attachPolicyToRole", async (state) => {
    const client = new IAMClient({});
    await client.send(
      new AttachRolePolicyCommand({
        RoleName: NAMES.instanceRoleName,
```

```
      PolicyArn: state.instancePolicyArn,
    }),
  );
}),
new ScenarioOutput(
  "attachedPolicyToRole",
  MESSAGES.attachedPolicyToRole
    .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
    .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName),
),
new ScenarioOutput(
  "creatingInstanceProfile",
  MESSAGES.creatingInstanceProfile.replace(
    "${INSTANCE_PROFILE_NAME}",
    NAMES.instanceProfileName,
  ),
),
new ScenarioAction("createInstanceProfile", async (state) => {
  const client = new IAMClient({});
  const {
    InstanceProfile: { Arn },
  } = await client.send(
    new CreateInstanceProfileCommand({
      InstanceProfileName: NAMES.instanceProfileName,
    }),
  );
  state.instanceProfileArn = Arn;

  await waitUntilInstanceProfileExists(
    { client },
    { InstanceProfileName: NAMES.instanceProfileName },
  );
}),
new ScenarioOutput("createdInstanceProfile", (state) =>
  MESSAGES.createdInstanceProfile
    .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
    .replace("${INSTANCE_PROFILE_ARN}", state.instanceProfileArn),
),
new ScenarioOutput(
  "addingRoleToInstanceProfile",
  MESSAGES.addingRoleToInstanceProfile
    .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
    .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName),
),
```

```
  new ScenarioAction("addRoleToInstanceProfile", () => {
    const client = new IAMClient({});
    return client.send(
      new AddRoleToInstanceProfileCommand({
        RoleName: NAMES.instanceRoleName,
        InstanceProfileName: NAMES.instanceProfileName,
      }),
    );
  }),
  new ScenarioOutput(
    "addedRoleToInstanceProfile",
    MESSAGES.addedRoleToInstanceProfile
      .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName),
  ),
  ...initParamsSteps,
  new ScenarioOutput("creatingLaunchTemplate", MESSAGES.creatingLaunchTemplate),
  new ScenarioAction("createLaunchTemplate", async () => {
    const ssmClient = new SSMClient({});
    const { Parameter } = await ssmClient.send(
      new GetParameterCommand({
        Name: "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",
      }),
    );
    const ec2Client = new EC2Client({});
    await ec2Client.send(
      new CreateLaunchTemplateCommand({
        LaunchTemplateName: NAMES.launchTemplateName,
        LaunchTemplateData: {
          InstanceType: "t3.micro",
          ImageId: Parameter.Value,
          IamInstanceProfile: { Name: NAMES.instanceProfileName },
          UserData: readFileSync(
            join(RESOURCES_PATH, "server_startup_script.sh"),
          ).toString("base64"),
          KeyName: NAMES.keyPairName,
        },
      }),
    );
  }),
  new ScenarioOutput(
    "createdLaunchTemplate",
    MESSAGES.createdLaunchTemplate.replace(
      "${LAUNCH_TEMPLATE_NAME}",
```

```
      NAMES.launchTemplateName,
    ),
  ),
  new ScenarioOutput(
    "creatingAutoScalingGroup",
    MESSAGES.creatingAutoScalingGroup.replace(
      "${AUTO_SCALING_GROUP_NAME}",
      NAMES.autoScalingGroupName,
    ),
  ),
  new ScenarioAction("createAutoScalingGroup", async (state) => {
    const ec2Client = new EC2Client({});
    const { AvailabilityZones } = await ec2Client.send(
      new DescribeAvailabilityZonesCommand({}),
    );
    state.availabilityZoneNames = AvailabilityZones.map((az) => az.ZoneName);
    const autoScalingClient = new AutoScalingClient({});
    await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
      autoScalingClient.send(
        new CreateAutoScalingGroupCommand({
          AvailabilityZones: state.availabilityZoneNames,
          AutoScalingGroupName: NAMES.autoScalingGroupName,
          LaunchTemplate: {
            LaunchTemplateName: NAMES.launchTemplateName,
            Version: "$Default",
          },
          MinSize: 3,
          MaxSize: 3,
        }),
      ),
    );
  }),
  new ScenarioOutput(
    "createdAutoScalingGroup",
    /**
     * @param {{ availabilityZoneNames: string[] }} state
     */
    (state) =>
      MESSAGES.createdAutoScalingGroup
        .replace("${AUTO_SCALING_GROUP_NAME}", NAMES.autoScalingGroupName)
        .replace(
          "${AVAILABILITY_ZONE_NAMES}",
          state.availabilityZoneNames.join(", "),
        ),
```

```
  ),
  new ScenarioInput("confirmContinue", MESSAGES.confirmContinue, {
    type: "confirm",
  }),
  new ScenarioOutput("loadBalancer", MESSAGES.loadBalancer),
  new ScenarioOutput("gettingVpc", MESSAGES.gettingVpc),
  new ScenarioAction("getVpc", async (state) => {
    const client = new EC2Client({});
    const { Vpcs } = await client.send(
      new DescribeVpcsCommand({
        Filters: [{ Name: "is-default", Values: ["true"] }],
      }),
    );
    state.defaultVpc = Vpcs[0].VpcId;
  }),
  new ScenarioOutput("gotVpc", (state) =>
    MESSAGES.gotVpc.replace("${VPC_ID}", state.defaultVpc),
  ),
  new ScenarioOutput("gettingSubnets", MESSAGES.gettingSubnets),
  new ScenarioAction("getSubnets", async (state) => {
    const client = new EC2Client({});
    const { Subnets } = await client.send(
      new DescribeSubnetsCommand({
        Filters: [
          { Name: "vpc-id", Values: [state.defaultVpc] },
          { Name: "availability-zone", Values: state.availabilityZoneNames },
          { Name: "default-for-az", Values: ["true"] },
        ],
      }),
    );
    state.subnets = Subnets.map((subnet) => subnet.SubnetId);
  }),
  new ScenarioOutput(
    "gotSubnets",
    /**
     * @param {{ subnets: string[] }} state
     */
    (state) =>
      MESSAGES.gotSubnets.replace("${SUBNETS}", state.subnets.join(", ")),
  ),
  new ScenarioOutput(
    "creatingLoadBalancerTargetGroup",
    MESSAGES.creatingLoadBalancerTargetGroup.replace(
      "${TARGET_GROUP_NAME}",
```

```
      NAMES.loadBalancerTargetGroupName,
    ),
  ),
  new ScenarioAction("createLoadBalancerTargetGroup", async (state) => {
    const client = new ElasticLoadBalancingV2Client({});
    const { TargetGroups } = await client.send(
      new CreateTargetGroupCommand({
        Name: NAMES.loadBalancerTargetGroupName,
        Protocol: "HTTP",
        Port: 80,
        HealthCheckPath: "/healthcheck",
        HealthCheckIntervalSeconds: 10,
        HealthCheckTimeoutSeconds: 5,
        HealthyThresholdCount: 2,
        UnhealthyThresholdCount: 2,
        VpcId: state.defaultVpc,
      }),
    );
    const targetGroup = TargetGroups[0];
    state.targetGroupArn = targetGroup.TargetGroupArn;
    state.targetGroupProtocol = targetGroup.Protocol;
    state.targetGroupPort = targetGroup.Port;
  }),
  new ScenarioOutput(
    "createdLoadBalancerTargetGroup",
    MESSAGES.createdLoadBalancerTargetGroup.replace(
      "${TARGET_GROUP_NAME}",
      NAMES.loadBalancerTargetGroupName,
    ),
  ),
  new ScenarioOutput(
    "creatingLoadBalancer",
    MESSAGES.creatingLoadBalancer.replace("${LB_NAME}", NAMES.loadBalancerName),
  ),
  new ScenarioAction("createLoadBalancer", async (state) => {
    const client = new ElasticLoadBalancingV2Client({});
    const { LoadBalancers } = await client.send(
      new CreateLoadBalancerCommand({
        Name: NAMES.loadBalancerName,
        Subnets: state.subnets,
      }),
    );
    state.loadBalancerDns = LoadBalancers[0].DNSName;
    state.loadBalancerArn = LoadBalancers[0].LoadBalancerArn;
```

```
    await waitUntilLoadBalancerAvailable(
      { client },
      { Names: [NAMES.loadBalancerName] },
    );
  }),
  new ScenarioOutput("createdLoadBalancer", (state) =>
    MESSAGES.createdLoadBalancer
      .replace("${LB_NAME}", NAMES.loadBalancerName)
      .replace("${DNS_NAME}", state.loadBalancerDns),
  ),
  new ScenarioOutput(
    "creatingListener",
    MESSAGES.creatingLoadBalancerListener
      .replace("${LB_NAME}", NAMES.loadBalancerName)
      .replace("${TARGET_GROUP_NAME}", NAMES.loadBalancerTargetGroupName),
  ),
  new ScenarioAction("createListener", async (state) => {
    const client = new ElasticLoadBalancingV2Client({});
    const { Listeners } = await client.send(
      new CreateListenerCommand({
        LoadBalancerArn: state.loadBalancerArn,
        Protocol: state.targetGroupProtocol,
        Port: state.targetGroupPort,
        DefaultActions: [
          { Type: "forward", TargetGroupArn: state.targetGroupArn },
        ],
      }),
    );
    const listener = Listeners[0];
    state.loadBalancerListenerArn = listener.ListenerArn;
  }),
  new ScenarioOutput("createdListener", (state) =>
    MESSAGES.createdLoadBalancerListener.replace(
      "${LB_LISTENER_ARN}",
      state.loadBalancerListenerArn,
    ),
  ),
  new ScenarioOutput(
    "attachingLoadBalancerTargetGroup",
    MESSAGES.attachingLoadBalancerTargetGroup
      .replace("${TARGET_GROUP_NAME}", NAMES.loadBalancerTargetGroupName)
      .replace("${AUTO_SCALING_GROUP_NAME}", NAMES.autoScalingGroupName),
  ),
  new ScenarioAction("attachLoadBalancerTargetGroup", async (state) => {
```

```
    const client = new AutoScalingClient({});
    await client.send(
      new AttachLoadBalancerTargetGroupsCommand({
        AutoScalingGroupName: NAMES.autoScalingGroupName,
        TargetGroupARNs: [state.targetGroupArn],
      }),
    );
  }),
  new ScenarioOutput(
    "attachedLoadBalancerTargetGroup",
    MESSAGES.attachedLoadBalancerTargetGroup,
  ),
  new ScenarioOutput("verifyingInboundPort", MESSAGES.verifyingInboundPort),
  new ScenarioAction(
    "verifyInboundPort",
    /**
     *
     * @param {{ defaultSecurityGroup: import('@aws-sdk/client-
ec2').SecurityGroup}} state
     */
    async (state) => {
      const client = new EC2Client({});
      const { SecurityGroups } = await client.send(
        new DescribeSecurityGroupsCommand({
          Filters: [{ Name: "group-name", Values: ["default"] }],
        }),
      );
      if (!SecurityGroups) {
        state.verifyInboundPortError = new Error(MESSAGES.noSecurityGroups);
      }
      state.defaultSecurityGroup = SecurityGroups[0];

      /**
       * @type {string}
       */
      const ipResponse = (await axios.get("http://checkip.amazonaws.com")).data;
      state.myIp = ipResponse.trim();
      const myIpRules = state.defaultSecurityGroup.IpPermissions.filter(
        ({ IpRanges }) =>
          IpRanges.some(
            ({ CidrIp }) =>
              CidrIp.startsWith(state.myIp) || CidrIp === "0.0.0.0/0",
          ),
      )
```

```
        .filter(({ IpProtocol }) => IpProtocol === "tcp")
        .filter(({ FromPort }) => FromPort === 80);

      state.myIpRules = myIpRules;
    },
  ),
  new ScenarioOutput(
    "verifiedInboundPort",
    /**
     * @param {{ myIpRules: any[] }} state
     */
    (state) => {
      if (state.myIpRules.length > 0) {
        return MESSAGES.foundIpRules.replace(
          "${IP_RULES}",
          JSON.stringify(state.myIpRules, null, 2),
        );
      }
      return MESSAGES.noIpRules;
    },
  ),
  new ScenarioInput(
    "shouldAddInboundRule",
    /**
     * @param {{ myIpRules: any[] }} state
     */
    (state) => {
      if (state.myIpRules.length > 0) {
        return false;
      }
      return MESSAGES.noIpRules;
    },
    { type: "confirm" },
  ),
  new ScenarioAction(
    "addInboundRule",
    /**
     * @param {{ defaultSecurityGroup: import('@aws-sdk/client-
ec2').SecurityGroup }} state
     */
    async (state) => {
      if (!state.shouldAddInboundRule) {
        return;
      }
```

```
      const client = new EC2Client({});
      await client.send(
        new AuthorizeSecurityGroupIngressCommand({
          GroupId: state.defaultSecurityGroup.GroupId,
          CidrIp: `${state.myIp}/32`,
          FromPort: 80,
          ToPort: 80,
          IpProtocol: "tcp",
        }),
      );
    },
  ),
  new ScenarioOutput("addedInboundRule", (state) => {
    if (state.shouldAddInboundRule) {
      return MESSAGES.addedInboundRule.replace("${IP_ADDRESS}", state.myIp);
    }
    return false;
  }),
  new ScenarioOutput("verifyingEndpoint", (state) =>
    MESSAGES.verifyingEndpoint.replace("${DNS_NAME}", state.loadBalancerDns),
  ),
  new ScenarioAction("verifyEndpoint", async (state) => {
    try {
      const response = await retry({ intervalInMs: 2000, maxRetries: 30 }, () =>
        axios.get(`http://${state.loadBalancerDns}`),
      );
      state.endpointResponse = JSON.stringify(response.data, null, 2);
    } catch (e) {
      state.verifyEndpointError = e;
    }
  }),
  new ScenarioOutput("verifiedEndpoint", (state) => {
    if (state.verifyEndpointError) {
      console.error(state.verifyEndpointError);
    } else {
      return MESSAGES.verifiedEndpoint.replace(
        "${ENDPOINT_RESPONSE}",
        state.endpointResponse,
      );
    }
  }),
  saveState,
];
```

Crea i passaggi per eseguire la demo.

```
import { readFileSync } from "node:fs";
import { join } from "node:path";

import axios from "axios";

import {
  DescribeTargetGroupsCommand,
  DescribeTargetHealthCommand,
  ElasticLoadBalancingV2Client,
} from "@aws-sdk/client-elastic-load-balancing-v2";
import {
  DescribeInstanceInformationCommand,
  PutParameterCommand,
  SSMClient,
  SendCommandCommand,
} from "@aws-sdk/client-ssm";
import {
  IAMClient,
  CreatePolicyCommand,
  CreateRoleCommand,
  AttachRolePolicyCommand,
  CreateInstanceProfileCommand,
  AddRoleToInstanceProfileCommand,
  waitUntilInstanceProfileExists,
} from "@aws-sdk/client-iam";
import {
  AutoScalingClient,
  DescribeAutoScalingGroupsCommand,
  TerminateInstanceInAutoScalingGroupCommand,
} from "@aws-sdk/client-auto-scaling";
import {
  DescribeIamInstanceProfileAssociationsCommand,
  EC2Client,
  RebootInstancesCommand,
  ReplaceIamInstanceProfileAssociationCommand,
} from "@aws-sdk/client-ec2";

import {
  ScenarioAction,
```

```
  ScenarioInput,
  ScenarioOutput,
} from "@aws-doc-sdk-examples/lib/scenario/scenario.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

import { MESSAGES, NAMES, RESOURCES_PATH } from "./constants.js";
import { findLoadBalancer } from "./shared.js";

const getRecommendation = new ScenarioAction(
  "getRecommendation",
  async (state) => {
    const loadBalancer = await findLoadBalancer(NAMES.loadBalancerName);
    if (loadBalancer) {
      state.loadBalancerDnsName = loadBalancer.DNSName;
      try {
        state.recommendation = (
          await axios.get(`http://${state.loadBalancerDnsName}`)
        ).data;
      } catch (e) {
        state.recommendation = e instanceof Error ? e.message : e;
      }
    } else {
      throw new Error(MESSAGES.demoFindLoadBalancerError);
    }
  },
);

const getRecommendationResult = new ScenarioOutput(
  "getRecommendationResult",
  (state) =>
    `Recommendation:\n${JSON.stringify(state.recommendation, null, 2)}`,
  { preformatted: true },
);

const getHealthCheck = new ScenarioAction("getHealthCheck", async (state) => {
  const client = new ElasticLoadBalancingV2Client({});
  const { TargetGroups } = await client.send(
    new DescribeTargetGroupsCommand({
      Names: [NAMES.loadBalancerTargetGroupName],
    }),
  );

  const { TargetHealthDescriptions } = await client.send(
    new DescribeTargetHealthCommand({
```

```
      TargetGroupArn: TargetGroups[0].TargetGroupArn,
    }),
  );
  state.targetHealthDescriptions = TargetHealthDescriptions;
});

const getHealthCheckResult = new ScenarioOutput(
  "getHealthCheckResult",
  /**
   * @param {{ targetHealthDescriptions: import('@aws-sdk/client-elastic-load-
balancing-v2').TargetHealthDescription[]}} state
   */
  (state) => {
    const status = state.targetHealthDescriptions
      .map((th) => `${th.Target.Id}: ${th.TargetHealth.State}`)
      .join("\n");
    return `Health check:\n${status}`;
  },
  { preformatted: true },
);

const loadBalancerLoop = new ScenarioAction(
  "loadBalancerLoop",
  getRecommendation.action,
  {
    whileConfig: {
      whileFn: ({ loadBalancerCheck }) => loadBalancerCheck,
      input: new ScenarioInput(
        "loadBalancerCheck",
        MESSAGES.demoLoadBalancerCheck,
        {
          type: "confirm",
        },
      ),
      output: getRecommendationResult,
    },
  },
);

const healthCheckLoop = new ScenarioAction(
  "healthCheckLoop",
  getHealthCheck.action,
  {
    whileConfig: {
```

```
      whileFn: ({ healthCheck }) => healthCheck,
      input: new ScenarioInput("healthCheck", MESSAGES.demoHealthCheck, {
        type: "confirm",
      }),
      output: getHealthCheckResult,
    },
  },
);

const statusSteps = [
  getRecommendation,
  getRecommendationResult,
  getHealthCheck,
  getHealthCheckResult,
];

/**
 * @type {import('@aws-doc-sdk-examples/lib/scenario.js').Step[]}
 */
export const demoSteps = [
  new ScenarioOutput("header", MESSAGES.demoHeader, { header: true }),
  new ScenarioOutput("sanityCheck", MESSAGES.demoSanityCheck),
  ...statusSteps,
  new ScenarioInput(
    "brokenDependencyConfirmation",
    MESSAGES.demoBrokenDependencyConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("brokenDependency", async (state) => {
    if (!state.brokenDependencyConfirmation) {
      process.exit();
    } else {
      const client = new SSMClient({});
      state.badTableName = `fake-table-${Date.now()}`;
      await client.send(
        new PutParameterCommand({
          Name: NAMES.ssmTableNameKey,
          Value: state.badTableName,
          Overwrite: true,
          Type: "String",
        }),
      );
    }
  }),
```

```
  new ScenarioOutput("testBrokenDependency", (state) =>
    MESSAGES.demoTestBrokenDependency.replace(
      "${TABLE_NAME}",
      state.badTableName,
    ),
  ),
  ...statusSteps,
  new ScenarioInput(
    "staticResponseConfirmation",
    MESSAGES.demoStaticResponseConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("staticResponse", async (state) => {
    if (!state.staticResponseConfirmation) {
      process.exit();
    } else {
      const client = new SSMClient({});
      await client.send(
        new PutParameterCommand({
          Name: NAMES.ssmFailureResponseKey,
          Value: "static",
          Overwrite: true,
          Type: "String",
        }),
      );
    }
  }),
  new ScenarioOutput("testStaticResponse", MESSAGES.demoTestStaticResponse),
  ...statusSteps,
  new ScenarioInput(
    "badCredentialsConfirmation",
    MESSAGES.demoBadCredentialsConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("badCredentialsExit", (state) => {
    if (!state.badCredentialsConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("fixDynamoDBName", async () => {
    const client = new SSMClient({});
    await client.send(
      new PutParameterCommand({
        Name: NAMES.ssmTableNameKey,
```

```
        Value: NAMES.tableName,
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioAction(
    "badCredentials",
    /**
     * @param {{ targetInstance: import('@aws-sdk/client-auto-
scaling').Instance }} state
     */
    async (state) => {
      await createSsmOnlyInstanceProfile();
      const autoScalingClient = new AutoScalingClient({});
      const { AutoScalingGroups } = await autoScalingClient.send(
        new DescribeAutoScalingGroupsCommand({
          AutoScalingGroupNames: [NAMES.autoScalingGroupName],
        }),
      );
      state.targetInstance = AutoScalingGroups[0].Instances[0];
      const ec2Client = new EC2Client({});
      const { IamInstanceProfileAssociations } = await ec2Client.send(
        new DescribeIamInstanceProfileAssociationsCommand({
          Filters: [
            { Name: "instance-id", Values: [state.targetInstance.InstanceId] },
          ],
        }),
      );
      state.instanceProfileAssociationId =
        IamInstanceProfileAssociations[0].AssociationId;
      await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
        ec2Client.send(
          new ReplaceIamInstanceProfileAssociationCommand({
            AssociationId: state.instanceProfileAssociationId,
            IamInstanceProfile: { Name: NAMES.ssmOnlyInstanceProfileName },
          }),
        ),
      );

      await ec2Client.send(
        new RebootInstancesCommand({
          InstanceIds: [state.targetInstance.InstanceId],
        }),
```

```
      );

      const ssmClient = new SSMClient({});
      await retry({ intervalInMs: 20000, maxRetries: 15 }, async () => {
        const { InstanceInformationList } = await ssmClient.send(
          new DescribeInstanceInformationCommand({}),
        );

        const instance = InstanceInformationList.find(
          (info) => info.InstanceId === state.targetInstance.InstanceId,
        );

        if (!instance) {
          throw new Error("Instance not found.");
        }
      });

      await ssmClient.send(
        new SendCommandCommand({
          InstanceIds: [state.targetInstance.InstanceId],
          DocumentName: "AWS-RunShellScript",
          Parameters: { commands: ["cd / && sudo python3 server.py 80"] },
        }),
      );
    },
  ),
  new ScenarioOutput(
    "testBadCredentials",
    /**
     * @param {{ targetInstance: import('@aws-sdk/client-
ssm').InstanceInformation}} state
     */
    (state) =>
      MESSAGES.demoTestBadCredentials.replace(
        "${INSTANCE_ID}",
        state.targetInstance.InstanceId,
      ),
  ),
  loadBalancerLoop,
  new ScenarioInput(
    "deepHealthCheckConfirmation",
    MESSAGES.demoDeepHealthCheckConfirmation,
    { type: "confirm" },
  ),
```

```
  new ScenarioAction("deepHealthCheckExit", (state) => {
    if (!state.deepHealthCheckConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("deepHealthCheck", async () => {
    const client = new SSMClient({});
    await client.send(
      new PutParameterCommand({
        Name: NAMES.ssmHealthCheckKey,
        Value: "deep",
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioOutput("testDeepHealthCheck", MESSAGES.demoTestDeepHealthCheck),
  healthCheckLoop,
  loadBalancerLoop,
  new ScenarioInput(
    "killInstanceConfirmation",
    /**
     * @param {{ targetInstance: import('@aws-sdk/client-
ssm').InstanceInformation }} state
     */
    (state) =>
      MESSAGES.demoKillInstanceConfirmation.replace(
        "${INSTANCE_ID}",
        state.targetInstance.InstanceId,
      ),
    { type: "confirm" },
  ),
  new ScenarioAction("killInstanceExit", (state) => {
    if (!state.killInstanceConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction(
    "killInstance",
    /**
     * @param {{ targetInstance: import('@aws-sdk/client-
ssm').InstanceInformation }} state
     */
    async (state) => {
```

```
      const client = new AutoScalingClient({});
      await client.send(
        new TerminateInstanceInAutoScalingGroupCommand({
          InstanceId: state.targetInstance.InstanceId,
          ShouldDecrementDesiredCapacity: false,
        }),
      );
    },
  ),
  new ScenarioOutput("testKillInstance", MESSAGES.demoTestKillInstance),
  healthCheckLoop,
  loadBalancerLoop,
  new ScenarioInput("failOpenConfirmation", MESSAGES.demoFailOpenConfirmation, {
    type: "confirm",
  }),
  new ScenarioAction("failOpenExit", (state) => {
    if (!state.failOpenConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("failOpen", () => {
    const client = new SSMClient({});
    return client.send(
      new PutParameterCommand({
        Name: NAMES.ssmTableNameKey,
        Value: `fake-table-${Date.now()}`,
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioOutput("testFailOpen", MESSAGES.demoFailOpenTest),
  healthCheckLoop,
  loadBalancerLoop,
  new ScenarioInput(
    "resetTableConfirmation",
    MESSAGES.demoResetTableConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("resetTableExit", (state) => {
    if (!state.resetTableConfirmation) {
      process.exit();
    }
  }),
```

```
  new ScenarioAction("resetTable", async () => {
    const client = new SSMClient({});
    await client.send(
      new PutParameterCommand({
        Name: NAMES.ssmTableNameKey,
        Value: NAMES.tableName,
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioOutput("testResetTable", MESSAGES.demoTestResetTable),
  healthCheckLoop,
  loadBalancerLoop,
];

async function createSsmOnlyInstanceProfile() {
  const iamClient = new IAMClient({});
  const { Policy } = await iamClient.send(
    new CreatePolicyCommand({
      PolicyName: NAMES.ssmOnlyPolicyName,
      PolicyDocument: readFileSync(
        join(RESOURCES_PATH, "ssm_only_policy.json"),
      ),
    }),
  );
  await iamClient.send(
    new CreateRoleCommand({
      RoleName: NAMES.ssmOnlyRoleName,
      AssumeRolePolicyDocument: JSON.stringify({
        Version: "2012-10-17",
        Statement: [
          {
            Effect: "Allow",
            Principal: { Service: "ec2.amazonaws.com" },
            Action: "sts:AssumeRole",
          },
        ],
      }),
    }),
  );
  await iamClient.send(
    new AttachRolePolicyCommand({
      RoleName: NAMES.ssmOnlyRoleName,
```

```
      PolicyArn: Policy.Arn,
    }),
  );
  await iamClient.send(
    new AttachRolePolicyCommand({
      RoleName: NAMES.ssmOnlyRoleName,
      PolicyArn: "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
    }),
  );
  const { InstanceProfile } = await iamClient.send(
    new CreateInstanceProfileCommand({
      InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
    }),
  );
  await waitUntilInstanceProfileExists(
    { client: iamClient },
    { InstanceProfileName: NAMES.ssmOnlyInstanceProfileName },
  );
  await iamClient.send(
    new AddRoleToInstanceProfileCommand({
      InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
      RoleName: NAMES.ssmOnlyRoleName,
    }),
  );

  return InstanceProfile;
}
```

Crea i passaggi per distruggere tutte le risorse.

```
import { unlinkSync } from "node:fs";

import { DynamoDBClient, DeleteTableCommand } from "@aws-sdk/client-dynamodb";
import {
  EC2Client,
  DeleteKeyPairCommand,
  DeleteLaunchTemplateCommand,
  RevokeSecurityGroupIngressCommand,
} from "@aws-sdk/client-ec2";
import {
  IAMClient,
  DeleteInstanceProfileCommand,
```

```
  RemoveRoleFromInstanceProfileCommand,
  DeletePolicyCommand,
  DeleteRoleCommand,
  DetachRolePolicyCommand,
  paginateListPolicies,
} from "@aws-sdk/client-iam";
import {
  AutoScalingClient,
  DeleteAutoScalingGroupCommand,
  TerminateInstanceInAutoScalingGroupCommand,
  UpdateAutoScalingGroupCommand,
  paginateDescribeAutoScalingGroups,
} from "@aws-sdk/client-auto-scaling";
import {
  DeleteLoadBalancerCommand,
  DeleteTargetGroupCommand,
  DescribeTargetGroupsCommand,
  ElasticLoadBalancingV2Client,
} from "@aws-sdk/client-elastic-load-balancing-v2";

import {
  ScenarioOutput,
  ScenarioInput,
  ScenarioAction,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";
import { loadState } from "@aws-doc-sdk-examples/lib/scenario/steps-common.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

import { MESSAGES, NAMES } from "./constants.js";
import { findLoadBalancer } from "./shared.js";

/**
 * @type {import('@aws-doc-sdk-examples/lib/scenario.js').Step[]}
 */
export const destroySteps = [
  loadState,
  new ScenarioInput("destroy", MESSAGES.destroy, { type: "confirm" }),
  new ScenarioAction(
    "abort",
    (state) => state.destroy === false && process.exit(),
  ),
  new ScenarioAction("deleteTable", async (c) => {
    try {
      const client = new DynamoDBClient({});
```

```
      await client.send(new DeleteTableCommand({ TableName: NAMES.tableName }));
    } catch (e) {
      c.deleteTableError = e;
    }
  }),
  new ScenarioOutput("deleteTableResult", (state) => {
    if (state.deleteTableError) {
      console.error(state.deleteTableError);
      return MESSAGES.deleteTableError.replace(
        "${TABLE_NAME}",
        NAMES.tableName,
      );
    }
    return MESSAGES.deletedTable.replace("${TABLE_NAME}", NAMES.tableName);
  }),
  new ScenarioAction("deleteKeyPair", async (state) => {
    try {
      const client = new EC2Client({});
      await client.send(
        new DeleteKeyPairCommand({ KeyName: NAMES.keyPairName }),
      );
      unlinkSync(`${NAMES.keyPairName}.pem`);
    } catch (e) {
      state.deleteKeyPairError = e;
    }
  }),
  new ScenarioOutput("deleteKeyPairResult", (state) => {
    if (state.deleteKeyPairError) {
      console.error(state.deleteKeyPairError);
      return MESSAGES.deleteKeyPairError.replace(
        "${KEY_PAIR_NAME}",
        NAMES.keyPairName,
      );
    }
    return MESSAGES.deletedKeyPair.replace(
      "${KEY_PAIR_NAME}",
      NAMES.keyPairName,
    );
  }),
  new ScenarioAction("detachPolicyFromRole", async (state) => {
    try {
      const client = new IAMClient({});
      const policy = await findPolicy(NAMES.instancePolicyName);
```

```
      if (!policy) {
        state.detachPolicyFromRoleError = new Error(
          `Policy ${NAMES.instancePolicyName} not found.`,
        );
      } else {
        await client.send(
          new DetachRolePolicyCommand({
            RoleName: NAMES.instanceRoleName,
            PolicyArn: policy.Arn,
          }),
        );
      }
    } catch (e) {
      state.detachPolicyFromRoleError = e;
    }
  }),
  new ScenarioOutput("detachedPolicyFromRole", (state) => {
    if (state.detachPolicyFromRoleError) {
      console.error(state.detachPolicyFromRoleError);
      return MESSAGES.detachPolicyFromRoleError
        .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
        .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
    }
    return MESSAGES.detachedPolicyFromRole
      .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
  }),
  new ScenarioAction("deleteInstancePolicy", async (state) => {
    const client = new IAMClient({});
    const policy = await findPolicy(NAMES.instancePolicyName);

    if (!policy) {
      state.deletePolicyError = new Error(
        `Policy ${NAMES.instancePolicyName} not found.`,
      );
    } else {
      return client.send(
        new DeletePolicyCommand({
          PolicyArn: policy.Arn,
        }),
      );
    }
  }),
  new ScenarioOutput("deletePolicyResult", (state) => {
```

```
    if (state.deletePolicyError) {
      console.error(state.deletePolicyError);
      return MESSAGES.deletePolicyError.replace(
        "${INSTANCE_POLICY_NAME}",
        NAMES.instancePolicyName,
      );
    }
    return MESSAGES.deletedPolicy.replace(
      "${INSTANCE_POLICY_NAME}",
      NAMES.instancePolicyName,
    );
  }),
  new ScenarioAction("removeRoleFromInstanceProfile", async (state) => {
    try {
      const client = new IAMClient({});
      await client.send(
        new RemoveRoleFromInstanceProfileCommand({
          RoleName: NAMES.instanceRoleName,
          InstanceProfileName: NAMES.instanceProfileName,
        }),
      );
    } catch (e) {
      state.removeRoleFromInstanceProfileError = e;
    }
  }),
  new ScenarioOutput("removeRoleFromInstanceProfileResult", (state) => {
    if (state.removeRoleFromInstanceProfile) {
      console.error(state.removeRoleFromInstanceProfileError);
      return MESSAGES.removeRoleFromInstanceProfileError
        .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
        .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
    }
    return MESSAGES.removedRoleFromInstanceProfile
      .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
  }),
  new ScenarioAction("deleteInstanceRole", async (state) => {
    try {
      const client = new IAMClient({});
      await client.send(
        new DeleteRoleCommand({
          RoleName: NAMES.instanceRoleName,
        }),
      );
```

```
    } catch (e) {
      state.deleteInstanceRoleError = e;
    }
  }),
  new ScenarioOutput("deleteInstanceRoleResult", (state) => {
    if (state.deleteInstanceRoleError) {
      console.error(state.deleteInstanceRoleError);
      return MESSAGES.deleteInstanceRoleError.replace(
        "${INSTANCE_ROLE_NAME}",
        NAMES.instanceRoleName,
      );
    }
    return MESSAGES.deletedInstanceRole.replace(
      "${INSTANCE_ROLE_NAME}",
      NAMES.instanceRoleName,
    );
  }),
  new ScenarioAction("deleteInstanceProfile", async (state) => {
    try {
      const client = new IAMClient({});
      await client.send(
        new DeleteInstanceProfileCommand({
          InstanceProfileName: NAMES.instanceProfileName,
        }),
      );
    } catch (e) {
      state.deleteInstanceProfileError = e;
    }
  }),
  new ScenarioOutput("deleteInstanceProfileResult", (state) => {
    if (state.deleteInstanceProfileError) {
      console.error(state.deleteInstanceProfileError);
      return MESSAGES.deleteInstanceProfileError.replace(
        "${INSTANCE_PROFILE_NAME}",
        NAMES.instanceProfileName,
      );
    }
    return MESSAGES.deletedInstanceProfile.replace(
      "${INSTANCE_PROFILE_NAME}",
      NAMES.instanceProfileName,
    );
  }),
  new ScenarioAction("deleteAutoScalingGroup", async (state) => {
    try {
```

```
      await terminateGroupInstances(NAMES.autoScalingGroupName);
      await retry({ intervalInMs: 60000, maxRetries: 60 }, async () => {
        await deleteAutoScalingGroup(NAMES.autoScalingGroupName);
      });
    } catch (e) {
      state.deleteAutoScalingGroupError = e;
    }
  }),
  new ScenarioOutput("deleteAutoScalingGroupResult", (state) => {
    if (state.deleteAutoScalingGroupError) {
      console.error(state.deleteAutoScalingGroupError);
      return MESSAGES.deleteAutoScalingGroupError.replace(
        "${AUTO_SCALING_GROUP_NAME}",
        NAMES.autoScalingGroupName,
      );
    }
    return MESSAGES.deletedAutoScalingGroup.replace(
      "${AUTO_SCALING_GROUP_NAME}",
      NAMES.autoScalingGroupName,
    );
  }),
  new ScenarioAction("deleteLaunchTemplate", async (state) => {
    const client = new EC2Client({});
    try {
      await client.send(
        new DeleteLaunchTemplateCommand({
          LaunchTemplateName: NAMES.launchTemplateName,
        }),
      );
    } catch (e) {
      state.deleteLaunchTemplateError = e;
    }
  }),
  new ScenarioOutput("deleteLaunchTemplateResult", (state) => {
    if (state.deleteLaunchTemplateError) {
      console.error(state.deleteLaunchTemplateError);
      return MESSAGES.deleteLaunchTemplateError.replace(
        "${LAUNCH_TEMPLATE_NAME}",
        NAMES.launchTemplateName,
      );
    }
    return MESSAGES.deletedLaunchTemplate.replace(
      "${LAUNCH_TEMPLATE_NAME}",
      NAMES.launchTemplateName,
```

```
    );
  }),
  new ScenarioAction("deleteLoadBalancer", async (state) => {
    try {
      const client = new ElasticLoadBalancingV2Client({});
      const loadBalancer = await findLoadBalancer(NAMES.loadBalancerName);
      await client.send(
        new DeleteLoadBalancerCommand({
          LoadBalancerArn: loadBalancer.LoadBalancerArn,
        }),
      );
      await retry({ intervalInMs: 1000, maxRetries: 60 }, async () => {
        const lb = await findLoadBalancer(NAMES.loadBalancerName);
        if (lb) {
          throw new Error("Load balancer still exists.");
        }
      });
    } catch (e) {
      state.deleteLoadBalancerError = e;
    }
  }),
  new ScenarioOutput("deleteLoadBalancerResult", (state) => {
    if (state.deleteLoadBalancerError) {
      console.error(state.deleteLoadBalancerError);
      return MESSAGES.deleteLoadBalancerError.replace(
        "${LB_NAME}",
        NAMES.loadBalancerName,
      );
    }
    return MESSAGES.deletedLoadBalancer.replace(
      "${LB_NAME}",
      NAMES.loadBalancerName,
    );
  }),
  new ScenarioAction("deleteLoadBalancerTargetGroup", async (state) => {
    const client = new ElasticLoadBalancingV2Client({});
    try {
      const { TargetGroups } = await client.send(
        new DescribeTargetGroupsCommand({
          Names: [NAMES.loadBalancerTargetGroupName],
        }),
      );

      await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
```

```
        client.send(
          new DeleteTargetGroupCommand({
            TargetGroupArn: TargetGroups[0].TargetGroupArn,
          }),
        ),
      );
    } catch (e) {
      state.deleteLoadBalancerTargetGroupError = e;
    }
  }),
  new ScenarioOutput("deleteLoadBalancerTargetGroupResult", (state) => {
    if (state.deleteLoadBalancerTargetGroupError) {
      console.error(state.deleteLoadBalancerTargetGroupError);
      return MESSAGES.deleteLoadBalancerTargetGroupError.replace(
        "${TARGET_GROUP_NAME}",
        NAMES.loadBalancerTargetGroupName,
      );
    }
    return MESSAGES.deletedLoadBalancerTargetGroup.replace(
      "${TARGET_GROUP_NAME}",
      NAMES.loadBalancerTargetGroupName,
    );
  }),
  new ScenarioAction("detachSsmOnlyRoleFromProfile", async (state) => {
    try {
      const client = new IAMClient({});
      await client.send(
        new RemoveRoleFromInstanceProfileCommand({
          InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
          RoleName: NAMES.ssmOnlyRoleName,
        }),
      );
    } catch (e) {
      state.detachSsmOnlyRoleFromProfileError = e;
    }
  }),
  new ScenarioOutput("detachSsmOnlyRoleFromProfileResult", (state) => {
    if (state.detachSsmOnlyRoleFromProfileError) {
      console.error(state.detachSsmOnlyRoleFromProfileError);
      return MESSAGES.detachSsmOnlyRoleFromProfileError
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${PROFILE_NAME}", NAMES.ssmOnlyInstanceProfileName);
    }
    return MESSAGES.detachedSsmOnlyRoleFromProfile
```

```
      .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
      .replace("${PROFILE_NAME}", NAMES.ssmOnlyInstanceProfileName);
  }),
  new ScenarioAction("detachSsmOnlyCustomRolePolicy", async (state) => {
    try {
      const iamClient = new IAMClient({});
      const ssmOnlyPolicy = await findPolicy(NAMES.ssmOnlyPolicyName);
      await iamClient.send(
        new DetachRolePolicyCommand({
          RoleName: NAMES.ssmOnlyRoleName,
          PolicyArn: ssmOnlyPolicy.Arn,
        }),
      );
    } catch (e) {
      state.detachSsmOnlyCustomRolePolicyError = e;
    }
  }),
  new ScenarioOutput("detachSsmOnlyCustomRolePolicyResult", (state) => {
    if (state.detachSsmOnlyCustomRolePolicyError) {
      console.error(state.detachSsmOnlyCustomRolePolicyError);
      return MESSAGES.detachSsmOnlyCustomRolePolicyError
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${POLICY_NAME}", NAMES.ssmOnlyPolicyName);
    }
    return MESSAGES.detachedSsmOnlyCustomRolePolicy
      .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
      .replace("${POLICY_NAME}", NAMES.ssmOnlyPolicyName);
  }),
  new ScenarioAction("detachSsmOnlyAWSRolePolicy", async (state) => {
    try {
      const iamClient = new IAMClient({});
      await iamClient.send(
        new DetachRolePolicyCommand({
          RoleName: NAMES.ssmOnlyRoleName,
          PolicyArn: "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
        }),
      );
    } catch (e) {
      state.detachSsmOnlyAWSRolePolicyError = e;
    }
  }),
  new ScenarioOutput("detachSsmOnlyAWSRolePolicyResult", (state) => {
    if (state.detachSsmOnlyAWSRolePolicyError) {
      console.error(state.detachSsmOnlyAWSRolePolicyError);
```

```
      return MESSAGES.detachSsmOnlyAWSRolePolicyError
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${POLICY_NAME}", "AmazonSSMManagedInstanceCore");
    }
    return MESSAGES.detachedSsmOnlyAWSRolePolicy
      .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
      .replace("${POLICY_NAME}", "AmazonSSMManagedInstanceCore");
  }),
  new ScenarioAction("deleteSsmOnlyInstanceProfile", async (state) => {
    try {
      const iamClient = new IAMClient({});
      await iamClient.send(
        new DeleteInstanceProfileCommand({
          InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
        }),
      );
    } catch (e) {
      state.deleteSsmOnlyInstanceProfileError = e;
    }
  }),
  new ScenarioOutput("deleteSsmOnlyInstanceProfileResult", (state) => {
    if (state.deleteSsmOnlyInstanceProfileError) {
      console.error(state.deleteSsmOnlyInstanceProfileError);
      return MESSAGES.deleteSsmOnlyInstanceProfileError.replace(
        "${INSTANCE_PROFILE_NAME}",
        NAMES.ssmOnlyInstanceProfileName,
      );
    }
    return MESSAGES.deletedSsmOnlyInstanceProfile.replace(
      "${INSTANCE_PROFILE_NAME}",
      NAMES.ssmOnlyInstanceProfileName,
    );
  }),
  new ScenarioAction("deleteSsmOnlyPolicy", async (state) => {
    try {
      const iamClient = new IAMClient({});
      const ssmOnlyPolicy = await findPolicy(NAMES.ssmOnlyPolicyName);
      await iamClient.send(
        new DeletePolicyCommand({
          PolicyArn: ssmOnlyPolicy.Arn,
        }),
      );
    } catch (e) {
      state.deleteSsmOnlyPolicyError = e;
```

```
    }
  }),
  new ScenarioOutput("deleteSsmOnlyPolicyResult", (state) => {
    if (state.deleteSsmOnlyPolicyError) {
      console.error(state.deleteSsmOnlyPolicyError);
      return MESSAGES.deleteSsmOnlyPolicyError.replace(
        "${POLICY_NAME}",
        NAMES.ssmOnlyPolicyName,
      );
    }
    return MESSAGES.deletedSsmOnlyPolicy.replace(
      "${POLICY_NAME}",
      NAMES.ssmOnlyPolicyName,
    );
  }),
  new ScenarioAction("deleteSsmOnlyRole", async (state) => {
    try {
      const iamClient = new IAMClient({});
      await iamClient.send(
        new DeleteRoleCommand({
          RoleName: NAMES.ssmOnlyRoleName,
        }),
      );
    } catch (e) {
      state.deleteSsmOnlyRoleError = e;
    }
  }),
  new ScenarioOutput("deleteSsmOnlyRoleResult", (state) => {
    if (state.deleteSsmOnlyRoleError) {
      console.error(state.deleteSsmOnlyRoleError);
      return MESSAGES.deleteSsmOnlyRoleError.replace(
        "${ROLE_NAME}",
        NAMES.ssmOnlyRoleName,
      );
    }
    return MESSAGES.deletedSsmOnlyRole.replace(
      "${ROLE_NAME}",
      NAMES.ssmOnlyRoleName,
    );
  }),
  new ScenarioAction(
    "revokeSecurityGroupIngress",
    async (
```

```
      /** @type {{ myIp: string, defaultSecurityGroup: { GroupId: string } }} */
 state,
    ) => {
      const ec2Client = new EC2Client({});

      try {
        await ec2Client.send(
          new RevokeSecurityGroupIngressCommand({
            GroupId: state.defaultSecurityGroup.GroupId,
            CidrIp: `${state.myIp}/32`,
            FromPort: 80,
            ToPort: 80,
            IpProtocol: "tcp",
          }),
        );
      } catch (e) {
        state.revokeSecurityGroupIngressError = e;
      }
    },
  ),
  new ScenarioOutput("revokeSecurityGroupIngressResult", (state) => {
    if (state.revokeSecurityGroupIngressError) {
      console.error(state.revokeSecurityGroupIngressError);
      return MESSAGES.revokeSecurityGroupIngressError.replace(
        "${IP}",
        state.myIp,
      );
    }
    return MESSAGES.revokedSecurityGroupIngress.replace("${IP}", state.myIp);
  }),
];

/**
 * @param {string} policyName
 */
async function findPolicy(policyName) {
  const client = new IAMClient({});
  const paginatedPolicies = paginateListPolicies({ client }, {});
  for await (const page of paginatedPolicies) {
    const policy = page.Policies.find((p) => p.PolicyName === policyName);
    if (policy) {
      return policy;
    }
  }
```

```
}

/**
 * @param {string} groupName
 */
async function deleteAutoScalingGroup(groupName) {
  const client = new AutoScalingClient({});
  try {
    await client.send(
      new DeleteAutoScalingGroupCommand({
        AutoScalingGroupName: groupName,
      }),
    );
  } catch (err) {
    if (!(err instanceof Error)) {
      throw err;
    }
    console.log(err.name);
    throw err;
  }
}

/**
 * @param {string} groupName
 */
async function terminateGroupInstances(groupName) {
  const autoScalingClient = new AutoScalingClient({});
  const group = await findAutoScalingGroup(groupName);
  await autoScalingClient.send(
    new UpdateAutoScalingGroupCommand({
      AutoScalingGroupName: group.AutoScalingGroupName,
      MinSize: 0,
    }),
  );
  for (const i of group.Instances) {
    await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
      autoScalingClient.send(
        new TerminateInstanceInAutoScalingGroupCommand({
          InstanceId: i.InstanceId,
          ShouldDecrementDesiredCapacity: true,
        }),
      ),
    );
  }
```

```
}

async function findAutoScalingGroup(groupName) {
  const client = new AutoScalingClient({});
  const paginatedGroups = paginateDescribeAutoScalingGroups({ client }, {});
  for await (const page of paginatedGroups) {
    const group = page.AutoScalingGroups.find(
      (g) => g.AutoScalingGroupName === groupName,
    );
    if (group) {
      return group;
    }
  }
  throw new Error(`Auto scaling group ${groupName} not found.`);
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per JavaScript .
  - AttachLoadBalancerTargetGroups
  - CreateAutoScalingGroup
  - CreateInstanceProfile
  - CreateLaunchTemplate
  - CreateListener
  - CreateLoadBalancer
  - CreateTargetGroup
  - DeleteAutoScalingGroup
  - DeleteInstanceProfile
  - DeleteLaunchTemplate
  - DeleteLoadBalancer
  - DeleteTargetGroup
  - DescribeAutoScalingGroups
  - DescribeAvailabilityZones
  - DescribeIamInstanceProfileAssociations
  - DescribeInstances
  - DescribeLoadBalancers

- DescribeSubnets
- DescribeTargetGroups
- DescribeTargetHealth
- DescribeVpcs
- RebootInstances
- ReplaceIamInstanceProfileAssociation
- TerminateInstanceInAutoScalingGroup
- UpdateAutoScalingGroup

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Esegui lo scenario interattivo al prompt dei comandi.

```
class Runner:
    """
    Manages the deployment, demonstration, and destruction of resources for the
 resilient service.
    """

    def __init__(
        self,
        resource_path: str,
        recommendation: RecommendationService,
        autoscaler: AutoScalingWrapper,
        loadbalancer: ElasticLoadBalancerWrapper,
        param_helper: ParameterHelper,
    ):
        """
        Initializes the Runner class with the necessary parameters.
```

```
        :param resource_path: The path to resource files used by this example,
such as IAM policies and instance scripts.
        :param recommendation: An instance of the RecommendationService class.
        :param autoscaler: An instance of the AutoScaler class.
        :param loadbalancer: An instance of the LoadBalancer class.
        :param param_helper: An instance of the ParameterHelper class.
        """
        self.resource_path = resource_path
        self.recommendation = recommendation
        self.autoscaler = autoscaler
        self.loadbalancer = loadbalancer
        self.param_helper = param_helper
        self.protocol = "HTTP"
        self.port = 80
        self.ssh_port = 22

        prefix = "doc-example-resilience"
        self.target_group_name = f"{prefix}-tg"
        self.load_balancer_name = f"{prefix}-lb"

    def deploy(self) -> None:
        """
        Deploys the resources required for the resilient service, including the
DynamoDB table,
        EC2 instances, Auto Scaling group, and load balancer.
        """
        recommendations_path = f"{self.resource_path}/recommendations.json"
        startup_script = f"{self.resource_path}/server_startup_script.sh"
        instance_policy = f"{self.resource_path}/instance_policy.json"

        logging.info("Starting deployment of resources for the resilient
service.")

        logging.info(
            "Creating and populating DynamoDB table '%s'.",
            self.recommendation.table_name,
        )
        self.recommendation.create()
        self.recommendation.populate(recommendations_path)

        logging.info(
            "Creating an EC2 launch template with the startup script '%s'.",
            startup_script,
        )
```

```
        self.autoscaler.create_template(startup_script, instance_policy)

        logging.info(
            "Creating an EC2 Auto Scaling group across multiple Availability
 Zones."
        )
        zones = self.autoscaler.create_autoscaling_group(3)

        logging.info("Creating variables that control the flow of the demo.")
        self.param_helper.reset()

        logging.info("Creating Elastic Load Balancing target group and load
 balancer.")

        vpc = self.autoscaler.get_default_vpc()
        subnets = self.autoscaler.get_subnets(vpc["VpcId"], zones)
        target_group = self.loadbalancer.create_target_group(
            self.target_group_name, self.protocol, self.port, vpc["VpcId"]
        )
        self.loadbalancer.create_load_balancer(
            self.load_balancer_name, [subnet["SubnetId"] for subnet in subnets]
        )
        self.loadbalancer.create_listener(self.load_balancer_name, target_group)

        self.autoscaler.attach_load_balancer_target_group(target_group)

        logging.info("Verifying access to the load balancer endpoint.")
        endpoint = self.loadbalancer.get_endpoint(self.load_balancer_name)
        lb_success = self.loadbalancer.verify_load_balancer_endpoint(endpoint)
        current_ip_address = requests.get("http://
checkip.amazonaws.com").text.strip()

        if not lb_success:
            logging.warning(
                "Couldn't connect to the load balancer. Verifying that the port
 is open..."
            )
            sec_group, port_is_open = self.autoscaler.verify_inbound_port(
                vpc, self.port, current_ip_address
            )
            sec_group, ssh_port_is_open = self.autoscaler.verify_inbound_port(
                vpc, self.ssh_port, current_ip_address
            )
            if not port_is_open:
```

```
                logging.warning(
                    "The default security group for your VPC must allow access
from this computer."
                )
                if q.ask(
                    f"Do you want to add a rule to security group
{sec_group['GroupId']} to allow\n"
                    f"inbound traffic on port {self.port} from your computer's IP
address of {current_ip_address}? (y/n) ",
                    q.is_yesno,
                ):
                    self.autoscaler.open_inbound_port(
                        sec_group["GroupId"], self.port, current_ip_address
                    )
            if not ssh_port_is_open:
                if q.ask(
                    f"Do you want to add a rule to security group
{sec_group['GroupId']} to allow\n"
                    f"inbound SSH traffic on port {self.ssh_port} for debugging
from your computer's IP address of {current_ip_address}? (y/n) ",
                    q.is_yesno,
                ):
                    self.autoscaler.open_inbound_port(
                        sec_group["GroupId"], self.ssh_port, current_ip_address
                    )
            lb_success =
self.loadbalancer.verify_load_balancer_endpoint(endpoint)

        if lb_success:
            logging.info(
                "Load balancer is ready. Access it at: http://%s",
current_ip_address
            )
        else:
            logging.error(
                "Couldn't get a successful response from the load balancer
endpoint. Please verify your VPC and security group settings."
            )

    def demo_choices(self) -> None:
        """
        Presents choices for interacting with the deployed service, such as
sending requests to
        the load balancer or checking the health of the targets.
```

```
        """
        actions = [
            "Send a GET request to the load balancer endpoint.",
            "Check the health of load balancer targets.",
            "Go to the next part of the demo.",
        ]
        choice = 0
        while choice != 2:
            logging.info("Choose an action to interact with the service.")
            choice = q.choose("Which action would you like to take? ", actions)
            if choice == 0:
                logging.info("Sending a GET request to the load balancer
endpoint.")
                endpoint =
self.loadbalancer.get_endpoint(self.load_balancer_name)
                logging.info("GET http://%s", endpoint)
                response = requests.get(f"http://{endpoint}")
                logging.info("Response: %s", response.status_code)
                if response.headers.get("content-type") == "application/json":
                    pp(response.json())
            elif choice == 1:
                logging.info("Checking the health of load balancer targets.")
                health =
self.loadbalancer.check_target_health(self.target_group_name)
                for target in health:
                    state = target["TargetHealth"]["State"]
                    logging.info(
                        "Target %s on port %d is %s",
                        target["Target"]["Id"],
                        target["Target"]["Port"],
                        state,
                    )
                    if state != "healthy":
                        logging.warning(
                            "%s: %s",
                            target["TargetHealth"]["Reason"],
                            target["TargetHealth"]["Description"],
                        )
                logging.info(
                    "Note that it can take a minute or two for the health check
to update."
                )
            elif choice == 2:
                logging.info("Proceeding to the next part of the demo.")
```

```
   def demo(self) -> None:
       """
       Runs the demonstration, showing how the service responds to different
failure scenarios
       and how a resilient architecture can keep the service running.
       """
       ssm_only_policy = f"{self.resource_path}/ssm_only_policy.json"

       logging.info("Resetting parameters to starting values for the demo.")
       self.param_helper.reset()

       logging.info(
           "Starting demonstration of the service's resilience under various
failure conditions."
       )
       self.demo_choices()

       logging.info(
           "Simulating failure by changing the Systems Manager parameter to a
non-existent table."
       )
       self.param_helper.put(self.param_helper.table, "this-is-not-a-table")
       logging.info("Sending GET requests will now return failure codes.")
       self.demo_choices()

       logging.info("Switching to static response mode to mitigate failure.")
       self.param_helper.put(self.param_helper.failure_response, "static")
       logging.info("Sending GET requests will now return static responses.")
       self.demo_choices()

       logging.info("Restoring normal operation of the recommendation service.")
       self.param_helper.put(self.param_helper.table,
self.recommendation.table_name)

       logging.info(
           "Introducing a failure by assigning bad credentials to one of the
instances."
       )
       self.autoscaler.create_instance_profile(
           ssm_only_policy,
           self.autoscaler.bad_creds_policy_name,
           self.autoscaler.bad_creds_role_name,
           self.autoscaler.bad_creds_profile_name,
```

```
            ["AmazonSSMManagedInstanceCore"],
        )
        instances = self.autoscaler.get_instances()
        bad_instance_id = instances[0]
        instance_profile = self.autoscaler.get_instance_profile(bad_instance_id)
        logging.info(
            "Replacing instance profile with bad credentials for instance %s.",
            bad_instance_id,
        )
        self.autoscaler.replace_instance_profile(
            bad_instance_id,
            self.autoscaler.bad_creds_profile_name,
            instance_profile["AssociationId"],
        )
        logging.info(
            "Sending GET requests may return either a valid recommendation or a
static response."
        )
        self.demo_choices()

        logging.info("Implementing deep health checks to detect unhealthy
instances.")
        self.param_helper.put(self.param_helper.health_check, "deep")
        logging.info("Checking the health of the load balancer targets.")
        self.demo_choices()

        logging.info(
            "Terminating the unhealthy instance to let the auto scaler replace
it."
        )
        self.autoscaler.terminate_instance(bad_instance_id)
        logging.info("The service remains resilient during instance
replacement.")
        self.demo_choices()

        logging.info("Simulating a complete failure of the recommendation
service.")
        self.param_helper.put(self.param_helper.table, "this-is-not-a-table")
        logging.info(
            "All instances will report as unhealthy, but the service will still
return static responses."
        )
        self.demo_choices()
        self.param_helper.reset()
```

```
    def destroy(self, automation=False) -> None:
        """
        Destroys all resources created for the demo, including the load balancer,
 Auto Scaling group,
        EC2 instances, and DynamoDB table.
        """
        logging.info(
            "This concludes the demo. Preparing to clean up all AWS resources
 created during the demo."
        )
        if automation:
            cleanup = True
        else:
            cleanup = q.ask(
                "Do you want to clean up all demo resources? (y/n) ", q.is_yesno
            )

        if cleanup:
            logging.info("Deleting load balancer and related resources.")
            self.loadbalancer.delete_load_balancer(self.load_balancer_name)
            self.loadbalancer.delete_target_group(self.target_group_name)
            self.autoscaler.delete_autoscaling_group(self.autoscaler.group_name)
            self.autoscaler.delete_key_pair()
            self.autoscaler.delete_template()
            self.autoscaler.delete_instance_profile(
                self.autoscaler.bad_creds_profile_name,
                self.autoscaler.bad_creds_role_name,
            )
            logging.info("Deleting DynamoDB table and other resources.")
            self.recommendation.destroy()
        else:
            logging.warning(
                "Resources have not been deleted. Ensure you clean them up
 manually to avoid unexpected charges."
            )


def main() -> None:
    """
    Main function to parse arguments and run the appropriate actions for the
 demo.
    """
    parser = argparse.ArgumentParser()
```

```
    parser.add_argument(
        "--action",
        required=True,
        choices=["all", "deploy", "demo", "destroy"],
        help="The action to take for the demo. When 'all' is specified, resources
are\n"
        "deployed, the demo is run, and resources are destroyed.",
    )
    parser.add_argument(
        "--resource_path",
        default="../../../scenarios/features/resilient_service/resources",
        help="The path to resource files used by this example, such as IAM
policies and\n"
        "instance scripts.",
    )
    args = parser.parse_args()

    logging.info("Starting the Resilient Service demo.")

    prefix = "doc-example-resilience"

    # Service Clients
    ddb_client = boto3.client("dynamodb")
    elb_client = boto3.client("elbv2")
    autoscaling_client = boto3.client("autoscaling")
    ec2_client = boto3.client("ec2")
    ssm_client = boto3.client("ssm")
    iam_client = boto3.client("iam")

    # Wrapper instantiations
    recommendation = RecommendationService(
        "doc-example-recommendation-service", ddb_client
    )
    autoscaling_wrapper = AutoScalingWrapper(
        prefix,
        "t3.micro",
        "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    )
    elb_wrapper = ElasticLoadBalancerWrapper(elb_client)
    param_helper = ParameterHelper(recommendation.table_name, ssm_client)
```

```
    # Demo invocation
    runner = Runner(
        args.resource_path,
        recommendation,
        autoscaling_wrapper,
        elb_wrapper,
        param_helper,
    )
    actions = [args.action] if args.action != "all" else ["deploy", "demo",
 "destroy"]
    for action in actions:
        if action == "deploy":
            runner.deploy()
        elif action == "demo":
            runner.demo()
        elif action == "destroy":
            runner.destroy()

    logging.info("Demo completed successfully.")


if __name__ == "__main__":
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    main()
```

Crea una classe che racchiuda le operazioni di dimensionamento automatico e Amazon EC2.

```
class AutoScalingWrapper:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix: str,
        inst_type: str,
        ami_param: str,
        autoscaling_client: boto3.client,
        ec2_client: boto3.client,
        ssm_client: boto3.client,
        iam_client: boto3.client,
```

```
    ):
        """
        Initializes the AutoScaler class with the necessary parameters.

        :param resource_prefix: The prefix for naming AWS resources that are
created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
that is created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
        self.ami_param = ami_param
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        sts_client = boto3.client("sts")
        self.account_id = sts_client.get_caller_identity()["Account"]

        self.key_pair_name = f"{resource_prefix}-key-pair"
        self.launch_template_name = f"{resource_prefix}-template-"
        self.group_name = f"{resource_prefix}-group"

        # Happy path
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
        self.instance_profile_name = f"{resource_prefix}-prof"

        # Failure mode
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
        self.bad_creds_role_name = f"{resource_prefix}-bc-role"
        self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"


    def create_policy(self, policy_file: str, policy_name: str) -> str:
        """
        Creates a new IAM policy or retrieves the ARN of an existing policy.

        :param policy_file: The path to a JSON file that contains the policy
definition.
```

```
        :param policy_name: The name to give the created policy.
        :return: The ARN of the created or existing policy.
        """
        with open(policy_file) as file:
            policy_doc = file.read()

        try:
            response = self.iam_client.create_policy(
                PolicyName=policy_name, PolicyDocument=policy_doc
            )
            policy_arn = response["Policy"]["Arn"]
            log.info(f"Policy '{policy_name}' created successfully. ARN:
 {policy_arn}")
            return policy_arn

        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                # If the policy already exists, get its ARN
                response = self.iam_client.get_policy(
                    PolicyArn=f"arn:aws:iam::{self.account_id}:policy/
{policy_name}"
                )
                policy_arn = response["Policy"]["Arn"]
                log.info(f"Policy '{policy_name}' already exists. ARN:
 {policy_arn}")
                return policy_arn
            log.error(f"Full error:\n\t{err}")

    def create_role(self, role_name: str, assume_role_doc: dict) -> str:
        """
        Creates a new IAM role or retrieves the ARN of an existing role.

        :param role_name: The name to give the created role.
        :param assume_role_doc: The assume role policy document that specifies
 which
                                entities can assume the role.
        :return: The ARN of the created or existing role.
        """
        try:
            response = self.iam_client.create_role(
                RoleName=role_name,
 AssumeRolePolicyDocument=json.dumps(assume_role_doc)
            )
            role_arn = response["Role"]["Arn"]
```

```
            log.info(f"Role '{role_name}' created successfully. ARN: {role_arn}")
            return role_arn

        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                # If the role already exists, get its ARN
                response = self.iam_client.get_role(RoleName=role_name)
                role_arn = response["Role"]["Arn"]
                log.info(f"Role '{role_name}' already exists. ARN: {role_arn}")
                return role_arn
            log.error(f"Full error:\n\t{err}")

    def attach_policy(
        self,
        role_name: str,
        policy_arn: str,
        aws_managed_policies: Tuple[str, ...] = (),
    ) -> None:
        """
        Attaches an IAM policy to a role and optionally attaches additional AWS-
managed policies.

        :param role_name: The name of the role to attach the policy to.
        :param policy_arn: The ARN of the policy to attach.
        :param aws_managed_policies: A tuple of AWS-managed policy names to
 attach to the role.
        """
        try:
            self.iam_client.attach_role_policy(RoleName=role_name,
 PolicyArn=policy_arn)
            for aws_policy in aws_managed_policies:
                self.iam_client.attach_role_policy(
                    RoleName=role_name,
                    PolicyArn=f"arn:aws:iam::aws:policy/{aws_policy}",
                )
            log.info(f"Attached policy {policy_arn} to role {role_name}.")
        except ClientError as err:
            log.error(f"Failed to attach policy {policy_arn} to role
 {role_name}.")
            log.error(f"Full error:\n\t{err}")

    def create_instance_profile(
        self,
        policy_file: str,
```

```
        policy_name: str,
        role_name: str,
        profile_name: str,
        aws_managed_policies: Tuple[str, ...] = (),
    ) -> str:
        """
        Creates a policy, role, and profile that is associated with instances
created by
        this class. An instance's associated profile defines a role that is
assumed by the
        instance. The role has attached policies that specify the AWS permissions
granted to
        clients that run on the instance.

        :param policy_file: The name of a JSON file that contains the policy
definition to
                            create and attach to the role.
        :param policy_name: The name to give the created policy.
        :param role_name: The name to give the created role.
        :param profile_name: The name to the created profile.
        :param aws_managed_policies: Additional AWS-managed policies that are
attached to
                                     the role, such as
AmazonSSMManagedInstanceCore to grant
                                     use of Systems Manager to send commands to
the instance.
        :return: The ARN of the profile that is created.
        """
        assume_role_doc = {
            "Version":"2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {"Service": "ec2.amazonaws.com"},
                    "Action": "sts:AssumeRole",
                }
            ],
        }
        policy_arn = self.create_policy(policy_file, policy_name)
        self.create_role(role_name, assume_role_doc)
        self.attach_policy(role_name, policy_arn, aws_managed_policies)

        try:
            profile_response = self.iam_client.create_instance_profile(
```

```
                InstanceProfileName=profile_name
            )
            waiter = self.iam_client.get_waiter("instance_profile_exists")
            waiter.wait(InstanceProfileName=profile_name)
            time.sleep(10)  # wait a little longer
            profile_arn = profile_response["InstanceProfile"]["Arn"]
            self.iam_client.add_role_to_instance_profile(
                InstanceProfileName=profile_name, RoleName=role_name
            )
            log.info("Created profile %s and added role %s.", profile_name,
role_name)
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                prof_response = self.iam_client.get_instance_profile(
                    InstanceProfileName=profile_name
                )
                profile_arn = prof_response["InstanceProfile"]["Arn"]
                log.info(
                    "Instance profile %s already exists, nothing to do.",
profile_name
                )
            log.error(f"Full error:\n\t{err}")
        return profile_arn


    def get_instance_profile(self, instance_id: str) -> Dict[str, Any]:
        """
        Gets data about the profile associated with an instance.

        :param instance_id: The ID of the instance to look up.
        :return: The profile data.
        """
        try:
            response =
self.ec2_client.describe_iam_instance_profile_associations(
                Filters=[{"Name": "instance-id", "Values": [instance_id]}]
            )
            if not response["IamInstanceProfileAssociations"]:
                log.info(f"No instance profile found for instance
{instance_id}.")
            profile_data = response["IamInstanceProfileAssociations"][0]
            log.info(f"Retrieved instance profile for instance {instance_id}.")
            return profile_data
        except ClientError as err:
```

```
            log.error(
                f"Failed to retrieve instance profile for instance
{instance_id}."
            )
            error_code = err.response["Error"]["Code"]
            if error_code == "InvalidInstanceID.NotFound":
                log.error(f"The instance ID '{instance_id}' does not exist.")
            log.error(f"Full error:\n\t{err}")


    def replace_instance_profile(
        self,
        instance_id: str,
        new_instance_profile_name: str,
        profile_association_id: str,
    ) -> None:
        """
        Replaces the profile associated with a running instance. After the
profile is
        replaced, the instance is rebooted to ensure that it uses the new
profile. When
        the instance is ready, Systems Manager is used to restart the Python web
server.

        :param instance_id: The ID of the instance to restart.
        :param new_instance_profile_name: The name of the new profile to
associate with
                                          the specified instance.
        :param profile_association_id: The ID of the existing profile association
for the
                                       instance.
        """
        try:
            self.ec2_client.replace_iam_instance_profile_association(
                IamInstanceProfile={"Name": new_instance_profile_name},
                AssociationId=profile_association_id,
            )
            log.info(
                "Replaced instance profile for association %s with profile %s.",
                profile_association_id,
                new_instance_profile_name,
            )
            time.sleep(5)
```

```
            self.ec2_client.reboot_instances(InstanceIds=[instance_id])
            log.info("Rebooting instance %s.", instance_id)
            waiter = self.ec2_client.get_waiter("instance_running")
            log.info("Waiting for instance %s to be running.", instance_id)
            waiter.wait(InstanceIds=[instance_id])
            log.info("Instance %s is now running.", instance_id)

            self.ssm_client.send_command(
                InstanceIds=[instance_id],
                DocumentName="AWS-RunShellScript",
                Parameters={"commands": ["cd / && sudo python3 server.py 80"]},
            )
            log.info(f"Restarted the Python web server on instance
'{instance_id}'.")
        except ClientError as err:
            log.error("Failed to replace instance profile.")
            error_code = err.response["Error"]["Code"]
            if error_code == "InvalidAssociationID.NotFound":
                log.error(
                    f"Association ID '{profile_association_id}' does not exist."
                    "Please check the association ID and try again."
                )
            if error_code == "InvalidInstanceId":
                log.error(
                    f"The specified instance ID '{instance_id}' does not exist or
is not available for SSM. "
                    f"Please verify the instance ID and try again."
                )
            log.error(f"Full error:\n\t{err}")


    def delete_instance_profile(self, profile_name: str, role_name: str) -> None:
        """
        Detaches a role from an instance profile, detaches policies from the
role,
        and deletes all the resources.

        :param profile_name: The name of the profile to delete.
        :param role_name: The name of the role to delete.
        """
        try:
            self.iam_client.remove_role_from_instance_profile(
                InstanceProfileName=profile_name, RoleName=role_name
            )
```

```
self.iam_client.delete_instance_profile(InstanceProfileName=profile_name)
            log.info("Deleted instance profile %s.", profile_name)
            attached_policies = self.iam_client.list_attached_role_policies(
                RoleName=role_name
            )
            for pol in attached_policies["AttachedPolicies"]:
                self.iam_client.detach_role_policy(
                    RoleName=role_name, PolicyArn=pol["PolicyArn"]
                )
                if not pol["PolicyArn"].startswith("arn:aws:iam::aws"):
                    self.iam_client.delete_policy(PolicyArn=pol["PolicyArn"])
                log.info("Detached and deleted policy %s.", pol["PolicyName"])
            self.iam_client.delete_role(RoleName=role_name)
            log.info("Deleted role %s.", role_name)
        except ClientError as err:
            log.error(
                f"Couldn't delete instance profile {profile_name} or detach "
                f"policies and delete role {role_name}: {err}"
            )
            if err.response["Error"]["Code"] == "NoSuchEntity":
                log.info(
                    "Instance profile %s doesn't exist, nothing to do.",
profile_name
                )


    def create_key_pair(self, key_pair_name: str) -> None:
        """
        Creates a new key pair.

        :param key_pair_name: The name of the key pair to create.
        """
        try:
            response = self.ec2_client.create_key_pair(KeyName=key_pair_name)
            with open(f"{key_pair_name}.pem", "w") as file:
                file.write(response["KeyMaterial"])
            chmod(f"{key_pair_name}.pem", 0o600)
            log.info("Created key pair %s.", key_pair_name)
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(f"Failed to create key pair {key_pair_name}.")
            if error_code == "InvalidKeyPair.Duplicate":
```

```
                log.error(f"A key pair with the name '{key_pair_name}' already
exists.")
            log.error(f"Full error:\n\t{err}")


    def delete_key_pair(self) -> None:
        """
        Deletes a key pair.
        """
        try:
            self.ec2_client.delete_key_pair(KeyName=self.key_pair_name)
            remove(f"{self.key_pair_name}.pem")
            log.info("Deleted key pair %s.", self.key_pair_name)
        except ClientError as err:
            log.error(f"Couldn't delete key pair '{self.key_pair_name}'.")
            log.error(f"Full error:\n\t{err}")
        except FileNotFoundError as err:
            log.info("Key pair %s doesn't exist, nothing to do.",
self.key_pair_name)
            log.error(f"Full error:\n\t{err}")


    def create_template(
        self, server_startup_script_file: str, instance_policy_file: str
    ) -> Dict[str, Any]:
        """
        Creates an Amazon EC2 launch template to use with Amazon EC2 Auto
Scaling. The
        launch template specifies a Bash script in its user data field that runs
after
        the instance is started. This script installs Python packages and starts
a
        Python web server on the instance.

        :param server_startup_script_file: The path to a Bash script file that is
run
                                           when an instance starts.
        :param instance_policy_file: The path to a file that defines a
permissions policy
                                     to create and attach to the instance
profile.
        :return: Information about the newly created template.
        """
        template = {}
```

```
        try:
            # Create key pair and instance profile
            self.create_key_pair(self.key_pair_name)
            self.create_instance_profile(
                instance_policy_file,
                self.instance_policy_name,
                self.instance_role_name,
                self.instance_profile_name,
            )

            # Read the startup script
            with open(server_startup_script_file) as file:
                start_server_script = file.read()

            # Get the latest AMI ID
            ami_latest = self.ssm_client.get_parameter(Name=self.ami_param)
            ami_id = ami_latest["Parameter"]["Value"]

            # Create the launch template
            lt_response = self.ec2_client.create_launch_template(
                LaunchTemplateName=self.launch_template_name,
                LaunchTemplateData={
                    "InstanceType": self.inst_type,
                    "ImageId": ami_id,
                    "IamInstanceProfile": {"Name": self.instance_profile_name},
                    "UserData": base64.b64encode(
                        start_server_script.encode(encoding="utf-8")
                    ).decode(encoding="utf-8"),
                    "KeyName": self.key_pair_name,
                },
            )
            template = lt_response["LaunchTemplate"]
            log.info(
                f"Created launch template {self.launch_template_name} for AMI
{ami_id} on {self.inst_type}."
            )
        except ClientError as err:
            log.error(f"Failed to create launch template
{self.launch_template_name}.")
            error_code = err.response["Error"]["Code"]
            if error_code == "InvalidLaunchTemplateName.AlreadyExistsException":
                log.info(
                    f"Launch template {self.launch_template_name} already exists,
nothing to do."
```

```
                )
            log.error(f"Full error:\n\t{err}")
        return template


    def delete_template(self):
        """
        Deletes a launch template.
        """
        try:
            self.ec2_client.delete_launch_template(
                LaunchTemplateName=self.launch_template_name
            )
            self.delete_instance_profile(
                self.instance_profile_name, self.instance_role_name
            )
            log.info("Launch template %s deleted.", self.launch_template_name)
        except ClientError as err:
            if (
                err.response["Error"]["Code"]
                == "InvalidLaunchTemplateName.NotFoundException"
            ):
                log.info(
                    "Launch template %s does not exist, nothing to do.",
                    self.launch_template_name,
                )
            log.error(f"Full error:\n\t{err}")


    def get_availability_zones(self) -> List[str]:
        """
        Gets a list of Availability Zones in the AWS Region of the Amazon EC2
client.

        :return: The list of Availability Zones for the client Region.
        """
        try:
            response = self.ec2_client.describe_availability_zones()
            zones = [zone["ZoneName"] for zone in response["AvailabilityZones"]]
            log.info(f"Retrieved {len(zones)} availability zones: {zones}.")
        except ClientError as err:
            log.error("Failed to retrieve availability zones.")
            log.error(f"Full error:\n\t{err}")
        else:
```

```
            return zones


    def create_autoscaling_group(self, group_size: int) -> List[str]:
        """
        Creates an EC2 Auto Scaling group with the specified size.

        :param group_size: The number of instances to set for the minimum and
maximum in
                           the group.
        :return: The list of Availability Zones specified for the group.
        """
        try:
            zones = self.get_availability_zones()
            self.autoscaling_client.create_auto_scaling_group(
                AutoScalingGroupName=self.group_name,
                AvailabilityZones=zones,
                LaunchTemplate={
                    "LaunchTemplateName": self.launch_template_name,
                    "Version": "$Default",
                },
                MinSize=group_size,
                MaxSize=group_size,
            )
            log.info(
                f"Created EC2 Auto Scaling group {self.group_name} with
availability zones {zones}."
            )
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            if error_code == "AlreadyExists":
                log.info(
                    f"EC2 Auto Scaling group {self.group_name} already exists,
nothing to do."
                )
            else:
                log.error(f"Failed to create EC2 Auto Scaling group
{self.group_name}.")
                log.error(f"Full error:\n\t{err}")
        else:
            return zones


    def get_instances(self) -> List[str]:
```

```
        """
        Gets data about the instances in the EC2 Auto Scaling group.

        :return: A list of instance IDs in the Auto Scaling group.
        """
        try:
            as_response = self.autoscaling_client.describe_auto_scaling_groups(
                AutoScalingGroupNames=[self.group_name]
            )
            instance_ids = [
                i["InstanceId"]
                for i in as_response["AutoScalingGroups"][0]["Instances"]
            ]
            log.info(
                f"Retrieved {len(instance_ids)} instances for Auto Scaling group
{self.group_name}."
            )
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(
                f"Failed to retrieve instances for Auto Scaling group
{self.group_name}."
            )
            if error_code == "ResourceNotFound":
                log.error(f"The Auto Scaling group '{self.group_name}' does not
exist.")
            log.error(f"Full error:\n\t{err}")
        else:
            return instance_ids


    def terminate_instance(self, instance_id: str, decrementsetting=False) ->
None:
        """
        Terminates an instance in an EC2 Auto Scaling group. After an instance is
        terminated, it can no longer be accessed.

        :param instance_id: The ID of the instance to terminate.
        :param decrementsetting: If True, do not replace terminated instances.
        """
        try:
            self.autoscaling_client.terminate_instance_in_auto_scaling_group(
                InstanceId=instance_id,
                ShouldDecrementDesiredCapacity=decrementsetting,
```

```
            )
            log.info("Terminated instance %s.", instance_id)

            # Adding a waiter to ensure the instance is terminated
            waiter = self.ec2_client.get_waiter("instance_terminated")
            log.info("Waiting for instance %s to be terminated...", instance_id)
            waiter.wait(InstanceIds=[instance_id])
            log.info(
                f"Instance '{instance_id}' has been terminated and will be
replaced."
            )

        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(f"Failed to terminate instance '{instance_id}'.")
            if error_code == "ScalingActivityInProgressFault":
                log.error(
                    "Scaling activity is currently in progress. "
                    "Wait for the scaling activity to complete before attempting
to terminate the instance again."
                )
            elif error_code == "ResourceContentionFault":
                log.error(
                    "The request failed due to a resource contention issue. "
                    "Ensure that no conflicting operations are being performed on
the resource."
                )
            log.error(f"Full error:\n\t{err}")

    def attach_load_balancer_target_group(
        self, lb_target_group: Dict[str, Any]
    ) -> None:
        """
        Attaches an Elastic Load Balancing (ELB) target group to this EC2 Auto
Scaling group.
        The target group specifies how the load balancer forwards requests to the
instances
        in the group.

        :param lb_target_group: Data about the ELB target group to attach.
        """
        try:
            self.autoscaling_client.attach_load_balancer_target_groups(
                AutoScalingGroupName=self.group_name,
```

```
                TargetGroupARNs=[lb_target_group["TargetGroupArn"]],
            )
            log.info(
                "Attached load balancer target group %s to auto scaling group
%s.",
                lb_target_group["TargetGroupName"],
                self.group_name,
            )
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(
                f"Failed to attach load balancer target group
'{lb_target_group['TargetGroupName']}'."
            )
            if error_code == "ResourceContentionFault":
                log.error(
                    "The request failed due to a resource contention issue. "
                    "Ensure that no conflicting operations are being performed on
the resource."
                )
            elif error_code == "ServiceLinkedRoleFailure":
                log.error(
                    "The operation failed because the service-linked role is not
ready or does not exist. "
                    "Check that the service-linked role exists and is correctly
configured."
                )
            log.error(f"Full error:\n\t{err}")


    def delete_autoscaling_group(self, group_name: str) -> None:
        """
        Terminates all instances in the group, then deletes the EC2 Auto Scaling
group.

        :param group_name: The name of the group to delete.
        """
        try:
            response = self.autoscaling_client.describe_auto_scaling_groups(
                AutoScalingGroupNames=[group_name]
            )
            groups = response.get("AutoScalingGroups", [])
            if len(groups) > 0:
                self.autoscaling_client.update_auto_scaling_group(
```

```
                    AutoScalingGroupName=group_name, MinSize=0
                )
                instance_ids = [inst["InstanceId"] for inst in groups[0]
["Instances"]]
                for inst_id in instance_ids:
                    self.terminate_instance(inst_id)

                # Wait for all instances to be terminated
                if instance_ids:
                    waiter = self.ec2_client.get_waiter("instance_terminated")
                    log.info("Waiting for all instances to be terminated...")
                    waiter.wait(InstanceIds=instance_ids)
                    log.info("All instances have been terminated.")
            else:
                log.info(f"No groups found named '{group_name}'! Nothing to do.")
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(f"Failed to delete Auto Scaling group '{group_name}'.")
            if error_code == "ScalingActivityInProgressFault":
                log.error(
                    "Scaling activity is currently in progress. "
                    "Wait for the scaling activity to complete before attempting
 to delete the group again."
                )
            elif error_code == "ResourceContentionFault":
                log.error(
                    "The request failed due to a resource contention issue. "
                    "Ensure that no conflicting operations are being performed on
 the group."
                )
            log.error(f"Full error:\n\t{err}")


    def get_default_vpc(self) -> Dict[str, Any]:
        """
        Gets the default VPC for the account.

        :return: Data about the default VPC.
        """
        try:
            response = self.ec2_client.describe_vpcs(
                Filters=[{"Name": "is-default", "Values": ["true"]}]
            )
        except ClientError as err:
```

```
            error_code = err.response["Error"]["Code"]
            log.error("Failed to retrieve the default VPC.")
            if error_code == "UnauthorizedOperation":
                log.error(
                    "You do not have the necessary permissions to describe VPCs.
 "
                    "Ensure that your AWS IAM user or role has the correct
 permissions."
                )
            elif error_code == "InvalidParameterValue":
                log.error(
                    "One or more parameters are invalid. Check the request
 parameters."
                )

            log.error(f"Full error:\n\t{err}")
        else:
            if "Vpcs" in response and response["Vpcs"]:
                log.info(f"Retrieved default VPC: {response['Vpcs'][0]
['VpcId']}")
                return response["Vpcs"][0]
            else:
                pass


    def verify_inbound_port(
        self, vpc: Dict[str, Any], port: int, ip_address: str
    ) -> Tuple[Dict[str, Any], bool]:
        """
        Verify the default security group of the specified VPC allows ingress
 from this
        computer. This can be done by allowing ingress from this computer's IP
        address. In some situations, such as connecting from a corporate network,
 you
        must instead specify a prefix list ID. You can also temporarily open the
 port to
        any IP address while running this example. If you do, be sure to remove
 public
        access when you're done.

        :param vpc: The VPC used by this example.
        :param port: The port to verify.
        :param ip_address: This computer's IP address.
```

```
        :return: The default security group of the specified VPC, and a value
that indicates
                 whether the specified port is open.
        """
        try:
            response = self.ec2_client.describe_security_groups(
                Filters=[
                    {"Name": "group-name", "Values": ["default"]},
                    {"Name": "vpc-id", "Values": [vpc["VpcId"]]},
                ]
            )
            sec_group = response["SecurityGroups"][0]
            port_is_open = False
            log.info(f"Found default security group {sec_group['GroupId']}.")

            for ip_perm in sec_group["IpPermissions"]:
                if ip_perm.get("FromPort", 0) == port:
                    log.info(f"Found inbound rule: {ip_perm}")
                    for ip_range in ip_perm["IpRanges"]:
                        cidr = ip_range.get("CidrIp", "")
                        if cidr.startswith(ip_address) or cidr == "0.0.0.0/0":
                            port_is_open = True
                    if ip_perm["PrefixListIds"]:
                        port_is_open = True
                    if not port_is_open:
                        log.info(
                            f"The inbound rule does not appear to be open to
either this computer's IP "
                            f"address of {ip_address}, to all IP addresses
(0.0.0.0/0), or to a prefix list ID."
                        )
                    else:
                        break
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(
                f"Failed to verify inbound rule for port {port} for VPC
{vpc['VpcId']}."
            )
            if error_code == "InvalidVpcID.NotFound":
                log.error(
                    f"The specified VPC ID '{vpc['VpcId']}' does not exist.
Please check the VPC ID."
                )
```

```
            log.error(f"Full error:\n\t{err}")
        else:
            return sec_group, port_is_open


    def open_inbound_port(self, sec_group_id: str, port: int, ip_address: str) ->
None:
        """
        Add an ingress rule to the specified security group that allows access on
the
        specified port from the specified IP address.

        :param sec_group_id: The ID of the security group to modify.
        :param port: The port to open.
        :param ip_address: The IP address that is granted access.
        """
        try:
            self.ec2_client.authorize_security_group_ingress(
                GroupId=sec_group_id,
                CidrIp=f"{ip_address}/32",
                FromPort=port,
                ToPort=port,
                IpProtocol="tcp",
            )
            log.info(
                "Authorized ingress to %s on port %s from %s.",
                sec_group_id,
                port,
                ip_address,
            )
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(
                f"Failed to authorize ingress to security group '{sec_group_id}'
on port {port} from {ip_address}."
            )
            if error_code == "InvalidGroupId.Malformed":
                log.error(
                    "The security group ID is malformed. "
                    "Please verify that the security group ID is correct."
                )
            elif error_code == "InvalidPermission.Duplicate":
                log.error(
                    "The specified rule already exists in the security group. "
```

```
                    "Check the existing rules for this security group."
                )
            log.error(f"Full error:\n\t{err}")


    def get_subnets(self, vpc_id: str, zones: List[str] = None) -> List[Dict[str,
Any]]:
        """
        Gets the default subnets in a VPC for a specified list of Availability
Zones.

        :param vpc_id: The ID of the VPC to look up.
        :param zones: The list of Availability Zones to look up.
        :return: The list of subnets found.
        """
        # Ensure that 'zones' is a list, even if None is passed
        if zones is None:
            zones = []
        try:
            paginator = self.ec2_client.get_paginator("describe_subnets")
            page_iterator = paginator.paginate(
                Filters=[
                    {"Name": "vpc-id", "Values": [vpc_id]},
                    {"Name": "availability-zone", "Values": zones},
                    {"Name": "default-for-az", "Values": ["true"]},
                ]
            )

            subnets = []
            for page in page_iterator:
                subnets.extend(page["Subnets"])

            log.info("Found %s subnets for the specified zones.", len(subnets))
            return subnets
        except ClientError as err:
            log.error(
                f"Failed to retrieve subnets for VPC '{vpc_id}' in zones
{zones}."
            )
            error_code = err.response["Error"]["Code"]
            if error_code == "InvalidVpcID.NotFound":
                log.error(
                    "The specified VPC ID does not exist. "
                    "Please check the VPC ID and try again."
```

```
                )
            # Add more error-specific handling as needed
            log.error(f"Full error:\n\t{err}")
```

Crea una classe che racchiuda le operazioni di Elastic Load Balancing.

```
class ElasticLoadBalancerWrapper:
    """Encapsulates Elastic Load Balancing (ELB) actions."""

    def __init__(self, elb_client: boto3.client):
        """
        Initializes the LoadBalancer class with the necessary parameters.
        """
        self.elb_client = elb_client


    def create_target_group(
        self, target_group_name: str, protocol: str, port: int, vpc_id: str
    ) -> Dict[str, Any]:
        """
        Creates an Elastic Load Balancing target group. The target group
 specifies how
        the load balancer forwards requests to instances in the group and how
 instance
        health is checked.

        To speed up this demo, the health check is configured with shortened
 times and
        lower thresholds. In production, you might want to decrease the
 sensitivity of
        your health checks to avoid unwanted failures.

        :param target_group_name: The name of the target group to create.
        :param protocol: The protocol to use to forward requests, such as 'HTTP'.
        :param port: The port to use to forward requests, such as 80.
        :param vpc_id: The ID of the VPC in which the load balancer exists.
        :return: Data about the newly created target group.
        """
        try:
```

```
            response = self.elb_client.create_target_group(
                Name=target_group_name,
                Protocol=protocol,
                Port=port,
                HealthCheckPath="/healthcheck",
                HealthCheckIntervalSeconds=10,
                HealthCheckTimeoutSeconds=5,
                HealthyThresholdCount=2,
                UnhealthyThresholdCount=2,
                VpcId=vpc_id,
            )
            target_group = response["TargetGroups"][0]
            log.info(f"Created load balancing target group
'{target_group_name}'.")
            return target_group
        except ClientError as err:
            log.error(
                f"Couldn't create load balancing target group
'{target_group_name}'."
            )
            error_code = err.response["Error"]["Code"]

            if error_code == "DuplicateTargetGroupName":
                log.error(
                    f"Target group name {target_group_name} already exists. "
                    "Check if the target group already exists."
                    "Consider using a different name or deleting the existing
target group if appropriate."
                )
            elif error_code == "TooManyTargetGroups":
                log.error(
                    "Too many target groups exist in the account. "
                    "Consider deleting unused target groups to create space for
new ones."
                )
            log.error(f"Full error:\n\t{err}")


    def delete_target_group(self, target_group_name) -> None:
        """
        Deletes the target group.
        """
        try:
            # Describe the target group to get its ARN
```

```
            response =
self.elb_client.describe_target_groups(Names=[target_group_name])
            tg_arn = response["TargetGroups"][0]["TargetGroupArn"]

            # Delete the target group
            self.elb_client.delete_target_group(TargetGroupArn=tg_arn)
            log.info("Deleted load balancing target group %s.",
target_group_name)

            # Use a custom waiter to wait until the target group is no longer
available
            self.wait_for_target_group_deletion(self.elb_client, tg_arn)
            log.info("Target group %s successfully deleted.", target_group_name)

        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(f"Failed to delete target group '{target_group_name}'.")
            if error_code == "TargetGroupNotFound":
                log.error(
                    "Load balancer target group either already deleted or never
existed. "
                    "Verify the name and check that the resource exists in the
AWS Console."
                )
            elif error_code == "ResourceInUseException":
                log.error(
                    "Target group still in use by another resource. "
                    "Ensure that the target group is no longer associated with
any load balancers or resources.",
                )
            log.error(f"Full error:\n\t{err}")

    def wait_for_target_group_deletion(
        self, elb_client, target_group_arn, max_attempts=10, delay=30
    ):
        for attempt in range(max_attempts):
            try:

elb_client.describe_target_groups(TargetGroupArns=[target_group_arn])
                print(
                    f"Attempt {attempt + 1}: Target group {target_group_arn}
still exists."
                )
            except ClientError as e:
```

```
                if e.response["Error"]["Code"] == "TargetGroupNotFound":
                    print(
                        f"Target group {target_group_arn} has been successfully
deleted."
                    )
                    return
                else:
                    raise
            time.sleep(delay)
        raise TimeoutError(
            f"Target group {target_group_arn} was not deleted after {max_attempts
* delay} seconds."
        )


    def create_load_balancer(
        self,
        load_balancer_name: str,
        subnet_ids: List[str],
    ) -> Dict[str, Any]:
        """
        Creates an Elastic Load Balancing load balancer that uses the specified
subnets
        and forwards requests to the specified target group.

        :param load_balancer_name: The name of the load balancer to create.
        :param subnet_ids: A list of subnets to associate with the load balancer.
        :return: Data about the newly created load balancer.
        """
        try:
            response = self.elb_client.create_load_balancer(
                Name=load_balancer_name, Subnets=subnet_ids
            )
            load_balancer = response["LoadBalancers"][0]
            log.info(f"Created load balancer '{load_balancer_name}'.")

            waiter = self.elb_client.get_waiter("load_balancer_available")
            log.info(
                f"Waiting for load balancer '{load_balancer_name}' to be
available..."
            )
            waiter.wait(Names=[load_balancer_name])
            log.info(f"Load balancer '{load_balancer_name}' is now available!")
```

```
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(
                f"Failed to create load balancer '{load_balancer_name}'. Error
code: {error_code}, Message: {err.response['Error']['Message']}"
            )

            if error_code == "DuplicateLoadBalancerNameException":
                log.error(
                    f"A load balancer with the name '{load_balancer_name}'
already exists. "
                    "Load balancer names must be unique within the AWS region. "
                    "Please choose a different name and try again."
                )
            if error_code == "TooManyLoadBalancersException":
                log.error(
                    "The maximum number of load balancers has been reached in
this account and region. "
                    "You can delete unused load balancers or request an increase
in the service quota from AWS Support."
                )
            log.error(f"Full error:\n\t{err}")
        else:
            return load_balancer


    def create_listener(
        self,
        load_balancer_name: str,
        target_group: Dict[str, Any],
    ) -> Dict[str, Any]:
        """
        Creates a listener for the specified load balancer that forwards requests
to the
        specified target group.

        :param load_balancer_name: The name of the load balancer to create a
listener for.
        :param target_group: An existing target group that is added as a listener
to the
                             load balancer.
        :return: Data about the newly created listener.
        """
        try:
```

```
            # Retrieve the load balancer ARN
            load_balancer_response = self.elb_client.describe_load_balancers(
                Names=[load_balancer_name]
            )
            load_balancer_arn = load_balancer_response["LoadBalancers"][0][
                "LoadBalancerArn"
            ]

            # Create the listener
            response = self.elb_client.create_listener(
                LoadBalancerArn=load_balancer_arn,
                Protocol=target_group["Protocol"],
                Port=target_group["Port"],
                DefaultActions=[
                    {
                        "Type": "forward",
                        "TargetGroupArn": target_group["TargetGroupArn"],
                    }
                ],
            )
            log.info(
                f"Created listener to forward traffic from load balancer
'{load_balancer_name}' to target group '{target_group['TargetGroupName']}'."
            )
            return response["Listeners"][0]
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(
                f"Failed to add a listener on '{load_balancer_name}' for target
group '{target_group['TargetGroupName']}'."
            )

            if error_code == "ListenerNotFoundException":
                log.error(
                    f"The listener could not be found for the load balancer
'{load_balancer_name}'. "
                    "Please check the load balancer name and target group
configuration."
                )
            if error_code == "InvalidConfigurationRequestException":
                log.error(
                    f"The configuration provided for the listener on load
balancer '{load_balancer_name}' is invalid. "
```

```
                    "Please review the provided protocol, port, and target group
settings."
                )
            log.error(f"Full error:\n\t{err}")


    def delete_load_balancer(self, load_balancer_name) -> None:
        """
        Deletes a load balancer.

        :param load_balancer_name: The name of the load balancer to delete.
        """
        try:
            response = self.elb_client.describe_load_balancers(
                Names=[load_balancer_name]
            )
            lb_arn = response["LoadBalancers"][0]["LoadBalancerArn"]
            self.elb_client.delete_load_balancer(LoadBalancerArn=lb_arn)
            log.info("Deleted load balancer %s.", load_balancer_name)
            waiter = self.elb_client.get_waiter("load_balancers_deleted")
            log.info("Waiting for load balancer to be deleted...")
            waiter.wait(Names=[load_balancer_name])
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(
                f"Couldn't delete load balancer '{load_balancer_name}'. Error
code: {error_code}, Message: {err.response['Error']['Message']}"
            )

            if error_code == "LoadBalancerNotFoundException":
                log.error(
                    f"The load balancer '{load_balancer_name}' does not exist. "
                    "Please check the name and try again."
                )
            log.error(f"Full error:\n\t{err}")


    def get_endpoint(self, load_balancer_name) -> str:
        """
        Gets the HTTP endpoint of the load balancer.

        :return: The endpoint.
        """
        try:
```

```
            response = self.elb_client.describe_load_balancers(
                Names=[load_balancer_name]
            )
            return response["LoadBalancers"][0]["DNSName"]
        except ClientError as err:
            log.error(
                f"Couldn't get the endpoint for load balancer
{load_balancer_name}"
            )
            error_code = err.response["Error"]["Code"]
            if error_code == "LoadBalancerNotFoundException":
                log.error(
                    "Verify load balancer name and ensure it exists in the AWS
console."
                )
            log.error(f"Full error:\n\t{err}")

    @staticmethod
    def verify_load_balancer_endpoint(endpoint) -> bool:
        """
        Verify this computer can successfully send a GET request to the load
balancer endpoint.

        :param endpoint: The endpoint to verify.
        :return: True if the GET request is successful, False otherwise.
        """
        retries = 3
        verified = False
        while not verified and retries > 0:
            try:
                lb_response = requests.get(f"http://{endpoint}")
                log.info(
                    "Got response %s from load balancer endpoint.",
                    lb_response.status_code,
                )
                if lb_response.status_code == 200:
                    verified = True
                else:
                    retries = 0
            except requests.exceptions.ConnectionError:
                log.info(
                    "Got connection error from load balancer endpoint,
retrying..."
                )
```

```
                retries -= 1
                time.sleep(10)
        return verified

    def check_target_health(self, target_group_name: str) -> List[Dict[str,
Any]]:
        """
        Checks the health of the instances in the target group.

        :return: The health status of the target group.
        """
        try:
            tg_response = self.elb_client.describe_target_groups(
                Names=[target_group_name]
            )
            health_response = self.elb_client.describe_target_health(
                TargetGroupArn=tg_response["TargetGroups"][0]["TargetGroupArn"]
            )
        except ClientError as err:
            log.error(f"Couldn't check health of {target_group_name} target(s).")
            error_code = err.response["Error"]["Code"]
            if error_code == "LoadBalancerNotFoundException":
                log.error(
                    "Load balancer associated with the target group was not
found. "
                    "Ensure the load balancer exists, is in the correct AWS
region, and "
                    "that you have the necessary permissions to access it.",
                )
            elif error_code == "TargetGroupNotFoundException":
                log.error(
                    "Target group was not found. "
                    "Verify the target group name, check that it exists in the
correct region, "
                    "and ensure it has not been deleted or created in a different
account.",
                )
            log.error(f"Full error:\n\t{err}")
        else:
            return health_response["TargetHealthDescriptions"]
```

Crea una classe che utilizzi DynamoDB per simulare un servizio di raccomandazione.

```
class RecommendationService:
    """
    Encapsulates a DynamoDB table to use as a service that recommends books,
 movies,
    and songs.
    """

    def __init__(self, table_name: str, dynamodb_client: boto3.client):
        """
        Initializes the RecommendationService class with the necessary
 parameters.

        :param table_name: The name of the DynamoDB recommendations table.
        :param dynamodb_client: A Boto3 DynamoDB client.
        """
        self.table_name = table_name
        self.dynamodb_client = dynamodb_client

    def create(self) -> Dict[str, Any]:
        """
        Creates a DynamoDB table to use as a recommendation service. The table
 has a
        hash key named 'MediaType' that defines the type of media recommended,
 such as
        Book or Movie, and a range key named 'ItemId' that, combined with the
MediaType,
        forms a unique identifier for the recommended item.

        :return: Data about the newly created table.
        :raises RecommendationServiceError: If the table creation fails.
        """
        try:
            response = self.dynamodb_client.create_table(
                TableName=self.table_name,
                AttributeDefinitions=[
                    {"AttributeName": "MediaType", "AttributeType": "S"},
                    {"AttributeName": "ItemId", "AttributeType": "N"},
                ],
                KeySchema=[
```

```
                    {"AttributeName": "MediaType", "KeyType": "HASH"},
                    {"AttributeName": "ItemId", "KeyType": "RANGE"},
                ],
                ProvisionedThroughput={"ReadCapacityUnits": 5,
"WriteCapacityUnits": 5},
            )
            log.info("Creating table %s...", self.table_name)
            waiter = self.dynamodb_client.get_waiter("table_exists")
            waiter.wait(TableName=self.table_name)
            log.info("Table %s created.", self.table_name)
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceInUseException":
                log.info("Table %s exists, nothing to be done.", self.table_name)
            else:
                raise RecommendationServiceError(
                    self.table_name, f"ClientError when creating table: {err}."
                )
        else:
            return response

    def populate(self, data_file: str) -> None:
        """
        Populates the recommendations table from a JSON file.

        :param data_file: The path to the data file.
        :raises RecommendationServiceError: If the table population fails.
        """
        try:
            with open(data_file) as data:
                items = json.load(data)
            batch = [{"PutRequest": {"Item": item}} for item in items]
            self.dynamodb_client.batch_write_item(RequestItems={self.table_name:
batch})
            log.info(
                "Populated table %s with items from %s.", self.table_name,
data_file
            )
        except ClientError as err:
            raise RecommendationServiceError(
                self.table_name, f"Couldn't populate table from {data_file}:
{err}"
            )

    def destroy(self) -> None:
```

```
        """
        Deletes the recommendations table.

        :raises RecommendationServiceError: If the table deletion fails.
        """
        try:
            self.dynamodb_client.delete_table(TableName=self.table_name)
            log.info("Deleting table %s...", self.table_name)
            waiter = self.dynamodb_client.get_waiter("table_not_exists")
            waiter.wait(TableName=self.table_name)
            log.info("Table %s deleted.", self.table_name)
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                log.info("Table %s does not exist, nothing to do.",
 self.table_name)
            else:
                raise RecommendationServiceError(
                    self.table_name, f"ClientError when deleting table: {err}."
                )
```

Crea una classe che racchiuda le operazioni di Systems Manager.

```
class ParameterHelper:
    """
    Encapsulates Systems Manager parameters. This example uses these parameters
 to drive
    the demonstration of resilient architecture, such as failure of a dependency
 or
    how the service responds to a health check.
    """

    table: str = "doc-example-resilient-architecture-table"
    failure_response: str = "doc-example-resilient-architecture-failure-response"
    health_check: str = "doc-example-resilient-architecture-health-check"

    def __init__(self, table_name: str, ssm_client: boto3.client):
        """
        Initializes the ParameterHelper class with the necessary parameters.
```

```
        :param table_name: The name of the DynamoDB table that is used as a
recommendation
                           service.
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.table_name = table_name

    def reset(self) -> None:
        """
        Resets the Systems Manager parameters to starting values for the demo.
        These are the name of the DynamoDB recommendation table, no response when
a
        dependency fails, and shallow health checks.
        """
        self.put(self.table, self.table_name)
        self.put(self.failure_response, "none")
        self.put(self.health_check, "shallow")

    def put(self, name: str, value: str) -> None:
        """
        Sets the value of a named Systems Manager parameter.

        :param name: The name of the parameter.
        :param value: The new value of the parameter.
        :raises ParameterHelperError: If the parameter value cannot be set.
        """
        try:
            self.ssm_client.put_parameter(
                Name=name, Value=value, Overwrite=True, Type="String"
            )
            log.info("Setting parameter %s to '%s'.", name, value)
        except ClientError as err:
            error_code = err.response["Error"]["Code"]
            log.error(f"Failed to set parameter {name}.")
            if error_code == "ParameterLimitExceeded":
                log.error(
                    "The parameter limit has been exceeded. "
                    "Consider deleting unused parameters or request a limit
increase."
                )
            elif error_code == "ParameterAlreadyExists":
                log.error(
```

```
                    "The parameter already exists and overwrite is set to False.
"
                    "Use Overwrite=True to update the parameter."
                )
            log.error(f"Full error:\n\t{err}")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - AttachLoadBalancerTargetGroups
  - CreateAutoScalingGroup
  - CreateInstanceProfile
  - CreateLaunchTemplate
  - CreateListener
  - CreateLoadBalancer
  - CreateTargetGroup
  - DeleteAutoScalingGroup
  - DeleteInstanceProfile
  - DeleteLaunchTemplate
  - DeleteLoadBalancer
  - DeleteTargetGroup
  - DescribeAutoScalingGroups
  - DescribeAvailabilityZones
  - DescribeIamInstanceProfileAssociations
  - DescribeInstances
  - DescribeLoadBalancers
  - DescribeSubnets
  - DescribeTargetGroups
  - DescribeTargetHealth
  - DescribeVpcs
  - RebootInstances

- ReplaceIamInstanceProfileAssociation
- TerminateInstanceInAutoScalingGroup
- UpdateAutoScalingGroup

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Configurare la connettività dei servizi container

L'esempio di codice seguente mostra come:

- Crea l'infrastruttura VPC
- Configurare la registrazione
- Crea il cluster ECS
- Configura i ruoli IAM
- Crea il servizio con Service Connect
- Verifica della distribuzione
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# ECS Service Connect Tutorial Script v4 - Modified to use Default VPC
# This script creates an ECS cluster with Service Connect and deploys an nginx
 service
# Uses the default VPC to avoid VPC limits
```

```
set -e  # Exit on any error

# Configuration
SCRIPT_NAME="ECS Service Connect Tutorial"
LOG_FILE="ecs-service-connect-tutorial-v4-default-vpc.log"
REGION=${AWS_DEFAULT_REGION:-${AWS_REGION:-$(aws configure get region 2>/dev/
null)}}
if [ -z "$REGION" ]; then
    echo "ERROR: No AWS region configured."
    echo "Set one with: aws configure set region us-east-1"
    exit 1
fi
ENV_PREFIX="tutorial"
CLUSTER_NAME="${ENV_PREFIX}-cluster"
NAMESPACE_NAME="service-connect"

# Generate random suffix for unique resource names
RANDOM_SUFFIX=$(openssl rand -hex 6)

# Arrays to track created resources for cleanup
declare -a CREATED_RESOURCES=()

# Logging function
log() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | tee -a "$LOG_FILE"
}

# Error handling function
handle_error() {
    log "ERROR: Script failed at line $1"
    log "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Set up error handling
trap 'handle_error $LINENO' ERR

# Function to add resource to tracking array
track_resource() {
    CREATED_RESOURCES+=("$1")
    log "Tracking resource: $1"
}
```

```
# Function to check if command output contains actual errors
check_for_errors() {
    local output="$1"
    local command_name="$2"

    # Check for specific AWS CLI error patterns, not just any occurrence of
 "error"
    if echo "$output" | grep -qi "An error occurred\|InvalidParameterException\|
AccessDenied\|ValidationException\|ResourceNotFoundException"; then
        log "ERROR in $command_name: $output"
        return 1
    fi
    return 0
}

# Function to get AWS account ID
get_account_id() {
    ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
    log "Using AWS Account ID: $ACCOUNT_ID"
}

# Function to wait for resources to be ready
wait_for_resource() {
    local resource_type="$1"
    local resource_id="$2"

    case "$resource_type" in
        "cluster")
            log "Waiting for cluster $resource_id to be active..."
            local attempt=1
            local max_attempts=30
            while [ $attempt -le $max_attempts ]; do
                local status=$(aws ecs describe-clusters --clusters
 "$resource_id" --query 'clusters[0].status' --output text)
                if [ "$status" = "ACTIVE" ]; then
                    log "Cluster is now active"
                    return 0
                fi
                log "Cluster status: $status (attempt $attempt/$max_attempts)"
                sleep 10
                ((attempt++))
            done
            log "ERROR: Cluster did not become active within expected time"
```

```
            return 1
            ;;
        "service")
            log "Waiting for service $resource_id to be stable..."
            aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$resource_id"
            ;;
        "nat-gateway")
            log "Waiting for NAT Gateway $resource_id to be available..."
            aws ec2 wait nat-gateway-available --nat-gateway-ids "$resource_id"
            ;;
    esac
}

# Function to use default VPC infrastructure
setup_default_vpc_infrastructure() {
    log "Using default VPC infrastructure..."

    # Get default VPC
    VPC_ID=$(aws ec2 describe-vpcs --filters "Name=isDefault,Values=true" --query
 'Vpcs[0].VpcId' --output text)
    if [[ "$VPC_ID" == "None" || -z "$VPC_ID" ]]; then
        log "ERROR: No default VPC found. Please create a default VPC first."
        exit 1
    fi
    log "Using default VPC: $VPC_ID"

    # Get default subnets
    SUBNETS=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC_ID"
 "Name=default-for-az,Values=true" --query 'Subnets[].SubnetId' --output text)
    SUBNET_ARRAY=($SUBNETS)

    if [ ${#SUBNET_ARRAY[@]} -lt 2 ]; then
        log "ERROR: Need at least 2 subnets for ECS Service Connect. Found:
 ${#SUBNET_ARRAY[@]}"
        exit 1
    fi

    PUBLIC_SUBNET1=${SUBNET_ARRAY[0]}
    PUBLIC_SUBNET2=${SUBNET_ARRAY[1]}

    log "Using subnets: $PUBLIC_SUBNET1, $PUBLIC_SUBNET2"

    # Create security group for ECS tasks
```

```
    SG_OUTPUT=$(aws ec2 create-security-group \
        --group-name "${ENV_PREFIX}-ecs-sg-${RANDOM_SUFFIX}" \
        --description "Security group for ECS Service Connect tutorial" \
        --vpc-id "$VPC_ID" \
        --tag-specifications 'ResourceType=security-
group,Tags=[{Key=project,Value=doc-smith},{Key=tutorial,Value=amazon-ecs-service-
connect}]' 2>&1)
    check_for_errors "$SG_OUTPUT" "create-security-group"
    SECURITY_GROUP_ID=$(echo "$SG_OUTPUT" | grep -o '"GroupId": "[^"]*"' | cut -
d'"' -f4)
    track_resource "SG:$SECURITY_GROUP_ID"
    log "Created security group: $SECURITY_GROUP_ID"

    # Add inbound rules to security group
    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 80 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true

    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 443 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true

    log "Default VPC infrastructure setup completed"
}

# Function to create CloudWatch log groups
create_log_groups() {
    log "Creating CloudWatch log groups..."

    # Create log group for nginx container
    aws logs create-log-group --log-group-name "/ecs/service-connect-nginx"
 --tags project=doc-smith,tutorial=amazon-ecs-service-connect 2>&1 | grep -v
 "ResourceAlreadyExistsException" || {
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-nginx created"
            track_resource "LOG_GROUP:/ecs/service-connect-nginx"
        else
            log "Log group /ecs/service-connect-nginx already exists"
        fi
    }
```

```
    # Create log group for service connect proxy
    aws logs create-log-group --log-group-name "/ecs/service-connect-proxy"
 --tags project=doc-smith,tutorial=amazon-ecs-service-connect 2>&1 | grep -v
 "ResourceAlreadyExistsException" || {
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-proxy created"
            track_resource "LOG_GROUP:/ecs/service-connect-proxy"
        else
            log "Log group /ecs/service-connect-proxy already exists"
        fi
    }
}

# Function to create ECS cluster with Service Connect
create_ecs_cluster() {
    log "Creating ECS cluster with Service Connect..."

    CLUSTER_OUTPUT=$(aws ecs create-cluster \
        --cluster-name "$CLUSTER_NAME" \
        --service-connect-defaults namespace="$NAMESPACE_NAME" \
        --tags key=Environment,value=tutorial key=project,value=doc-smith
 key=tutorial,value=amazon-ecs-service-connect 2>&1)
    check_for_errors "$CLUSTER_OUTPUT" "create-cluster"

    track_resource "CLUSTER:$CLUSTER_NAME"
    log "Created ECS cluster: $CLUSTER_NAME"

    wait_for_resource "cluster" "$CLUSTER_NAME"

    # Track the Service Connect namespace that gets created
    # Wait a moment for the namespace to be created
    sleep 5
    NAMESPACE_ID=$(aws servicediscovery list-namespaces \
        --filters Name=TYPE,Values=HTTP \
        --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")

    if [[ -n "$NAMESPACE_ID" && "$NAMESPACE_ID" != "None" ]]; then
        track_resource "NAMESPACE:$NAMESPACE_ID"
        log "Service Connect namespace created: $NAMESPACE_ID"
    fi
}
```

```
# Function to create IAM roles
create_iam_roles() {
    log "Creating IAM roles..."

    # Check if ecsTaskExecutionRole exists
    if aws iam get-role --role-name ecsTaskExecutionRole >/dev/null 2>&1; then
        log "IAM role ecsTaskExecutionRole exists"
    else
        log "Creating ecsTaskExecutionRole..."
        aws iam create-role \
            --role-name ecsTaskExecutionRole \
            --assume-role-policy-document '{
                "Version":"2012-10-17",
                "Statement": [{
                    "Effect": "Allow",
                    "Principal": {"Service": "ecs-tasks.amazonaws.com"},
                    "Action": "sts:AssumeRole"
                }]
            }' >/dev/null 2>&1
        aws iam attach-role-policy \
            --role-name ecsTaskExecutionRole \
            --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy >/dev/null 2>&1
        track_resource "ROLE:ecsTaskExecutionRole"
        aws iam tag-role --role-name ecsTaskExecutionRole --tags
 Key=project,Value=doc-smith Key=tutorial,Value=amazon-ecs-service-connect
        log "Created ecsTaskExecutionRole"
        sleep 10
    fi

    # Check if ecsTaskRole exists, create if not
    if aws iam get-role --role-name ecsTaskRole >/dev/null 2>&1; then
        log "IAM role ecsTaskRole exists"
    else
        log "IAM role ecsTaskRole does not exist, will create it"

        # Create trust policy for ECS tasks
        cat > /tmp/ecs-task-trust-policy.json << EOF
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "ecs-tasks.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF

        aws iam create-role \
            --role-name ecsTaskRole \
            --assume-role-policy-document file:///tmp/ecs-task-trust-policy.json
 >/dev/null

        track_resource "IAM_ROLE:ecsTaskRole"
        aws iam tag-role --role-name ecsTaskRole --tags Key=project,Value=doc-
smith Key=tutorial,Value=amazon-ecs-service-connect
        log "Created ecsTaskRole"

        # Wait for role to be available
        sleep 10
    fi
}

# Function to create task definition
create_task_definition() {
    log "Creating task definition..."

    # Create task definition JSON
    cat > /tmp/task-definition.json << EOF
{
    "family": "service-connect-nginx",
    "networkMode": "awsvpc",
    "requiresCompatibilities": ["FARGATE"],
    "cpu": "256",
    "memory": "512",
    "executionRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskExecutionRole",
    "taskRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskRole",
    "containerDefinitions": [
        {
            "name": "nginx",
            "image": "public.ecr.aws/docker/library/nginx:latest",
            "portMappings": [
                {
                    "containerPort": 80,
```

```
                    "protocol": "tcp",
                    "name": "nginx-port"
                }
            ],
            "essential": true,
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/service-connect-nginx",
                    "awslogs-region": "${REGION}",
                    "awslogs-stream-prefix": "ecs"
                }
            }
        }
    ]
}
EOF

    TASK_DEF_OUTPUT=$(aws ecs register-task-definition --cli-input-json file:///
tmp/task-definition.json 2>&1)
    check_for_errors "$TASK_DEF_OUTPUT" "register-task-definition"

    TASK_DEF_ARN=$(echo "$TASK_DEF_OUTPUT" | grep -o '"taskDefinitionArn":
 "[^"]*"' | cut -d'"' -f4)
    track_resource "TASK_DEF:service-connect-nginx"
    log "Created task definition: $TASK_DEF_ARN"

    # Clean up temporary file
    rm -f /tmp/task-definition.json
}

# Function to create ECS service with Service Connect
create_ecs_service() {
    log "Creating ECS service with Service Connect..."

    # Create service definition JSON
    cat > /tmp/service-definition.json << EOF
{
    "serviceName": "service-connect-nginx-service",
    "cluster": "${CLUSTER_NAME}",
    "taskDefinition": "service-connect-nginx",
    "desiredCount": 1,
    "launchType": "FARGATE",
    "networkConfiguration": {
```

```
        "awsvpcConfiguration": {
            "subnets": ["${PUBLIC_SUBNET1}", "${PUBLIC_SUBNET2}"],
            "securityGroups": ["${SECURITY_GROUP_ID}"],
            "assignPublicIp": "ENABLED"
        }
    },
    "serviceConnectConfiguration": {
        "enabled": true,
        "namespace": "${NAMESPACE_NAME}",
        "services": [
            {
                "portName": "nginx-port",
                "discoveryName": "nginx",
                "clientAliases": [
                    {
                        "port": 80,
                        "dnsName": "nginx"
                    }
                ]
            }
        ],
        "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
                "awslogs-group": "/ecs/service-connect-proxy",
                "awslogs-region": "${REGION}",
                "awslogs-stream-prefix": "ecs-service-connect"
            }
        }
    },
    "tags": [
        {
            "key": "Environment",
            "value": "tutorial"
        },
        {
            "key": "project",
            "value": "doc-smith"
        },
        {
            "key": "tutorial",
            "value": "amazon-ecs-service-connect"
        }
    ]
```

```
}
EOF

    SERVICE_OUTPUT=$(aws ecs create-service --cli-input-json file:///tmp/service-
definition.json 2>&1)
    check_for_errors "$SERVICE_OUTPUT" "create-service"

    track_resource "SERVICE:service-connect-nginx-service"
    log "Created ECS service: service-connect-nginx-service"

    wait_for_resource "service" "service-connect-nginx-service"

    # Clean up temporary file
    rm -f /tmp/service-definition.json
}

# Function to verify deployment
verify_deployment() {
    log "Verifying deployment..."

    # Check service status
    SERVICE_STATUS=$(aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].status' --output text)
    log "Service status: $SERVICE_STATUS"

    # Check running tasks
    RUNNING_COUNT=$(aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].runningCount' --output text)
    log "Running tasks: $RUNNING_COUNT"

    # Get task ARN
    TASK_ARN=$(aws ecs list-tasks \
        --cluster "$CLUSTER_NAME" \
        --service-name "service-connect-nginx-service" \
        --query 'taskArns[0]' --output text)

    if [[ "$TASK_ARN" != "None" && -n "$TASK_ARN" ]]; then
        log "Task ARN: $TASK_ARN"

        # Try to get task IP address
```

```
        TASK_IP=$(aws ecs describe-tasks \
            --cluster "$CLUSTER_NAME" \
            --tasks "$TASK_ARN" \
            --query 'tasks[0].attachments[0].details[?
name==`privateIPv4Address`].value' \
            --output text 2>/dev/null || echo "")

        if [[ -n "$TASK_IP" && "$TASK_IP" != "None" ]]; then
            log "Task IP address: $TASK_IP"
        else
            log "Could not retrieve task IP address"
        fi
    fi

    # Check Service Connect namespace
    NAMESPACE_STATUS=$(aws servicediscovery list-namespaces \
        --filters Name=TYPE,Values=HTTP \
        --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")

    if [[ -n "$NAMESPACE_STATUS" && "$NAMESPACE_STATUS" != "None" ]]; then
        log "Service Connect namespace '$NAMESPACE_NAME' is active"
    else
        log "Service Connect namespace '$NAMESPACE_NAME' not found or not active"
    fi

    # Display Service Connect configuration
    log "Service Connect configuration:"
    aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].serviceConnectConfiguration' 2>/dev/null || true
}

# Function to display created resources
display_resources() {
    echo ""
    echo "==========================================="
    echo "CREATED RESOURCES"
    echo "==========================================="
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "- $resource"
    done
    echo "==========================================="
```

```
    echo ""
}

# Function to cleanup resources
cleanup_resources() {
    log "Starting cleanup process..."

    # Delete resources in reverse order of creation
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        resource="${CREATED_RESOURCES[i]}"
        resource_type=$(echo "$resource" | cut -d':' -f1)
        resource_id=$(echo "$resource" | cut -d':' -f2)

        log "Cleaning up $resource_type: $resource_id"

        case "$resource_type" in
            "SERVICE")
                aws ecs update-service --cluster "$CLUSTER_NAME" --service
 "$resource_id" --desired-count 0 2>&1 | grep -qi "error" && log "Warning: Failed
 to scale down service $resource_id"
                aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$resource_id" 2>/dev/null || true
                aws ecs delete-service --cluster "$CLUSTER_NAME" --service
 "$resource_id" --force 2>&1 | grep -qi "error" && log "Warning: Failed to delete
 service $resource_id"
                ;;
            "TASK_DEF")
                TASK_DEF_ARNS=$(aws ecs list-task-definitions --family-prefix
 "$resource_id" --query 'taskDefinitionArns' --output text 2>/dev/null)
                for arn in $TASK_DEF_ARNS; do
                    aws ecs deregister-task-definition --task-definition "$arn"
 >/dev/null 2>&1 || true
                done
                ;;
            "ROLE")
                aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
                aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete role $resource_id"
                ;;
            "IAM_ROLE")
```

```
                aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
                aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete role $resource_id"
                ;;
            "CLUSTER")
                aws ecs delete-cluster --cluster "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete cluster $resource_id"
                ;;
            "SG")
                for attempt in 1 2 3 4 5; do
                    if aws ec2 delete-security-group --group-id "$resource_id"
 2>/dev/null; then
                        break
                    fi
                    log "Security group $resource_id still has dependencies,
 retrying in 30s ($attempt/5)..."
                    sleep 30
                done
                ;;
            "LOG_GROUP")
                aws logs delete-log-group --log-group-name "$resource_id" 2>&1 |
 grep -qi "error" && log "Warning: Failed to delete log group $resource_id"
                ;;
            "NAMESPACE")
                # First, delete any services in the namespace
                NAMESPACE_SERVICES=$(aws servicediscovery list-services \
                    --filters Name=NAMESPACE_ID,Values="$resource_id" \
                    --query 'Services[].Id' --output text 2>/dev/null || echo "")

                if [[ -n "$NAMESPACE_SERVICES" && "$NAMESPACE_SERVICES" !=
 "None" ]]; then
                    for service_id in $NAMESPACE_SERVICES; do
                        aws servicediscovery delete-service --id "$service_id" >/
dev/null 2>&1 || true
                        sleep 2
                    done
                fi

                # Then delete the namespace
                aws servicediscovery delete-namespace --id "$resource_id" >/dev/
null 2>&1 || true
                ;;
```

```
        esac

        sleep 2  # Brief pause between deletions
    done

    # Clean up temporary files
    rm -f /tmp/ecs-task-trust-policy.json
    rm -f /tmp/task-definition.json
    rm -f /tmp/service-definition.json

    log "Cleanup completed"
}

# Main execution
main() {
    log "Starting $SCRIPT_NAME v4 (Default VPC)"
    log "Region: $REGION"
    log "Log file: $LOG_FILE"

    # Get AWS account ID
    get_account_id

    # Setup infrastructure using default VPC
    setup_default_vpc_infrastructure

    # Create CloudWatch log groups
    create_log_groups

    # Create ECS cluster
    create_ecs_cluster

    # Create IAM roles
    create_iam_roles

    # Create task definition
    create_task_definition

    # Create ECS service
    create_ecs_service

    # Verify deployment
    verify_deployment

    log "Tutorial completed successfully!"
```

```
    # Display created resources
    display_resources

    # Ask user if they want to clean up
    echo ""
    echo "========================================="
    echo "CLEANUP CONFIRMATION"
    echo "========================================="
    echo "Do you want to clean up all created resources? (y/n): "
    CLEANUP_CHOICE="y"

    if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
        cleanup_resources
        log "All resources have been cleaned up"
    else
        log "Resources left intact. You can clean them up later by running the
 cleanup function."
        echo ""
        echo "To clean up resources later, you can use the AWS CLI commands or
 the AWS Management Console."
        echo "Remember to delete resources in the correct order to avoid
 dependency issues."
    fi
}

# Make script executable and run
chmod +x "$0"
main "$@"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress
  - CreateCluster
  - CreateLogGroup
  - CreateRole
  - CreateSecurityGroup
  - CreateService

- DeleteCluster
- DeleteLogGroup
- DeleteNamespace
- DeleteRole
- DeleteSecurityGroup
- DeleteService (AWS Cloud Map)
- DeleteService (Amazon ECS)
- DeregisterTaskDefinition
- DescribeClusters
- DescribeServices
- DescribeSubnets
- DescribeTasks
- DescribeVpcs
- DetachRolePolicy
- GetCallerIdentity
- GetRole
- ListNamespaces
- ListServices
- ListTaskDefinitions
- ListTasks
- RegisterTaskDefinition
- UpdateService
- Attendi (Amazon EC2)
- Attendi (Amazon ECS)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea un'attività contenitore per il tipo di avvio serverless

L'esempio di codice seguente mostra come:

- Creazione del cluster
- Creazione di una definizione di attività
- Crea il servizio
- Eliminazione

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon ECS Fargate Tutorial Script - Version 5
# This script creates an ECS cluster, task definition, and service using Fargate
 launch type
# Fixed version with proper resource dependency handling during cleanup
# Security improvements applied

set -e  # Exit on any error

# Initialize logging
LOG_FILE="ecs-fargate-tutorial-v5.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon ECS Fargate tutorial at $(date)"
echo "Log file: $LOG_FILE"

# Generate random identifier for unique resource names
RANDOM_ID=$(openssl rand -hex 6)
CLUSTER_NAME="fargate-cluster-${RANDOM_ID}"
SERVICE_NAME="fargate-service-${RANDOM_ID}"
TASK_FAMILY="sample-fargate-${RANDOM_ID}"
SECURITY_GROUP_NAME="ecs-fargate-sg-${RANDOM_ID}"
```

```
# Array to track created resources for cleanup
CREATED_RESOURCES=()

# Function to log and execute commands with input validation
execute_command() {
    local cmd="$1"
    local description="$2"

    # Validate that cmd is not empty
    if [[ -z "$cmd" ]]; then
        echo "ERROR: Command is empty"
        return 1
    fi

    echo ""
    echo "========================================="
    echo "EXECUTING: $description"
    echo "COMMAND: $cmd"
    echo "========================================="

    local output
    local exit_code
    set +e  # Temporarily disable exit on error
    output=$(eval "$cmd" 2>&1)
    exit_code=$?
    set -e  # Re-enable exit on error

    if [[ $exit_code -eq 0 ]]; then
        echo "SUCCESS: $description"
        echo "OUTPUT: $output"
        return 0
    else
        echo "FAILED: $description"
        echo "EXIT CODE: $exit_code"
        echo "OUTPUT: $output"
        return 1
    fi
}

# Function to check for actual AWS API errors in command output
check_for_aws_errors() {
    local output="$1"
    local description="$2"
```

```
    # Look for specific AWS error patterns, not just the word "error"
    if echo "$output" | grep -qi "An error occurred\|InvalidParameter\|
AccessDenied\|ResourceNotFound\|ValidationException"; then
        echo "AWS API ERROR detected in output for: $description"
        echo "Output: $output"
        return 1
    fi
    return 0
}

# Function to safely extract JSON values
safe_json_extract() {
    local json="$1"
    local key="$2"
    local value

    value=$(echo "$json" | grep -o "\"$key\": \"[^\"]*\"" | cut -d'"' -f4 2>/dev/
null || echo "")
    echo "$value"
}

# Function to wait for network interfaces to be cleaned up
wait_for_network_interfaces_cleanup() {
    local security_group_id="$1"
    local max_attempts=30
    local attempt=1

    # Validate security group ID format
    if [[ ! "$security_group_id" =~ ^sg-[a-z0-9]{8,17}$ ]]; then
        echo "ERROR: Invalid security group ID format: $security_group_id"
        return 1
    fi

    echo "Waiting for network interfaces to be cleaned up..."

    while [[ $attempt -le $max_attempts ]]; do
        echo "Attempt $attempt/$max_attempts: Checking for dependent network
 interfaces..."

        # Check if there are any network interfaces still using this security
 group
        local eni_count
        eni_count=$(aws ec2 describe-network-interfaces \
            --filters "Name=group-id,Values=$security_group_id" \
```

```
            --query "length(NetworkInterfaces)" \
            --output text 2>/dev/null || echo "0")

        if [[ "$eni_count" == "0" ]]; then
            echo "No network interfaces found using security group
 $security_group_id"
            return 0
        else
            echo "Found $eni_count network interface(s) still using security
 group $security_group_id"
            echo "Waiting 10 seconds before next check..."
            sleep 10
            ((attempt++))
        fi
    done

    echo "WARNING: Network interfaces may still be attached after $max_attempts
 attempts"
    echo "This is normal and the security group deletion will be retried"
    return 1
}

# Function to retry security group deletion with exponential backoff
retry_security_group_deletion() {
    local security_group_id="$1"
    local max_attempts=10
    local attempt=1
    local wait_time=5

    # Validate security group ID format
    if [[ ! "$security_group_id" =~ ^sg-[a-z0-9]{8,17}$ ]]; then
        echo "ERROR: Invalid security group ID format: $security_group_id"
        return 1
    fi

    while [[ $attempt -le $max_attempts ]]; do
        echo "Attempt $attempt/$max_attempts: Trying to delete security group
 $security_group_id"

        if execute_command "aws ec2 delete-security-group --group-id
 '$security_group_id'" "Delete security group (attempt $attempt)"; then
            echo "Successfully deleted security group $security_group_id"
            return 0
        else
```

```
            if [[ $attempt -eq $max_attempts ]]; then
                echo "FAILED: Could not delete security group $security_group_id
 after $max_attempts attempts"
                echo "This may be due to network interfaces that are still being
 cleaned up by AWS"
                echo "You can manually delete it later using: aws ec2 delete-
security-group --group-id $security_group_id"
                return 1
            else
                echo "Waiting $wait_time seconds before retry..."
                sleep $wait_time
                wait_time=$((wait_time * 2))  # Exponential backoff
                ((attempt++))
            fi
        fi
    done
}

# Function to cleanup resources with proper dependency handling
cleanup_resources() {
    echo ""
    echo "==========================================="
    echo "CLEANUP PROCESS"
    echo "==========================================="
    echo "The following resources were created:"
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "  - $resource"
    done
    echo ""
    echo "Auto-confirming cleanup of all created resources..."

    CLEANUP_CHOICE="y"

    if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
        echo "Starting cleanup process..."

        # Step 1: Scale service to 0 tasks first, then delete service
        if [[ " ${CREATED_RESOURCES[*]} " =~ " ECS Service: $SERVICE_NAME " ]];
 then
            echo ""
            echo "Step 1: Scaling service to 0 tasks..."
            if execute_command "aws ecs update-service --cluster '$CLUSTER_NAME'
 --service '$SERVICE_NAME' --desired-count 0" "Scale service to 0 tasks"; then
                echo "Waiting for service to stabilize after scaling to 0..."
```

```
                execute_command "aws ecs wait services-stable --cluster
'$CLUSTER_NAME' --services '$SERVICE_NAME'" "Wait for service to stabilize"

                echo "Deleting service..."
                execute_command "aws ecs delete-service --cluster '$CLUSTER_NAME'
--service '$SERVICE_NAME'" "Delete ECS service"
            else
                echo "WARNING: Failed to scale service. Attempting to delete
anyway..."
                execute_command "aws ecs delete-service --cluster '$CLUSTER_NAME'
--service '$SERVICE_NAME' --force" "Force delete ECS service"
            fi
        fi

        # Step 2: Wait a bit for tasks to fully terminate
        echo ""
        echo "Step 2: Waiting for tasks to fully terminate..."
        sleep 15

        # Step 3: Delete cluster
        if [[ " ${CREATED_RESOURCES[*]} " =~ " ECS Cluster: $CLUSTER_NAME " ]];
then
            echo ""
            echo "Step 3: Deleting cluster..."
            execute_command "aws ecs delete-cluster --cluster '$CLUSTER_NAME'"
"Delete ECS cluster"
        fi

        # Step 4: Wait for network interfaces to be cleaned up, then delete
security group
        if [[ -n "$SECURITY_GROUP_ID" && "$SECURITY_GROUP_ID" != "None" ]]; then
            echo ""
            echo "Step 4: Cleaning up security group..."

            # First, wait for network interfaces to be cleaned up
            wait_for_network_interfaces_cleanup "$SECURITY_GROUP_ID"

            # Then retry security group deletion with backoff
            retry_security_group_deletion "$SECURITY_GROUP_ID"
        fi

        # Step 5: Clean up task definition (deregister all revisions)
        if [[ " ${CREATED_RESOURCES[*]} " =~ " Task Definition: $TASK_FAMILY
" ]]; then
```

```
            echo ""
            echo "Step 5: Deregistering task definition revisions..."

            # Get all revisions of the task definition
            local revisions
            revisions=$(aws ecs list-task-definitions --family-prefix
 "$TASK_FAMILY" --query "taskDefinitionArns" --output text 2>/dev/null || echo
 "")

            if [[ -n "$revisions" && "$revisions" != "None" ]]; then
                for revision_arn in $revisions; do
                    echo "Deregistering task definition: $revision_arn"
                    execute_command "aws ecs deregister-task-definition --task-
definition '$revision_arn'" "Deregister task definition $revision_arn" || true
                done
            else
                echo "No task definition revisions found to deregister"
            fi
        fi

        echo ""
        echo "============================================"
        echo "CLEANUP COMPLETED"
        echo "============================================"
        echo "All resources have been cleaned up successfully!"

    else
        echo "Cleanup skipped. Resources remain active."
        echo ""
        echo "To clean up manually later, use the following commands in order:"
        echo "1. Scale service to 0: aws ecs update-service --cluster
 $CLUSTER_NAME --service $SERVICE_NAME --desired-count 0"
        echo "2. Wait for stability: aws ecs wait services-stable --cluster
 $CLUSTER_NAME --services $SERVICE_NAME"
        echo "3. Delete service: aws ecs delete-service --cluster $CLUSTER_NAME
 --service $SERVICE_NAME"
        echo "4. Delete cluster: aws ecs delete-cluster --cluster $CLUSTER_NAME"
        echo "5. Wait 2-3 minutes, then delete security group: aws ec2 delete-
security-group --group-id $SECURITY_GROUP_ID"
        if [[ " ${CREATED_RESOURCES[*]} " =~ " Task Definition: $TASK_FAMILY
 " ]]; then
            echo "6. Deregister task definitions: aws ecs list-task-definitions
 --family-prefix $TASK_FAMILY"
```

```
            echo "   Then for each ARN: aws ecs deregister-task-definition --
task-definition <ARN>"
        fi
    fi
}

# Trap to handle script interruption
trap cleanup_resources EXIT

echo "Using random identifier: $RANDOM_ID"
echo "Cluster name: $CLUSTER_NAME"
echo "Service name: $SERVICE_NAME"
echo "Task family: $TASK_FAMILY"

# Step 1: Ensure ECS task execution role exists
echo ""
echo "============================================="
echo "STEP 1: VERIFY ECS TASK EXECUTION ROLE"
echo "============================================="

ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
if [[ ! "$ACCOUNT_ID" =~ ^[0-9]{12}$ ]]; then
    echo "ERROR: Invalid AWS Account ID retrieved: $ACCOUNT_ID"
    exit 1
fi

EXECUTION_ROLE_ARN="arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskExecutionRole"

# Check if role exists
if aws iam get-role --role-name ecsTaskExecutionRole >/dev/null 2>&1; then
    echo "ECS task execution role already exists"
else
    echo "Creating ECS task execution role..."

    # Create trust policy with strict validation
    cat > trust-policy.json << 'EOF'
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            },
```

```
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF

    # Validate JSON before using
    if ! jq empty trust-policy.json 2>/dev/null; then
        echo "ERROR: Invalid JSON in trust policy"
        rm -f trust-policy.json
        exit 1
    fi

    execute_command "aws iam create-role --role-name ecsTaskExecutionRole --
assume-role-policy-document file://trust-policy.json" "Create ECS task execution
 role"

    aws iam tag-role --role-name ecsTaskExecutionRole --tags
 Key=project,Value=doc-smith Key=tutorial,Value=amazon-ecs-fargate-linux

    execute_command "aws iam attach-role-policy --role-name
 ecsTaskExecutionRole --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy" "Attach ECS task execution policy"

    # Clean up temporary file securely
    shred -vfz -n 3 trust-policy.json 2>/dev/null || rm -f trust-policy.json

    CREATED_RESOURCES+=("IAM Role: ecsTaskExecutionRole")
fi

# Step 2: Create ECS cluster
echo ""
echo "============================================="
echo "STEP 2: CREATE ECS CLUSTER"
echo "============================================="

CLUSTER_OUTPUT=$(execute_command "aws ecs create-cluster --cluster-name
 '$CLUSTER_NAME' --tags key=project,value=doc-smith key=tutorial,value=amazon-
ecs-fargate-linux" "Create ECS cluster")
check_for_aws_errors "$CLUSTER_OUTPUT" "Create ECS cluster"

CREATED_RESOURCES+=("ECS Cluster: $CLUSTER_NAME")

# Step 3: Create task definition
```

```
echo ""
echo "=========================================="
echo "STEP 3: CREATE TASK DEFINITION"
echo "=========================================="

# Create task definition JSON with validated inputs
cat > task-definition.json << EOF
{
    "family": "$TASK_FAMILY",
    "networkMode": "awsvpc",
    "requiresCompatibilities": ["FARGATE"],
    "cpu": "256",
    "memory": "512",
    "executionRoleArn": "$EXECUTION_ROLE_ARN",
    "containerDefinitions": [
        {
            "name": "fargate-app",
            "image": "public.ecr.aws/docker/library/httpd:2.4-alpine",
            "portMappings": [
                {
                    "containerPort": 80,
                    "hostPort": 80,
                    "protocol": "tcp"
                }
            ],
            "essential": true,
            "entryPoint": ["sh", "-c"],
            "command": [
                "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
 App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </
head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</
h1> <h2>Congratulations!</h2> <p>Your application is now running on a container
 in Amazon ECS.</p> </div></body></html>' >  /usr/local/apache2/htdocs/index.html
 && httpd-foreground\""
            ],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/fargate-sample",
                    "awslogs-region": "us-east-1",
                    "awslogs-stream-prefix": "ecs"
                }
            }
        }
```

```
    ]
}
EOF

# Validate JSON before using
if ! jq empty task-definition.json 2>/dev/null; then
    echo "ERROR: Invalid JSON in task definition"
    rm -f task-definition.json
    exit 1
fi

TASK_DEF_OUTPUT=$(execute_command "aws ecs register-task-definition --cli-input-
json file://task-definition.json" "Register task definition")
check_for_aws_errors "$TASK_DEF_OUTPUT" "Register task definition"

# Clean up temporary file securely
shred -vfz -n 3 task-definition.json 2>/dev/null || rm -f task-definition.json

CREATED_RESOURCES+=("Task Definition: $TASK_FAMILY")

# Step 4: Set up networking
echo ""
echo "==========================================="
echo "STEP 4: SET UP NETWORKING"
echo "==========================================="

# Get default VPC ID
VPC_ID=$(aws ec2 describe-vpcs --filters "Name=is-default,Values=true" --query
 "Vpcs[0].VpcId" --output text)
if [[ "$VPC_ID" == "None" || -z "$VPC_ID" ]]; then
    echo "ERROR: No default VPC found. Please create a default VPC or specify a
 custom VPC."
    exit 1
fi

# Validate VPC ID format
if [[ ! "$VPC_ID" =~ ^vpc-[a-z0-9]{8,17}$ ]]; then
    echo "ERROR: Invalid VPC ID format: $VPC_ID"
    exit 1
fi

echo "Using default VPC: $VPC_ID"

# Create security group with restricted access
```

```
# Note: This allows HTTP access from anywhere for demo purposes
# In production, restrict source to specific IP ranges or security groups
SECURITY_GROUP_OUTPUT=$(execute_command "aws ec2 create-security-group --
group-name '$SECURITY_GROUP_NAME' --description 'Security group for ECS
 Fargate tutorial - HTTP access' --vpc-id '$VPC_ID' --tag-specifications
 'ResourceType=security-group,Tags=[{Key=project,Value=doc-smith},
{Key=tutorial,Value=amazon-ecs-fargate-linux}]'" "Create security group")
check_for_aws_errors "$SECURITY_GROUP_OUTPUT" "Create security group"

SECURITY_GROUP_ID=$(echo "$SECURITY_GROUP_OUTPUT" | grep -o '"GroupId": "[^"]*"'
 | head -1 | cut -d'"' -f4)
if [[ -z "$SECURITY_GROUP_ID" || "$SECURITY_GROUP_ID" == "None" ]]; then
    SECURITY_GROUP_ID=$(aws ec2 describe-security-groups --group-names
 "$SECURITY_GROUP_NAME" --filters "Name=vpc-id,Values=$VPC_ID" --query
 "SecurityGroups[0].GroupId" --output text)
fi

# Validate security group ID format
if [[ ! "$SECURITY_GROUP_ID" =~ ^sg-[a-z0-9]{8,17}$ ]]; then
    echo "ERROR: Invalid security group ID format: $SECURITY_GROUP_ID"
    exit 1
fi

echo "Created security group: $SECURITY_GROUP_ID"
CREATED_RESOURCES+=("Security Group: $SECURITY_GROUP_ID")

# Add HTTP inbound rule
# WARNING: This allows HTTP access from anywhere (0.0.0.0/0)
# In production environments, restrict this to specific IP ranges
execute_command "aws ec2 authorize-security-group-ingress --group-id
 '$SECURITY_GROUP_ID' --protocol tcp --port 80 --cidr 0.0.0.0/0" "Add HTTP
 inbound rule to security group"

# Get subnet IDs from default VPC
echo "Getting subnet IDs from default VPC..."
SUBNET_IDS_RAW=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC_ID"
 --query "Subnets[*].SubnetId" --output text)
if [[ -z "$SUBNET_IDS_RAW" ]]; then
    echo "ERROR: No subnets found in default VPC"
    exit 1
fi

# Convert to proper comma-separated format, handling both spaces and tabs
SUBNET_IDS_COMMA=$(echo "$SUBNET_IDS_RAW" | tr -s '[:space:]' ',' | sed 's/,$//')
```

```
echo "Raw subnet IDs: $SUBNET_IDS_RAW"
echo "Formatted subnet IDs: $SUBNET_IDS_COMMA"

# Validate subnet IDs format
if [[ ! "$SUBNET_IDS_COMMA" =~ ^subnet-[a-z0-9]+(,subnet-[a-z0-9]+)*$ ]]; then
    echo "ERROR: Invalid subnet ID format: $SUBNET_IDS_COMMA"
    exit 1
fi

# Step 5: Create ECS service
echo ""
echo "=========================================="
echo "STEP 5: CREATE ECS SERVICE"
echo "=========================================="

# Create the service with proper JSON formatting for network configuration
SERVICE_CMD="aws ecs create-service --cluster '$CLUSTER_NAME' --service-name
 '$SERVICE_NAME' --task-definition '$TASK_FAMILY' --desired-count 1 --launch-
type FARGATE --network-configuration '{\"awsvpcConfiguration\":{\"subnets
\":[\"$(echo "$SUBNET_IDS_COMMA" | sed 's/,/","/g')\"],\"securityGroups
\":[\"$SECURITY_GROUP_ID\"],\"assignPublicIp\":\"ENABLED\"}}' --tags
 key=project,value=doc-smith key=tutorial,value=amazon-ecs-fargate-linux"

echo "Service creation command: $SERVICE_CMD"

SERVICE_OUTPUT=$(execute_command "$SERVICE_CMD" "Create ECS service")
check_for_aws_errors "$SERVICE_OUTPUT" "Create ECS service"

CREATED_RESOURCES+=("ECS Service: $SERVICE_NAME")

# Step 6: Wait for service to stabilize and get public IP
echo ""
echo "=========================================="
echo "STEP 6: WAIT FOR SERVICE AND GET PUBLIC IP"
echo "=========================================="

echo "Waiting for service to stabilize (this may take a few minutes)..."
execute_command "aws ecs wait services-stable --cluster '$CLUSTER_NAME' --
services '$SERVICE_NAME'" "Wait for service to stabilize"

# Get task ARN
TASK_ARN=$(aws ecs list-tasks --cluster "$CLUSTER_NAME" --service-name
 "$SERVICE_NAME" --query "taskArns[0]" --output text)
if [[ "$TASK_ARN" == "None" || -z "$TASK_ARN" ]]; then
```

```
    echo "ERROR: No running tasks found for service"
    exit 1
fi

echo "Task ARN: $TASK_ARN"

# Get network interface ID
ENI_ID=$(aws ecs describe-tasks --cluster "$CLUSTER_NAME" --tasks "$TASK_ARN"
 --query "tasks[0].attachments[0].details[?name=='networkInterfaceId'].value" --
output text)
if [[ "$ENI_ID" == "None" || -z "$ENI_ID" ]]; then
    echo "ERROR: Could not retrieve network interface ID"
    exit 1
fi

# Validate ENI ID format
if [[ ! "$ENI_ID" =~ ^eni-[a-z0-9]{8,17}$ ]]; then
    echo "ERROR: Invalid network interface ID format: $ENI_ID"
    exit 1
fi

echo "Network Interface ID: $ENI_ID"

# Get public IP
PUBLIC_IP=$(aws ec2 describe-network-interfaces --network-interface-ids "$ENI_ID"
 --query "NetworkInterfaces[0].Association.PublicIp" --output text)
if [[ "$PUBLIC_IP" == "None" || -z "$PUBLIC_IP" ]]; then
    echo "WARNING: No public IP assigned to the task"
    echo "The task may be in a private subnet or public IP assignment failed"
else
    # Validate IP format
    if [[ ! "$PUBLIC_IP" =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]];
 then
        echo "ERROR: Invalid IP address format: $PUBLIC_IP"
        PUBLIC_IP=""
    else
        echo ""
        echo "========================================="
        echo "SUCCESS! APPLICATION IS RUNNING"
        echo "========================================="
        echo "Your application is available at: http://$PUBLIC_IP"
        echo "You can test it by opening this URL in your browser"
        echo ""
    fi
```

```
fi

# Display service information
echo ""
echo "============================================="
echo "SERVICE INFORMATION"
echo "============================================="
execute_command "aws ecs describe-services --cluster '$CLUSTER_NAME' --services
 '$SERVICE_NAME'" "Get service details"

echo ""
echo "============================================="
echo "TUTORIAL COMPLETED SUCCESSFULLY"
echo "============================================="
echo "Resources created:"
for resource in "${CREATED_RESOURCES[@]}"; do
    echo "  - $resource"
done

if [[ -n "$PUBLIC_IP" && "$PUBLIC_IP" != "None" ]]; then
    echo ""
    echo "Application URL: http://$PUBLIC_IP"
fi

echo ""
echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress
  - CreateCluster
  - CreateRole
  - CreateSecurityGroup
  - CreateService
  - DeleteCluster
  - DeleteSecurityGroup
  - DeleteService
  - DeregisterTaskDefinition

- DescribeNetworkInterfaces
- DescribeSecurityGroups
- DescribeServices
- DescribeSubnets
- DescribeTasks
- DescribeVpcs
- GetCallerIdentity
- GetRole
- ListTaskDefinitions
- ListTasks
- RegisterTaskDefinition
- UpdateService
- Attendi

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea un'API REST con integrazione proxy di funzioni

L'esempio di codice seguente mostra come:

- Creare un ruolo IAM per l'esecuzione di Lambda
- Creare e distribuire una funzione Lambda
- Crea un'API REST
- Configurare l'integrazione del proxy Lambda
- Distribuzione e test dell'API
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

set -euo pipefail

# Simple API Gateway Lambda Integration Script
# This script creates a REST API with Lambda proxy integration

# Generate random identifiers
FUNCTION_NAME="GetStartedLambdaProxyIntegration-$(openssl rand -hex 4)"
ROLE_NAME="GetStartedLambdaBasicExecutionRole-$(openssl rand -hex 4)"
API_NAME="LambdaProxyAPI-$(openssl rand -hex 4)"

# Get AWS account info
ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
REGION=$(aws configure get region || echo "us-east-1")

# Validate inputs
if [[ -z "$ACCOUNT_ID" ]] || [[ -z "$REGION" ]]; then
    echo "Error: Failed to retrieve AWS account information" >&2
    exit 1
fi

echo "Creating Lambda function code..."

# Create Lambda function code with input validation
cat > lambda_function.py << 'EOF'
import json
import logging

logger = logging.getLogger()
logger.setLevel(logging.INFO)
```

```
def lambda_handler(event, context):
    try:
        logger.info("Received event: %s", json.dumps(event))

        greeter = 'World'

        # Safely retrieve greeter from query string parameters
        query_params = event.get('queryStringParameters') or {}
        if isinstance(query_params, dict) and 'greeter' in query_params:
            greeter_value = query_params.get('greeter')
            if isinstance(greeter_value, str) and greeter_value:
                greeter = greeter_value

        # Safely retrieve greeter from multi-value headers
        multi_headers = event.get('multiValueHeaders') or {}
        if isinstance(multi_headers, dict) and 'greeter' in multi_headers:
            greeter_list = multi_headers.get('greeter', [])
            if isinstance(greeter_list, list) and greeter_list:
                greeter = " and ".join(str(g) for g in greeter_list if g)

        # Safely retrieve greeter from headers
        headers = event.get('headers') or {}
        if isinstance(headers, dict) and 'greeter' in headers:
            greeter_value = headers.get('greeter')
            if isinstance(greeter_value, str) and greeter_value:
                greeter = greeter_value

        # Safely retrieve greeter from body
        body = event.get('body')
        if body and isinstance(body, str):
            try:
                body_dict = json.loads(body)
                if isinstance(body_dict, dict) and 'greeter' in body_dict:
                    greeter_value = body_dict.get('greeter')
                    if isinstance(greeter_value, str) and greeter_value:
                        greeter = greeter_value
            except (json.JSONDecodeError, ValueError) as e:
                logger.warning("Failed to parse body: %s", str(e))

        # Sanitize greeter to prevent injection
        greeter = greeter.replace('"', '\\"').replace("'", "\\'")

        response = {
```

```
            "statusCode": 200,
            "headers": {
                "Content-Type": "application/json"
            },
            "body": json.dumps({"message": f"Hello, {greeter}!"})
        }

        logger.info("Response: %s", json.dumps(response))
        return response

    except Exception as e:
        logger.error("Unexpected error: %s", str(e), exc_info=True)
        return {
            "statusCode": 500,
            "headers": {
                "Content-Type": "application/json"
            },
            "body": json.dumps({"error": "Internal server error"})
        }
EOF

# Create deployment package
zip -q function.zip lambda_function.py || {
    echo "Error: Failed to create function.zip" >&2
    exit 1
}

echo "Creating IAM role..."

# Create IAM trust policy
cat > trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

```
# Create IAM role with error handling
aws iam create-role \
    --role-name "$ROLE_NAME" \
    --assume-role-policy-document file://trust-policy.json \
    --description "Temporary role for Lambda execution" || {
    echo "Error: Failed to create IAM role" >&2
    exit 1
}

aws iam tag-role --role-name "$ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=apigateway-lambda-integration

# Attach execution policy
aws iam attach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" || {
    echo "Error: Failed to attach IAM policy" >&2
    exit 1
}

# Wait for role propagation
sleep 15

echo "Creating Lambda function..."

# Create Lambda function with Python 3.11 (more recent runtime)
aws lambda create-function \
    --function-name "$FUNCTION_NAME" \
    --runtime python3.11 \
    --role "arn:aws:iam::$ACCOUNT_ID:role/$ROLE_NAME" \
    --handler lambda_function.lambda_handler \
    --zip-file fileb://function.zip \
    --timeout 30 \
    --memory-size 128 \
    --environment "Variables={LOG_LEVEL=INFO}" \
    --tags project=doc-smith,tutorial=apigateway-lambda-integration || {
    echo "Error: Failed to create Lambda function" >&2
    exit 1
}

echo "Creating API Gateway..."
```

```
# Create REST API with minimum logging
API_RESPONSE=$(aws apigateway create-rest-api \
    --name "$API_NAME" \
    --endpoint-configuration types=REGIONAL \
    --description "API for Lambda proxy integration tutorial" \
    --tags project=doc-smith,tutorial=apigateway-lambda-integration \
    --output json)

API_ID=$(echo "$API_RESPONSE" | grep -o '"id": "[^"]*"' | head -1 | cut -d'"' -
f4)

if [[ -z "$API_ID" ]]; then
    echo "Error: Failed to create API Gateway" >&2
    exit 1
fi

# Get root resource ID
ROOT_RESOURCE_ID=$(aws apigateway get-resources --rest-api-id "$API_ID" --query
 'items[?path==`/`].id' --output text)

# Create helloworld resource
aws apigateway create-resource \
    --rest-api-id "$API_ID" \
    --parent-id "$ROOT_RESOURCE_ID" \
    --path-part helloworld || {
    echo "Error: Failed to create resource" >&2
    exit 1
}

# Get resource ID
RESOURCE_ID=$(aws apigateway get-resources --rest-api-id "$API_ID" --query
 "items[?pathPart=='helloworld'].id" --output text)

# Create ANY method with no authorization (intentional for tutorial)
aws apigateway put-method \
    --rest-api-id "$API_ID" \
    --resource-id "$RESOURCE_ID" \
    --http-method ANY \
    --authorization-type NONE || {
    echo "Error: Failed to create method" >&2
    exit 1
}

# Set up Lambda proxy integration
```

```
LAMBDA_URI="arn:aws:apigateway:$REGION:lambda:path/2015-03-31/functions/
arn:aws:lambda:$REGION:$ACCOUNT_ID:function:$FUNCTION_NAME/invocations"

aws apigateway put-integration \
    --rest-api-id "$API_ID" \
    --resource-id "$RESOURCE_ID" \
    --http-method ANY \
    --type AWS_PROXY \
    --integration-http-method POST \
    --uri "$LAMBDA_URI" || {
    echo "Error: Failed to create integration" >&2
    exit 1
}

# Grant API Gateway permission to invoke Lambda
STATEMENT_ID="apigateway-invoke-$(openssl rand -hex 4)"
SOURCE_ARN="arn:aws:execute-api:$REGION:$ACCOUNT_ID:$API_ID/*/*"

aws lambda add-permission \
    --function-name "$FUNCTION_NAME" \
    --statement-id "$STATEMENT_ID" \
    --action lambda:InvokeFunction \
    --principal apigateway.amazonaws.com \
    --source-arn "$SOURCE_ARN" || {
    echo "Error: Failed to add Lambda permission" >&2
    exit 1
}

# Deploy API
aws apigateway create-deployment \
    --rest-api-id "$API_ID" \
    --stage-name test \
    --description "Test deployment" || {
    echo "Error: Failed to deploy API" >&2
    exit 1
}

echo "Testing API..."

# Test the API
INVOKE_URL="https://$API_ID.execute-api.$REGION.amazonaws.com/test/helloworld"

echo "API URL: $INVOKE_URL"
```

```
# Test with query parameter (with proper URL encoding)
echo "Testing with query parameter:"
curl -s -X GET "$INVOKE_URL?greeter=John" | jq . 2>/dev/null || curl -s -X GET
 "$INVOKE_URL?greeter=John"
echo ""

# Test with header
echo "Testing with header:"
curl -s -X GET "$INVOKE_URL" \
    -H 'content-type: application/json' \
    -H 'greeter: John' | jq . 2>/dev/null || curl -s -X GET "$INVOKE_URL" \
    -H 'content-type: application/json' \
    -H 'greeter: John'
echo ""

# Test with body
echo "Testing with POST body:"
curl -s -X POST "$INVOKE_URL" \
    -H 'content-type: application/json' \
    -d '{"greeter": "John"}' | jq . 2>/dev/null || curl -s -X POST "$INVOKE_URL"
 \
    -H 'content-type: application/json' \
    -d '{"greeter": "John"}'
echo ""

echo "Tutorial completed! API is available at: $INVOKE_URL"

# Cleanup
echo "Cleaning up resources..."

# Delete API
aws apigateway delete-rest-api --rest-api-id "$API_ID" || echo "Warning: Failed
 to delete API" >&2

# Delete Lambda function
aws lambda delete-function --function-name "$FUNCTION_NAME" || echo "Warning:
 Failed to delete Lambda function" >&2

# Detach policy and delete role
aws iam detach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" || echo "Warning: Failed to detach policy" >&2
```

```
aws iam delete-role --role-name "$ROLE_NAME" || echo "Warning: Failed to delete
 role" >&2

# Clean up local files securely
rm -f lambda_function.py function.zip trust-policy.json

echo "Cleanup completed!"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AddPermission
  - AttachRolePolicy
  - CreateDeployment
  - CreateFunction
  - CreateResource
  - CreateRestApi
  - CreateRole
  - DeleteFunction
  - DeleteRestApi
  - DeleteRole
  - DetachRolePolicy
  - GetCallerIdentity
  - GetResources
  - GetRestApis
  - PutIntegration
  - PutMethod

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea utenti IAM di sola lettura e lettura-scrittura utilizzando un AWS SDK

L'esempio di codice seguente mostra come creare degli utenti e collegarvi delle policy.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come Centro identità AWS IAM.

- Creare due utenti IAM.
- Collega una policy che consenta a un utente di ottenere e inserire oggetti in un bucket Amazon S3.
- Collega una policy che consenta all'altro utente di ottenere oggetti dal bucket.
- Ottieni autorizzazioni diverse per il bucket in base alle credenziali dell'utente.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
import logging
import time

import boto3
from botocore.exceptions import ClientError

import access_key_wrapper
import policy_wrapper

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def create_user(user_name):
    """
    Creates a user. By default, a user has no permissions or access keys.
```

```
    :param user_name: The name of the user.
    :return: The newly created user.
    """
    try:
        user = iam.create_user(UserName=user_name)
        logger.info("Created user %s.", user.name)
    except ClientError:
        logger.exception("Couldn't create user %s.", user_name)
        raise
    else:
        return user



def update_user(user_name, new_user_name):
    """
    Updates a user's name.

    :param user_name: The current name of the user to update.
    :param new_user_name: The new name to assign to the user.
    :return: The updated user.
    """
    try:
        user = iam.User(user_name)
        user.update(NewUserName=new_user_name)
        logger.info("Renamed %s to %s.", user_name, new_user_name)
    except ClientError:
        logger.exception("Couldn't update name for user %s.", user_name)
        raise
    return user



def list_users():
    """
    Lists the users in the current account.

    :return: The list of users.
    """
    try:
        users = list(iam.users.all())
        logger.info("Got %s users.", len(users))
    except ClientError:
```

```
        logger.exception("Couldn't get users.")
        raise
    else:
        return users



def delete_user(user_name):
    """
    Deletes a user. Before a user can be deleted, all associated resources,
    such as access keys and policies, must be deleted or detached.

    :param user_name: The name of the user.
    """
    try:
        iam.User(user_name).delete()
        logger.info("Deleted user %s.", user_name)
    except ClientError:
        logger.exception("Couldn't delete user %s.", user_name)
        raise



def attach_policy(user_name, policy_arn):
    """
    Attaches a policy to a user.

    :param user_name: The name of the user.
    :param policy_arn: The Amazon Resource Name (ARN) of the policy.
    """
    try:
        iam.User(user_name).attach_policy(PolicyArn=policy_arn)
        logger.info("Attached policy %s to user %s.", policy_arn, user_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to user %s.", policy_arn,
 user_name)
        raise



def detach_policy(user_name, policy_arn):
    """
    Detaches a policy from a user.
```

```
    :param user_name: The name of the user.
    :param policy_arn: The Amazon Resource Name (ARN) of the policy.
    """
    try:
        iam.User(user_name).detach_policy(PolicyArn=policy_arn)
        logger.info("Detached policy %s from user %s.", policy_arn, user_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from user %s.", policy_arn, user_name
        )
        raise
```

Crea funzioni che eseguono il wrapping delle operazioni delle policy IAM.

```
import json
import logging
import operator
import pprint
import time

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def create_policy(name, description, actions, resource_arn):
    """
    Creates a policy that contains a single statement.

    :param name: The name of the policy to create.
    :param description: The description of the policy.
    :param actions: The actions allowed by the policy. These typically take the
                    form of service:action, such as s3:PutObject.
    :param resource_arn: The Amazon Resource Name (ARN) of the resource this
 policy
                         applies to. This ARN can contain wildcards, such as
                         'arn:aws:s3:::amzn-s3-demo-bucket/*' to allow actions on
 all objects
                         in the bucket named 'amzn-s3-demo-bucket'.
```

```
    :return: The newly created policy.
    """
    policy_doc = {
        "Version":"2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
 resource_arn}],
    }
    try:
        policy = iam.create_policy(
            PolicyName=name,
            Description=description,
            PolicyDocument=json.dumps(policy_doc),
        )
        logger.info("Created policy %s.", policy.arn)
    except ClientError:
        logger.exception("Couldn't create policy %s.", name)
        raise
    else:
        return policy



def delete_policy(policy_arn):
    """
    Deletes a policy.

    :param policy_arn: The ARN of the policy to delete.
    """
    try:
        iam.Policy(policy_arn).delete()
        logger.info("Deleted policy %s.", policy_arn)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_arn)
        raise
```

Crea funzioni che eseguono il wrapping delle operazioni delle chiavi di accesso IAM.

```
import logging
import boto3
from botocore.exceptions import ClientError
```

```
logger = logging.getLogger(__name__)

iam = boto3.resource("iam")

def create_key(user_name):
    """
    Creates an access key for the specified user. Each user can have a
    maximum of two keys.

    :param user_name: The name of the user.
    :return: The created access key.
    """
    try:
        key_pair = iam.User(user_name).create_access_key_pair()
        logger.info(
            "Created access key pair for %s. Key ID is %s.",
            key_pair.user_name,
            key_pair.id,
        )
    except ClientError:
        logger.exception("Couldn't create access key pair for %s.", user_name)
        raise
    else:
        return key_pair



def delete_key(user_name, key_id):
    """
    Deletes a user's access key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to delete.
    """

    try:
        key = iam.AccessKey(user_name, key_id)
        key.delete()
        logger.info("Deleted access key %s for %s.", key.id, key.user_name)
    except ClientError:
        logger.exception("Couldn't delete key %s for %s", key_id, user_name)
        raise
```

Utilizza le funzioni di wrapping per creare utenti con policy diverse e usa le loro credenziali per accedere a un bucket Amazon S3.

```
def usage_demo():
    """
    Shows how to manage users, keys, and policies.
    This demonstration creates two users: one user who can put and get objects in
 an
    Amazon S3 bucket, and another user who can only get objects from the bucket.
    The demo then shows how the users can perform only the actions they are
 permitted
    to perform.
    """
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management user demo.")
    print("-" * 88)
    print(
        "Users can have policies and roles attached to grant them specific "
        "permissions."
    )
    s3 = boto3.resource("s3")
    bucket = s3.create_bucket(
        Bucket=f"demo-iam-bucket-{time.time_ns()}",
        CreateBucketConfiguration={
            "LocationConstraint": s3.meta.client.meta.region_name
        },
    )
    print(f"Created an Amazon S3 bucket named {bucket.name}.")
    user_read_writer = create_user("demo-iam-read-writer")
    user_reader = create_user("demo-iam-reader")
    print(f"Created two IAM users: {user_read_writer.name} and
 {user_reader.name}")
    update_user(user_read_writer.name, "demo-iam-creator")
    update_user(user_reader.name, "demo-iam-getter")
    users = list_users()
    user_read_writer = next(
        user for user in users if user.user_id == user_read_writer.user_id
    )
```

```
    user_reader = next(user for user in users if user.user_id ==
user_reader.user_id)
    print(
        f"Changed the names of the users to {user_read_writer.name} "
        f"and {user_reader.name}."
    )

    read_write_policy = policy_wrapper.create_policy(
        "demo-iam-read-write-policy",
        "Grants rights to create and get an object in the demo bucket.",
        ["s3:PutObject", "s3:GetObject"],
        f"arn:aws:s3:::{bucket.name}/*",
    )
    print(
        f"Created policy {read_write_policy.policy_name} with ARN:
{read_write_policy.arn}"
    )
    print(read_write_policy.description)
    read_policy = policy_wrapper.create_policy(
        "demo-iam-read-policy",
        "Grants rights to get an object from the demo bucket.",
        "s3:GetObject",
        f"arn:aws:s3:::{bucket.name}/*",
    )
    print(f"Created policy {read_policy.policy_name} with ARN:
{read_policy.arn}")
    print(read_policy.description)
    attach_policy(user_read_writer.name, read_write_policy.arn)
    print(f"Attached {read_write_policy.policy_name} to
{user_read_writer.name}.")
    attach_policy(user_reader.name, read_policy.arn)
    print(f"Attached {read_policy.policy_name} to {user_reader.name}.")

    user_read_writer_key = access_key_wrapper.create_key(user_read_writer.name)
    print(f"Created access key pair for {user_read_writer.name}.")
    user_reader_key = access_key_wrapper.create_key(user_reader.name)
    print(f"Created access key pair for {user_reader.name}.")

    s3_read_writer_resource = boto3.resource(
        "s3",
        aws_access_key_id=user_read_writer_key.id,
        aws_secret_access_key=user_read_writer_key.secret,
    )
    demo_object_key = f"object-{time.time_ns()}"
```

```
    demo_object = None
    while demo_object is None:
        try:
            demo_object = s3_read_writer_resource.Bucket(bucket.name).put_object(
                Key=demo_object_key, Body=b"AWS IAM demo object content!"
            )
        except ClientError as error:
            if error.response["Error"]["Code"] == "InvalidAccessKeyId":
                print("Access key not yet available. Waiting...")
                time.sleep(1)
            else:
                raise
    print(
        f"Put {demo_object_key} into {bucket.name} using "
        f"{user_read_writer.name}'s credentials."
    )

    read_writer_object = s3_read_writer_resource.Bucket(bucket.name).Object(
        demo_object_key
    )
    read_writer_content = read_writer_object.get()["Body"].read()
    print(f"Got object {read_writer_object.key} using read-writer user's
credentials.")
    print(f"Object content: {read_writer_content}")

    s3_reader_resource = boto3.resource(
        "s3",
        aws_access_key_id=user_reader_key.id,
        aws_secret_access_key=user_reader_key.secret,
    )
    demo_content = None
    while demo_content is None:
        try:
            demo_object =
s3_reader_resource.Bucket(bucket.name).Object(demo_object_key)
            demo_content = demo_object.get()["Body"].read()
            print(f"Got object {demo_object.key} using reader user's
credentials.")
            print(f"Object content: {demo_content}")
        except ClientError as error:
            if error.response["Error"]["Code"] == "InvalidAccessKeyId":
                print("Access key not yet available. Waiting...")
                time.sleep(1)
            else:
```

```
                raise

    try:
        demo_object.delete()
    except ClientError as error:
        if error.response["Error"]["Code"] == "AccessDenied":
            print("-" * 88)
            print(
                "Tried to delete the object using the reader user's credentials.
"
                "Got expected AccessDenied error because the reader is not "
                "allowed to delete objects."
            )
            print("-" * 88)

    access_key_wrapper.delete_key(user_reader.name, user_reader_key.id)
    detach_policy(user_reader.name, read_policy.arn)
    policy_wrapper.delete_policy(read_policy.arn)
    delete_user(user_reader.name)
    print(f"Deleted keys, detached and deleted policy, and deleted
{user_reader.name}.")

    access_key_wrapper.delete_key(user_read_writer.name, user_read_writer_key.id)
    detach_policy(user_read_writer.name, read_write_policy.arn)
    policy_wrapper.delete_policy(read_write_policy.arn)
    delete_user(user_read_writer.name)
    print(
        f"Deleted keys, detached and deleted policy, and deleted
{user_read_writer.name}."
    )

    bucket.objects.delete()
    bucket.delete()
    print(f"Emptied and deleted {bucket.name}.")
    print("Thanks for watching!")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - AttachUserPolicy

- CreateAccessKey
- CreatePolicy
- CreateUser
- DeleteAccessKey
- DeletePolicy
- DeleteUser
- DetachUserPolicy
- ListUsers
- UpdateUser

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione di un servizio container per le istanze di macchine virtuali

L'esempio di codice seguente mostra come:

- Crea un cluster ECS
- Crea e monitora un servizio
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# ECS EC2 Launch Type Tutorial Script - UPDATED VERSION
```

```
# This script demonstrates creating an ECS cluster, launching a container
 instance,
# registering a task definition, and creating a service using the EC2 launch
 type.
# Updated to match the tutorial draft with nginx web server and service creation.
#
# - UPDATED: Changed from sleep task to nginx web server with service

set -e  # Exit on any error

# Configuration
SCRIPT_NAME="ecs-ec2-tutorial"
LOG_FILE="${SCRIPT_NAME}-$(date +%Y%m%d-%H%M%S).log"
CLUSTER_NAME="tutorial-cluster-$(openssl rand -hex 4)"
TASK_FAMILY="nginx-task-$(openssl rand -hex 4)"
SERVICE_NAME="nginx-service-$(openssl rand -hex 4)"
KEY_PAIR_NAME="ecs-tutorial-key-$(openssl rand -hex 4)"
SECURITY_GROUP_NAME="ecs-tutorial-sg-$(openssl rand -hex 4)"

# Tags
PROJECT_TAG="doc-smith"
TUTORIAL_TAG="ecs-ec2"

# Get current AWS region dynamically
AWS_REGION=$(aws configure get region || echo "us-east-1")

# Resource tracking arrays
CREATED_RESOURCES=()
CLEANUP_ORDER=()

# Logging function
log() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | tee -a "$LOG_FILE"
}

# Error handling function
handle_error() {
    local exit_code=$?
    log "ERROR: Script failed with exit code $exit_code"
    log "ERROR: Last command: $BASH_COMMAND"

    echo ""
    echo "========================================="
    echo "ERROR OCCURRED - ATTEMPTING CLEANUP"
```

```
    echo "========================================="
    echo "Resources created before error:"
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "  - $resource"
    done

    cleanup_resources
    exit $exit_code
}

# Set error trap
trap handle_error ERR

# FIXED: Enhanced cleanup function with proper error handling and logging
cleanup_resources() {
    log "Starting cleanup process..."
    local cleanup_errors=0

    # Delete service first (this will stop tasks automatically)
    if [[ -n "${SERVICE_ARN:-}" ]]; then
        log "Updating service to desired count 0: $SERVICE_NAME"
        if ! aws ecs update-service --cluster "$CLUSTER_NAME" --service
 "$SERVICE_NAME" --desired-count 0 2>>"$LOG_FILE"; then
            log "WARNING: Failed to update service desired count to 0"
            ((cleanup_errors++))
        else
            log "Waiting for service tasks to stop..."
            sleep 30  # Give time for tasks to stop
        fi

        log "Deleting service: $SERVICE_NAME"
        if ! aws ecs delete-service --cluster "$CLUSTER_NAME" --service
 "$SERVICE_NAME" 2>>"$LOG_FILE"; then
            log "WARNING: Failed to delete service $SERVICE_NAME"
            ((cleanup_errors++))
        fi
    fi

    # Stop and delete any remaining tasks
    if [[ -n "${TASK_ARN:-}" ]]; then
        log "Stopping task: $TASK_ARN"
        if ! aws ecs stop-task --cluster "$CLUSTER_NAME" --task "$TASK_ARN" --
reason "Tutorial cleanup" 2>>"$LOG_FILE"; then
            log "WARNING: Failed to stop task $TASK_ARN"
```

```
            ((cleanup_errors++))
        else
            log "Waiting for task to stop..."
            if ! aws ecs wait tasks-stopped --cluster "$CLUSTER_NAME" --tasks
"$TASK_ARN" 2>>"$LOG_FILE"; then
                log "WARNING: Task stop wait failed for $TASK_ARN"
                ((cleanup_errors++))
            fi
        fi
    fi

    # Deregister task definition
    if [[ -n "${TASK_DEFINITION_ARN:-}" ]]; then
        log "Deregistering task definition: $TASK_DEFINITION_ARN"
        if ! aws ecs deregister-task-definition --task-definition
"$TASK_DEFINITION_ARN" 2>>"$LOG_FILE"; then
            log "WARNING: Failed to deregister task definition
$TASK_DEFINITION_ARN"
            ((cleanup_errors++))
        fi
    fi

    # Terminate EC2 instance
    if [[ -n "${INSTANCE_ID:-}" ]]; then
        log "Terminating EC2 instance: $INSTANCE_ID"
        if ! aws ec2 terminate-instances --instance-ids "$INSTANCE_ID"
2>>"$LOG_FILE"; then
            log "WARNING: Failed to terminate instance $INSTANCE_ID"
            ((cleanup_errors++))
        else
            log "Waiting for instance to terminate..."
            if ! aws ec2 wait instance-terminated --instance-ids "$INSTANCE_ID"
2>>"$LOG_FILE"; then
                log "WARNING: Instance termination wait failed for $INSTANCE_ID"
                ((cleanup_errors++))
            fi
        fi
    fi

    # Delete security group with retry logic
    if [[ -n "${SECURITY_GROUP_ID:-}" ]]; then
        log "Deleting security group: $SECURITY_GROUP_ID"
        local retry_count=0
        local max_retries=3
```

```
       while [[ $retry_count -lt $max_retries ]]; do
           if aws ec2 delete-security-group --group-id "$SECURITY_GROUP_ID"
2>>"$LOG_FILE"; then
               log "Successfully deleted security group"
               break
           else
               ((retry_count++))
               if [[ $retry_count -lt $max_retries ]]; then
                   log "Retry $retry_count/$max_retries: Waiting 10 seconds
before retrying security group deletion..."
                   sleep 10
               else
                   log "ERROR: Failed to delete security group after
$max_retries attempts"
                   ((cleanup_errors++))
               fi
           fi
       done
   fi

   # Delete key pair
   if [[ -n "${KEY_PAIR_NAME:-}" ]]; then
       log "Deleting key pair: $KEY_PAIR_NAME"
       if ! aws ec2 delete-key-pair --key-name "$KEY_PAIR_NAME" 2>>"$LOG_FILE";
then
           log "WARNING: Failed to delete key pair $KEY_PAIR_NAME"
           ((cleanup_errors++))
       fi
       rm -f "${KEY_PAIR_NAME}.pem" 2>>"$LOG_FILE" || log "WARNING: Failed to
remove local key file"
   fi

   # Delete ECS cluster
   if [[ -n "${CLUSTER_NAME:-}" ]]; then
       log "Deleting ECS cluster: $CLUSTER_NAME"
       if ! aws ecs delete-cluster --cluster "$CLUSTER_NAME" 2>>"$LOG_FILE";
then
           log "WARNING: Failed to delete cluster $CLUSTER_NAME"
           ((cleanup_errors++))
       fi
   fi

   if [[ $cleanup_errors -eq 0 ]]; then
```

```
        log "Cleanup completed successfully"
    else
        log "Cleanup completed with $cleanup_errors warnings/errors. Check log
 file for details."
    fi
}

# Function to check prerequisites
check_prerequisites() {
    log "Checking prerequisites..."

    # Check AWS CLI
    if ! command -v aws &> /dev/null; then
        log "ERROR: AWS CLI is not installed"
        exit 1
    fi

    # Check AWS credentials
    if ! aws sts get-caller-identity &> /dev/null; then
        log "ERROR: AWS credentials not configured"
        exit 1
    fi

    # Get caller identity
    CALLER_IDENTITY=$(aws sts get-caller-identity --output text --query
 'Account')
    log "AWS Account: $CALLER_IDENTITY"
    log "AWS Region: $AWS_REGION"

    # Check for default VPC
    DEFAULT_VPC=$(aws ec2 describe-vpcs --filters "Name=is-default,Values=true"
 --query 'Vpcs[0].VpcId' --output text)
    if [[ "$DEFAULT_VPC" == "None" ]]; then
        log "ERROR: No default VPC found. Please create a VPC first."
        exit 1
    fi
    log "Using default VPC: $DEFAULT_VPC"

    # Get default subnet
    DEFAULT_SUBNET=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=
$DEFAULT_VPC" "Name=default-for-az,Values=true" --query 'Subnets[0].SubnetId' --
output text)
    if [[ "$DEFAULT_SUBNET" == "None" ]]; then
        log "ERROR: No default subnet found"
```

```
        exit 1
    fi
    log "Using default subnet: $DEFAULT_SUBNET"

    log "Prerequisites check completed successfully"
}

# Function to create ECS cluster
create_cluster() {
    log "Creating ECS cluster: $CLUSTER_NAME"

    CLUSTER_ARN=$(aws ecs create-cluster --cluster-name "$CLUSTER_NAME" --
tags key=project,value=$PROJECT_TAG key=tutorial,value=$TUTORIAL_TAG --query
 'cluster.clusterArn' --output text)

    if [[ -z "$CLUSTER_ARN" ]]; then
        log "ERROR: Failed to create cluster"
        exit 1
    fi

    log "Created cluster: $CLUSTER_ARN"
    CREATED_RESOURCES+=("ECS Cluster: $CLUSTER_NAME")
}

# Function to create key pair
create_key_pair() {
    log "Creating EC2 key pair: $KEY_PAIR_NAME"

    # FIXED: Set secure umask before key creation
    umask 077
    aws ec2 create-key-pair --key-name "$KEY_PAIR_NAME" --query 'KeyMaterial' --
output text > "${KEY_PAIR_NAME}.pem"
    chmod 400 "${KEY_PAIR_NAME}.pem"
    umask 022  # Reset umask

    log "Created key pair: $KEY_PAIR_NAME"
    CREATED_RESOURCES+=("EC2 Key Pair: $KEY_PAIR_NAME")

    aws ec2 create-tags --resources "$KEY_PAIR_NAME" --tags Key=project,Value=
$PROJECT_TAG Key=tutorial,Value=$TUTORIAL_TAG 2>>"$LOG_FILE" || log "WARNING:
 Failed to tag key pair"
}

# Function to create security group
```

```
create_security_group() {
    log "Creating security group: $SECURITY_GROUP_NAME"

    SECURITY_GROUP_ID=$(aws ec2 create-security-group \
        --group-name "$SECURITY_GROUP_NAME" \
        --description "ECS tutorial security group" \
        --vpc-id "$DEFAULT_VPC" \
        --tag-specifications "ResourceType=security-
group,Tags=[{Key=project,Value=$PROJECT_TAG},{Key=tutorial,Value=$TUTORIAL_TAG}]"
 \
        --query 'GroupId' --output text)

    if [[ -z "$SECURITY_GROUP_ID" ]]; then
        log "ERROR: Failed to create security group"
        exit 1
    fi

    # Add HTTP access rule for nginx web server with restricted CIDR
    # SECURITY FIX: Restrict access to specific CIDR if available, otherwise
 document the risk
    log "WARNING: Security group allows HTTP (port 80) from 0.0.0.0/0 - restrict
 this in production"
    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 80 \
        --cidr "0.0.0.0/0"

    log "Created security group: $SECURITY_GROUP_ID"
    log "Added HTTP access on port 80"
    CREATED_RESOURCES+=("Security Group: $SECURITY_GROUP_ID")
}

# Function to get ECS optimized AMI
get_ecs_ami() {
    log "Getting ECS-optimized AMI ID..."

    ECS_AMI_ID=$(aws ssm get-parameters \
        --names /aws/service/ecs/optimized-ami/amazon-linux-2/recommended \
        --query 'Parameters[0].Value' --output text | jq -r '.image_id')

    if [[ -z "$ECS_AMI_ID" ]]; then
        log "ERROR: Failed to get ECS-optimized AMI ID"
        exit 1
```

```
    fi

    log "ECS-optimized AMI ID: $ECS_AMI_ID"
}

# Function to create IAM role for ECS instance (if it doesn't exist)
ensure_ecs_instance_role() {
    log "Checking for ecsInstanceRole..."

    if ! aws iam get-role --role-name ecsInstanceRole &> /dev/null; then
        log "Creating ecsInstanceRole..."

        # Create trust policy
        cat > ecs-instance-trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

        # SECURITY FIX: Validate JSON before using
        if ! jq empty ecs-instance-trust-policy.json 2>/dev/null; then
            log "ERROR: Invalid JSON in trust policy"
            rm -f ecs-instance-trust-policy.json
            exit 1
        fi

        # Create role
        aws iam create-role \
            --role-name ecsInstanceRole \
            --assume-role-policy-document file://ecs-instance-trust-policy.json

        aws iam tag-role --role-name ecsInstanceRole --tags Key=project,Value=
$PROJECT_TAG Key=tutorial,Value=$TUTORIAL_TAG

        # Attach managed policy
```

```
        aws iam attach-role-policy \
            --role-name ecsInstanceRole \
            --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role

        # Create instance profile
        aws iam create-instance-profile --instance-profile-name ecsInstanceRole

        # Add role to instance profile
        aws iam add-role-to-instance-profile \
            --instance-profile-name ecsInstanceRole \
            --role-name ecsInstanceRole

        # FIXED: Enhanced wait for role to be ready
        log "Waiting for IAM role to be ready..."
        aws iam wait role-exists --role-name ecsInstanceRole
        sleep 30  # Additional buffer for eventual consistency

        rm -f ecs-instance-trust-policy.json
        log "Created ecsInstanceRole"
        CREATED_RESOURCES+=("IAM Role: ecsInstanceRole")
    else
        log "ecsInstanceRole already exists"
    fi
}

# Function to launch container instance
launch_container_instance() {
    log "Launching ECS container instance..."

    # Create user data script
    cat > ecs-user-data.sh << EOF
#!/bin/bash
echo ECS_CLUSTER=$CLUSTER_NAME >> /etc/ecs/ecs.config
EOF

    # SECURITY FIX: Validate user data script before use
    if ! bash -n ecs-user-data.sh 2>/dev/null; then
        log "ERROR: Invalid user data script"
        rm -f ecs-user-data.sh
        exit 1
    fi

    INSTANCE_ID=$(aws ec2 run-instances \
```

```
        --image-id "$ECS_AMI_ID" \
        --instance-type t3.micro \
        --key-name "$KEY_PAIR_NAME" \
        --security-group-ids "$SECURITY_GROUP_ID" \
        --subnet-id "$DEFAULT_SUBNET" \
        --iam-instance-profile Name=ecsInstanceRole \
        --user-data file://ecs-user-data.sh \
        --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=ecs-
tutorial-instance},{Key=project,Value=$PROJECT_TAG},{Key=tutorial,Value=
$TUTORIAL_TAG}]" \
        --monitoring Enabled=false \
        --metadata-options HttpTokens=required,HttpPutResponseHopLimit=1 \
        --query 'Instances[0].InstanceId' --output text)

    if [[ -z "$INSTANCE_ID" ]]; then
        log "ERROR: Failed to launch EC2 instance"
        rm -f ecs-user-data.sh
        exit 1
    fi

    log "Launched EC2 instance: $INSTANCE_ID"
    log "Instance metadata options: IMDSv2 enforced with hop limit 1"
    CREATED_RESOURCES+=("EC2 Instance: $INSTANCE_ID")

    # Wait for instance to be running
    log "Waiting for instance to be running..."
    aws ec2 wait instance-running --instance-ids "$INSTANCE_ID"

    # Wait for ECS agent to register
    log "Waiting for ECS agent to register with cluster..."
    local max_attempts=30
    local attempt=0

    while [[ $attempt -lt $max_attempts ]]; do
        CONTAINER_INSTANCES=$(aws ecs list-container-instances --cluster
 "$CLUSTER_NAME" --query 'containerInstanceArns' --output text)
        if [[ -n "$CONTAINER_INSTANCES" && "$CONTAINER_INSTANCES" != "None" ]];
 then
            log "Container instance registered successfully"
            break
        fi

        attempt=$((attempt + 1))
```

```
        log "Waiting for container instance registration... (attempt $attempt/
$max_attempts)"
        sleep 10
    done

    if [[ $attempt -eq $max_attempts ]]; then
        log "ERROR: Container instance failed to register within expected time"
        rm -f ecs-user-data.sh
        exit 1
    fi

    rm -f ecs-user-data.sh
}

# Function to register task definition
register_task_definition() {
    log "Creating task definition..."

    # Create nginx task definition JSON matching the tutorial
    cat > task-definition.json << 'EOF'
{
    "family": "TASK_FAMILY_PLACEHOLDER",
    "containerDefinitions": [
        {
            "name": "nginx",
            "image": "public.ecr.aws/docker/library/nginx:latest",
            "cpu": 256,
            "memory": 512,
            "essential": true,
            "portMappings": [
                {
                    "containerPort": 80,
                    "hostPort": 80,
                    "protocol": "tcp"
                }
            ],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/nginx-task",
                    "awslogs-region": "REGION_PLACEHOLDER",
                    "awslogs-stream-prefix": "ecs"
                }
            }
```

```
        }
    ],
    "requiresCompatibilities": ["EC2"],
    "networkMode": "bridge",
    "tags": [
        {
            "key": "project",
            "value": "PROJECT_TAG_PLACEHOLDER"
        },
        {
            "key": "tutorial",
            "value": "TUTORIAL_TAG_PLACEHOLDER"
        }
    ]
}
EOF

    # Replace placeholders securely
    sed -i "s|TASK_FAMILY_PLACEHOLDER|$TASK_FAMILY|g" task-definition.json
    sed -i "s|REGION_PLACEHOLDER|$AWS_REGION|g" task-definition.json
    sed -i "s|PROJECT_TAG_PLACEHOLDER|$PROJECT_TAG|g" task-definition.json
    sed -i "s|TUTORIAL_TAG_PLACEHOLDER|$TUTORIAL_TAG|g" task-definition.json

    # FIXED: Validate JSON before registration
    if ! jq empty task-definition.json 2>/dev/null; then
        log "ERROR: Invalid JSON in task definition"
        rm -f task-definition.json
        exit 1
    fi

    TASK_DEFINITION_ARN=$(aws ecs register-task-definition \
        --cli-input-json file://task-definition.json \
        --query 'taskDefinition.taskDefinitionArn' --output text)

    if [[ -z "$TASK_DEFINITION_ARN" ]]; then
        log "ERROR: Failed to register task definition"
        rm -f task-definition.json
        exit 1
    fi

    log "Registered task definition: $TASK_DEFINITION_ARN"
    log "Task definition includes CloudWatch Logs configuration for monitoring"
    CREATED_RESOURCES+=("Task Definition: $TASK_DEFINITION_ARN")
```

```
    rm -f task-definition.json
}

# Function to create service
create_service() {
    log "Creating ECS service..."

    SERVICE_ARN=$(aws ecs create-service \
        --cluster "$CLUSTER_NAME" \
        --service-name "$SERVICE_NAME" \
        --task-definition "$TASK_FAMILY" \
        --desired-count 1 \
        --tags key=project,value=$PROJECT_TAG key=tutorial,value=$TUTORIAL_TAG \
        --query 'service.serviceArn' --output text)

    if [[ -z "$SERVICE_ARN" ]]; then
        log "ERROR: Failed to create service"
        exit 1
    fi

    log "Created service: $SERVICE_ARN"
    CREATED_RESOURCES+=("ECS Service: $SERVICE_NAME")

    # Wait for service to be stable
    log "Waiting for service to be stable..."
    aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$SERVICE_NAME"

    log "Service is now stable and running"

    # Get the task ARN for monitoring
    TASK_ARN=$(aws ecs list-tasks --cluster "$CLUSTER_NAME" --service-name
 "$SERVICE_NAME" --query 'taskArns[0]' --output text)
    if [[ -n "$TASK_ARN" && "$TASK_ARN" != "None" ]]; then
        log "Service task: $TASK_ARN"
        CREATED_RESOURCES+=("ECS Task: $TASK_ARN")
    fi
}

# Function to demonstrate monitoring and testing
demonstrate_monitoring() {
    log "Demonstrating monitoring capabilities..."

    # List services
```

```
    log "Listing services in cluster:"
    aws ecs list-services --cluster "$CLUSTER_NAME" --output table

    # Describe service
    log "Service details:"
    aws ecs describe-services --cluster "$CLUSTER_NAME" --
services "$SERVICE_NAME" --output table --query 'services[0].
{ServiceName:serviceName,Status:status,RunningCount:runningCount,DesiredCount:desiredCoun

    # List tasks
    log "Listing tasks in service:"
    aws ecs list-tasks --cluster "$CLUSTER_NAME" --service-name "$SERVICE_NAME"
 --output table

    # Describe task
    if [[ -n "$TASK_ARN" && "$TASK_ARN" != "None" ]]; then
        log "Task details:"
        aws ecs describe-tasks --cluster "$CLUSTER_NAME"
 --tasks "$TASK_ARN" --output table --query 'tasks[0].
{TaskArn:taskArn,LastStatus:lastStatus,DesiredStatus:desiredStatus,CreatedAt:createdAt}'
    fi

    # List container instances
    log "Container instances in cluster:"
    aws ecs list-container-instances --cluster "$CLUSTER_NAME" --output table

    # Describe container instance
    CONTAINER_INSTANCE_ARN=$(aws ecs list-container-instances --cluster
 "$CLUSTER_NAME" --query 'containerInstanceArns[0]' --output text)
    if [[ -n "$CONTAINER_INSTANCE_ARN" && "$CONTAINER_INSTANCE_ARN" != "None" ]];
 then
        log "Container instance details:"
        aws ecs describe-container-instances --cluster "$CLUSTER_NAME"
 --container-instances "$CONTAINER_INSTANCE_ARN" --output table --query
 'containerInstances[0].
{Arn:containerInstanceArn,Status:status,RunningTasks:runningTasksCount,PendingTasks:pendi
    fi

    # Test the nginx web server
    log "Testing nginx web server..."
    PUBLIC_IP=$(aws ec2 describe-instances --instance-ids "$INSTANCE_ID" --query
 'Reservations[0].Instances[0].PublicIpAddress' --output text)

    if [[ -n "$PUBLIC_IP" && "$PUBLIC_IP" != "None" ]]; then
```

```
        log "Container instance public IP: $PUBLIC_IP"
        log "Testing HTTP connection to nginx..."

        # Wait a moment for nginx to be fully ready
        sleep 10

        if curl -s --connect-timeout 10 "http://$PUBLIC_IP" | grep -q "Welcome to
 nginx"; then
            log "SUCCESS: Nginx web server is responding correctly"
            echo ""
            echo "=========================================="
            echo "WEB SERVER TEST SUCCESSFUL"
            echo "=========================================="
            echo "You can access your nginx web server at: http://$PUBLIC_IP"
            echo "The nginx welcome page should be visible in your browser."
        else
            log "WARNING: Nginx web server may not be fully ready yet. Try
 accessing http://$PUBLIC_IP in a few minutes."
        fi
    else
        log "WARNING: Could not retrieve public IP address"
    fi
}

# Main execution
main() {
    log "Starting ECS EC2 Launch Type Tutorial (UPDATED VERSION)"
    log "Log file: $LOG_FILE"
    log "Security improvements: IMDSv2 enforced, JSON validation, input
 sanitization, CloudWatch Logs configured"

    check_prerequisites
    create_cluster
    create_key_pair
    create_security_group
    get_ecs_ami
    ensure_ecs_instance_role
    launch_container_instance
    register_task_definition
    create_service
    demonstrate_monitoring

    log "Tutorial completed successfully!"
```

```
    echo ""
    echo "========================================="
    echo "TUTORIAL COMPLETED SUCCESSFULLY"
    echo "========================================="
    echo "Resources created:"
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "  - $resource"
    done
    echo ""
    echo "The nginx service will continue running and maintain the desired task
 count."
    echo "You can monitor the service status using:"
    echo "  aws ecs describe-services --cluster $CLUSTER_NAME --services
 $SERVICE_NAME"
    echo ""
    if [[ -n "${PUBLIC_IP:-}" ]]; then
        echo "Access your web server at: http://$PUBLIC_IP"
        echo ""
    fi
    echo "========================================="
    echo "CLEANUP CONFIRMATION"
    echo "========================================="
    log "Auto-confirming cleanup - proceeding with resource cleanup"
    cleanup_resources
    log "All resources have been cleaned up"
}

# Run main function
main "$@"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AddRoleToInstanceProfile
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress
  - CreateCluster
  - CreateInstanceProfile
  - CreateKeyPair
  - CreateRole
  - CreateSecurityGroup

- CreateService
- DeleteCluster
- DeleteKeyPair
- DeleteSecurityGroup
- DeleteService
- DeregisterTaskDefinition
- DescribeContainerInstances
- DescribeInstances
- DescribeServices
- DescribeSubnets
- DescribeTasks
- DescribeVpcs
- GetCallerIdentity
- GetParameters
- GetRole
- ListContainerInstances
- ListServices
- ListTasks
- RegisterTaskDefinition
- RunInstances
- StopTask
- TerminateInstances
- UpdateService
- Attendi (Amazon EC2)
- Attendi (Amazon ECS)
- Aspetta (IAM)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione di un'area di lavoro di monitoraggio gestita

L'esempio di codice seguente mostra come:

- Crea un ruolo IAM per il tuo spazio di lavoro
- Crea uno spazio di lavoro Grafana
- Configura l'autenticazione
- Configura le impostazioni opzionali
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon Managed Grafana Workspace Creation Script
# This script creates an Amazon Managed Grafana workspace and configures it

# Set up logging
LOG_FILE="grafana-workspace-creation.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon Managed Grafana workspace creation script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2

    if echo "$output" | grep -i "error\|exception\|fail" > /dev/null; then
        echo "ERROR: Command '$cmd' failed with output:"
```

```
        echo "$output"
        cleanup_on_error
        exit 1
    fi
}

# Function to clean up resources on error
cleanup_on_error() {
    echo "Error encountered. Attempting to clean up resources..."

    if [ -n "$WORKSPACE_ID" ]; then
        echo "Deleting workspace $WORKSPACE_ID..."
        aws grafana delete-workspace --workspace-id "$WORKSPACE_ID"
    fi

    if [ -n "$ROLE_NAME" ]; then
        echo "Detaching policies from role $ROLE_NAME..."
        if [ -n "$POLICY_ARN" ]; then
            aws iam detach-role-policy --role-name "$ROLE_NAME" --policy-arn
 "$POLICY_ARN"
        fi

        echo "Deleting role $ROLE_NAME..."
        aws iam delete-role --role-name "$ROLE_NAME"
    fi

    if [ -n "$POLICY_ARN" ]; then
        echo "Deleting policy..."
        aws iam delete-policy --policy-arn "$POLICY_ARN"
    fi

    # Clean up JSON files
    rm -f trust-policy.json cloudwatch-policy.json

    echo "Cleanup completed. See $LOG_FILE for details."
}

# Generate a random identifier for resource names
RANDOM_ID=$(openssl rand -hex 4)
WORKSPACE_NAME="GrafanaWorkspace-${RANDOM_ID}"
ROLE_NAME="GrafanaWorkspaceRole-${RANDOM_ID}"

echo "Using workspace name: $WORKSPACE_NAME"
echo "Using role name: $ROLE_NAME"
```

```
# Step 1: Get AWS account ID
echo "Getting AWS account ID..."
ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
check_error "$ACCOUNT_ID" "get-caller-identity"
echo "AWS Account ID: $ACCOUNT_ID"

# Step 2: Create IAM role for Grafana workspace
echo "Creating IAM role for Grafana workspace..."

# Create trust policy document
cat > trust-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "grafana.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create IAM role
ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$ROLE_NAME" \
  --assume-role-policy-document file://trust-policy.json \
  --description "Role for Amazon Managed Grafana workspace")

check_error "$ROLE_OUTPUT" "create-role"
aws iam tag-role --role-name "$ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=amazon-managed-grafana-gs
echo "IAM role created successfully"

# Extract role ARN
ROLE_ARN=$(echo "$ROLE_OUTPUT" | grep -o '"Arn": "[^"]*' | cut -d'"' -f4)
echo "Role ARN: $ROLE_ARN"

# Attach policies to the role
echo "Attaching policies to the role..."
```

```
# CloudWatch policy
cat > cloudwatch-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:GetMetricData"
      ],
      "Resource": "*"
    }
  ]
}
EOF

POLICY_OUTPUT=$(aws iam create-policy \
  --policy-name "GrafanaCloudWatchPolicy-${RANDOM_ID}" \
  --policy-document file://cloudwatch-policy.json)

check_error "$POLICY_OUTPUT" "create-policy"

POLICY_ARN=$(echo "$POLICY_OUTPUT" | grep -o '"Arn": "[^"]*' | cut -d'"' -f4)
echo "CloudWatch policy ARN: $POLICY_ARN"

ATTACH_OUTPUT=$(aws iam attach-role-policy \
  --role-name "$ROLE_NAME" \
  --policy-arn "$POLICY_ARN")

check_error "$ATTACH_OUTPUT" "attach-role-policy"
echo "CloudWatch policy attached to role"

# Step 3: Create the Grafana workspace
echo "Creating Amazon Managed Grafana workspace..."
WORKSPACE_OUTPUT=$(aws grafana create-workspace \
  --workspace-name "$WORKSPACE_NAME" \
  --authentication-providers "SAML" \
  --permission-type "CUSTOMER_MANAGED" \
  --account-access-type "CURRENT_ACCOUNT" \
```

```
  --workspace-role-arn "$ROLE_ARN" \
  --workspace-data-sources "CLOUDWATCH" "PROMETHEUS" "XRAY" \
  --grafana-version "10.4" \
  --tags Environment=Development,project=doc-smith,tutorial=amazon-managed-
grafana-gs)

check_error "$WORKSPACE_OUTPUT" "create-workspace"

echo "Workspace creation initiated:"
echo "$WORKSPACE_OUTPUT"

# Extract workspace ID
WORKSPACE_ID=$(echo "$WORKSPACE_OUTPUT" | grep -o '"id": "[^"]*' | cut -d'"' -f4)

if [ -z "$WORKSPACE_ID" ]; then
    echo "ERROR: Failed to extract workspace ID from output"
    exit 1
fi

echo "Workspace ID: $WORKSPACE_ID"

# Step 4: Wait for workspace to become active
echo "Waiting for workspace to become active. This may take several minutes..."
ACTIVE=false
MAX_ATTEMPTS=30
ATTEMPT=0

while [ $ACTIVE = false ] && [ $ATTEMPT -lt $MAX_ATTEMPTS ]; do
    ATTEMPT=$((ATTEMPT+1))
    echo "Checking workspace status (attempt $ATTEMPT of $MAX_ATTEMPTS)..."

    DESCRIBE_OUTPUT=$(aws grafana describe-workspace --workspace-id
 "$WORKSPACE_ID")
    check_error "$DESCRIBE_OUTPUT" "describe-workspace"

    STATUS=$(echo "$DESCRIBE_OUTPUT" | grep -o '"status": "[^"]*' | cut -d'"' -
f4)
    echo "Current status: $STATUS"

    if [ "$STATUS" = "ACTIVE" ]; then
        ACTIVE=true
        echo "Workspace is now ACTIVE"
    elif [ "$STATUS" = "FAILED" ]; then
        echo "ERROR: Workspace creation failed"
```

```
        cleanup_on_error
        exit 1
    else
        echo "Workspace is still being created. Waiting 30 seconds..."
        sleep 30
    fi
done

if [ $ACTIVE = false ]; then
    echo "ERROR: Workspace did not become active within the expected time"
    cleanup_on_error
    exit 1
fi

# Extract workspace endpoint URL
WORKSPACE_URL=$(echo "$DESCRIBE_OUTPUT" | grep -o '"endpoint": "[^"]*' | cut -
d'"' -f4)
echo "Workspace URL: https://$WORKSPACE_URL"

# Step 5: Display workspace information
echo ""
echo "==========================================="
echo "WORKSPACE INFORMATION"
echo "==========================================="
echo "Workspace ID: $WORKSPACE_ID"
echo "Workspace URL: https://$WORKSPACE_URL"
echo "Workspace Name: $WORKSPACE_NAME"
echo "IAM Role: $ROLE_NAME"
echo ""
echo "Note: Since SAML authentication is used, you need to configure SAML
 settings"
echo "using the AWS Management Console or the update-workspace-authentication
 command."
echo "==========================================="

# Step 6: Prompt for cleanup
echo ""
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Resources created:"
echo "- Amazon Managed Grafana workspace: $WORKSPACE_ID"
echo "- IAM Role: $ROLE_NAME"
echo "- IAM Policy: GrafanaCloudWatchPolicy-${RANDOM_ID}"
```

```
echo ""
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "$CLEANUP_CHOICE" =~ ^[Yy] ]]; then
    echo "Cleaning up resources..."

    echo "Deleting workspace $WORKSPACE_ID..."
    DELETE_OUTPUT=$(aws grafana delete-workspace --workspace-id "$WORKSPACE_ID")
    check_error "$DELETE_OUTPUT" "delete-workspace"

    echo "Waiting for workspace to be deleted..."
    DELETED=false
    ATTEMPT=0

    while [ $DELETED = false ] && [ $ATTEMPT -lt $MAX_ATTEMPTS ]; do
        ATTEMPT=$((ATTEMPT+1))
        echo "Checking deletion status (attempt $ATTEMPT of $MAX_ATTEMPTS)..."

        if aws grafana describe-workspace --workspace-id "$WORKSPACE_ID" 2>&1 |
 grep -i "not found\|does not exist" > /dev/null; then
            DELETED=true
            echo "Workspace has been deleted"
        else
            echo "Workspace is still being deleted. Waiting 30 seconds..."
            sleep 30
        fi
    done

    if [ $DELETED = false ]; then
        echo "WARNING: Workspace deletion is taking longer than expected. It may
 still be in progress."
    fi

    # Detach policy from role
    echo "Detaching policy from role..."
    aws iam detach-role-policy \
      --role-name "$ROLE_NAME" \
      --policy-arn "$POLICY_ARN"

    # Delete policy
    echo "Deleting IAM policy..."
    aws iam delete-policy \
      --policy-arn "$POLICY_ARN"
```

```
    # Delete role
    echo "Deleting IAM role..."
    aws iam delete-role \
      --role-name "$ROLE_NAME"

    # Clean up JSON files
    rm -f trust-policy.json cloudwatch-policy.json

    echo "Cleanup completed"
else
    echo "Skipping cleanup. Resources will remain in your AWS account."
fi

echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreatePolicy
  - CreateRole
  - CreateWorkspace
  - DeletePolicy
  - DeleteRole
  - DeleteWorkspace
  - DescribeWorkspace
  - DetachRolePolicy
  - GetCallerIdentity

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione di un dashboard di monitoraggio con il nome della funzione come variabile

L'esempio di codice seguente mostra come:

- Crea un pannello di controllo CloudWatch
- Aggiungi widget metrici Lambda con una variabile con nome di funzione
- Verifica il pannello di controllo
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Script to create a CloudWatch dashboard with Lambda function name as a variable
# This script creates a CloudWatch dashboard that allows you to switch between
 different Lambda functions

# Set up logging with secure permissions
LOG_FILE="${HOME}/.cloudwatch-dashboard-script.log"
touch "$LOG_FILE" && chmod 600 "$LOG_FILE"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "$(date): Starting CloudWatch dashboard creation script"

# Security: Set strict error handling
set -uo pipefail
trap 'handle_error "Script failed at line $LINENO"' ERR

# Function to handle errors
handle_error() {
    local error_msg="${1:-Unknown error}"
    echo "ERROR: $error_msg" >&2
    echo "Resources created:"
    echo "- CloudWatch Dashboard: LambdaMetricsDashboard"
    echo ""
```

```
    echo "========================================="
    echo "CLEANUP CONFIRMATION"
    echo "========================================="
    echo "An error occurred. Proceeding with automatic cleanup..."

    echo "Cleaning up resources..."
    aws cloudwatch delete-dashboards --dashboard-names LambdaMetricsDashboard 2>/
dev/null || true

    # Clean up temporary files
    if [ -n "${TEMP_DIR:-}" ] && [ -d "$TEMP_DIR" ]; then
        rm -rf "$TEMP_DIR"
    fi
    rm -f dashboard-body.json

    echo "Cleanup complete."
    exit 1
}

# Security: Validate AWS CLI is installed
if ! command -v aws &> /dev/null; then
    handle_error "AWS CLI is not installed. Please install it and try again."
fi

# Check if AWS CLI is installed and configured
echo "Checking AWS CLI configuration..."
if ! aws sts get-caller-identity > /dev/null 2>&1; then
    handle_error "AWS CLI is not properly configured. Please configure it with
 'aws configure' and try again."
fi

# Get the current region securely
REGION=$(aws configure get region 2>/dev/null || echo "")
if [ -z "$REGION" ]; then
    REGION="us-east-1"
    echo "No region found in AWS config, defaulting to $REGION"
fi
echo "Using region: $REGION"

# Validate region format
if ! [[ "$REGION" =~ ^[a-z]{2}-[a-z]+-[0-9]{1}$ ]]; then
    handle_error "Invalid AWS region format: $REGION"
fi
```

```
# Check if there are any Lambda functions in the account
echo "Checking for Lambda functions..."
LAMBDA_FUNCTIONS=$(aws lambda list-functions --region "$REGION" --query
 "Functions[*].FunctionName" --output text 2>/dev/null || echo "")

if [ -z "$LAMBDA_FUNCTIONS" ]; then
    echo "No Lambda functions found in your account. Creating a simple test
 function..."

    # Create a temporary directory for Lambda function code with secure
 permissions
    TEMP_DIR=$(mktemp -d)
    chmod 700 "$TEMP_DIR"
    trap 'rm -rf "$TEMP_DIR"' EXIT

    # Create a simple Lambda function
    cat > "$TEMP_DIR/index.js" << 'EOF'
exports.handler = async (event) => {
    console.log('Event:', JSON.stringify(event, null, 2));
    return {
        statusCode: 200,
        body: JSON.stringify('Hello from Lambda!'),
    };
};
EOF

    # Zip the function code
    if ! cd "$TEMP_DIR"; then
        handle_error "Failed to change to temporary directory"
    fi

    if ! zip -q function.zip index.js; then
        handle_error "Failed to create zip file"
    fi

    # Create a role for the Lambda function with restricted trust policy
    ROLE_NAME="LambdaDashboardTestRole-$(date +%s)"
    TRUST_POLICY='{"Version":"2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":
{"Service":"lambda.amazonaws.com"},"Action":"sts:AssumeRole"}]}'

    if ! ROLE_ARN=$(aws iam create-role \
        --role-name "$ROLE_NAME" \
        --assume-role-policy-document "$TRUST_POLICY" \
```

```
        --tags Key=project,Value=doc-smith Key=tutorial,Value=cloudwatch-
dynamicdash \
        --query "Role.Arn" \
        --output text 2>/dev/null); then
        handle_error "Failed to create IAM role for Lambda function"
    fi

    echo "Waiting for role to be available..."
    sleep 10

    # Attach basic Lambda execution policy
    if ! aws iam attach-role-policy \
        --role-name "$ROLE_NAME" \
        --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"; then
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
        handle_error "Failed to attach policy to IAM role"
    fi

    # Create the Lambda function
    FUNCTION_NAME="DashboardTestFunction-$(date +%s)"
    if ! aws lambda create-function \
        --function-name "$FUNCTION_NAME" \
        --runtime nodejs18.x \
        --role "$ROLE_ARN" \
        --handler index.handler \
        --zip-file fileb://function.zip \
        --tags project=doc-smith,tutorial=cloudwatch-dynamicdash \
        --region "$REGION" > /dev/null 2>&1; then
        aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null || true
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
        handle_error "Failed to create Lambda function"
    fi

    # Invoke the function to generate some metrics
    echo "Invoking Lambda function to generate metrics..."
    for i in {1..5}; do
        aws lambda invoke --function-name "$FUNCTION_NAME" --payload '{}' /dev/
null --region "$REGION" > /dev/null 2>&1 || true
        sleep 1
    done
```

```
    # Go back to original directory
    cd - > /dev/null

    # Set the function name for the dashboard
    DEFAULT_FUNCTION="$FUNCTION_NAME"
else
    # Use the first Lambda function as default
    DEFAULT_FUNCTION=$(echo "$LAMBDA_FUNCTIONS" | awk '{print $1}')
    echo "Found Lambda functions. Using $DEFAULT_FUNCTION as default."
    FUNCTION_NAME=""
    ROLE_NAME=""
fi

# Create a dashboard with Lambda metrics and a function name variable
echo "Creating CloudWatch dashboard with Lambda function name variable..."

# Create a JSON file for the dashboard body with secure permissions
DASHBOARD_JSON="dashboard-body-$$.json"
touch "$DASHBOARD_JSON" && chmod 600 "$DASHBOARD_JSON"

# Escape special characters in region and function name for JSON
REGION_ESCAPED=$(printf '%s\n' "$REGION" | sed 's:[\/&]:\\&:g')
FUNCTION_ESCAPED=$(printf '%s\n' "$DEFAULT_FUNCTION" | sed 's:[\/&]:\\&:g')

cat > "$DASHBOARD_JSON" << EOF
{
  "widgets": [
    {
      "type": "metric",
      "x": 0,
      "y": 0,
      "width": 12,
      "height": 6,
      "properties": {
        "metrics": [
          [ "AWS/Lambda", "Invocations", "FunctionName", "\${FunctionName}" ],
          [ ".", "Errors", ".", "." ],
          [ ".", "Throttles", ".", "." ]
        ],
        "view": "timeSeries",
        "stacked": false,
        "region": "$REGION_ESCAPED",
        "title": "Lambda Function Metrics for \${FunctionName}",
```

```
        "period": 300
      }
    },
    {
      "type": "metric",
      "x": 0,
      "y": 6,
      "width": 12,
      "height": 6,
      "properties": {
        "metrics": [
          [ "AWS/Lambda", "Duration", "FunctionName", "\${FunctionName}",
 { "stat": "Average" } ]
        ],
        "view": "timeSeries",
        "stacked": false,
        "region": "$REGION_ESCAPED",
        "title": "Duration for \${FunctionName}",
        "period": 300
      }
    },
    {
      "type": "metric",
      "x": 12,
      "y": 0,
      "width": 12,
      "height": 6,
      "properties": {
        "metrics": [
          [ "AWS/Lambda", "ConcurrentExecutions", "FunctionName", "\
${FunctionName}" ]
        ],
        "view": "timeSeries",
        "stacked": false,
        "region": "$REGION_ESCAPED",
        "title": "Concurrent Executions for \${FunctionName}",
        "period": 300
      }
    }
  ],
  "periodOverride": "auto",
  "variables": [
    {
      "type": "property",
```

```
      "id": "FunctionName",
      "property": "FunctionName",
      "label": "Lambda Function",
      "inputType": "select",
      "values": [
        {
          "value": "$FUNCTION_ESCAPED",
          "label": "$FUNCTION_ESCAPED"
        }
      ]
    }
  ]
}
EOF

# Validate JSON before sending
if ! jq empty "$DASHBOARD_JSON" 2>/dev/null; then
    handle_error "Invalid JSON generated for dashboard"
fi

# Create the dashboard using the JSON file
DASHBOARD_NAME="LambdaMetricsDashboard-$(date +%s)"
if ! DASHBOARD_RESULT=$(aws cloudwatch put-dashboard \
    --dashboard-name "$DASHBOARD_NAME" \
    --dashboard-body file://"$DASHBOARD_JSON" \
    --region "$REGION" 2>&1); then
    # If we created resources, clean them up
    if [ -n "${FUNCTION_NAME:-}" ]; then
        aws lambda delete-function --function-name "$FUNCTION_NAME" --region
 "$REGION" 2>/dev/null || true
        aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null || true
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi
    handle_error "Failed to create CloudWatch dashboard."
fi

# Display any validation messages but continue
if echo "$DASHBOARD_RESULT" | grep -q "DashboardValidationMessages"; then
    echo "Dashboard created with validation messages:"
    echo "$DASHBOARD_RESULT"
```

```
    echo "These validation messages are warnings and the dashboard should still
 function."
else
    echo "Dashboard created successfully!"
fi

# Verify the dashboard was created
echo "Verifying dashboard creation..."
if ! DASHBOARD_INFO=$(aws cloudwatch get-dashboard --dashboard-name
 "$DASHBOARD_NAME" --region "$REGION" 2>&1); then
    # If we created resources, clean them up
    if [ -n "${FUNCTION_NAME:-}" ]; then
        aws lambda delete-function --function-name "$FUNCTION_NAME" --region
 "$REGION" 2>/dev/null || true
        aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null || true
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi
    handle_error "Failed to verify dashboard creation."
fi

echo "Dashboard verification successful!"
echo "Dashboard details:"
echo "$DASHBOARD_INFO"

# List all dashboards to confirm
echo "Listing all dashboards:"
if ! DASHBOARDS=$(aws cloudwatch list-dashboards --region "$REGION" 2>&1); then
    # If we created resources, clean them up
    if [ -n "${FUNCTION_NAME:-}" ]; then
        aws lambda delete-function --function-name "$FUNCTION_NAME" --region
 "$REGION" 2>/dev/null || true
        aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null || true
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi
    handle_error "Failed to list dashboards."
fi
echo "$DASHBOARDS"
```

```
# Show instructions for accessing the dashboard
echo ""
echo "Dashboard created successfully! To access it:"
echo "1. Open the CloudWatch console at https://console.aws.amazon.com/
cloudwatch/"
echo "2. In the navigation pane, choose Dashboards"
echo "3. Select $DASHBOARD_NAME"
echo "4. You should see a dropdown menu labeled 'Lambda Function' at the top of
 the dashboard"
echo "5. Use this dropdown to select different Lambda functions and see their
 metrics"
echo ""

# Create a list of resources for cleanup
RESOURCES=("- CloudWatch Dashboard: $DASHBOARD_NAME")
if [ -n "${FUNCTION_NAME:-}" ]; then
    RESOURCES+=("- Lambda Function: $FUNCTION_NAME")
    RESOURCES+=("- IAM Role: $ROLE_NAME")
fi

# Prompt for cleanup with automatic yes
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Resources created:"
for resource in "${RESOURCES[@]}"; do
    echo "$resource"
done
echo ""
echo "Proceeding with automatic cleanup..."

CLEANUP_CHOICE="y"

if [[ "${CLEANUP_CHOICE,,}" == "y" ]]; then
    echo "Cleaning up resources..."

    # Delete the dashboard
    if aws cloudwatch delete-dashboards --dashboard-names "$DASHBOARD_NAME" --
region "$REGION" 2>/dev/null; then
        echo "Dashboard deleted successfully."
    else
        echo "WARNING: Failed to delete dashboard. You may need to delete it
 manually."
    fi
```

```
    # If we created a Lambda function, delete it and its role
    if [ -n "${FUNCTION_NAME:-}" ]; then
        echo "Deleting Lambda function..."
        if aws lambda delete-function --function-name "$FUNCTION_NAME" --region
 "$REGION" 2>/dev/null; then
            echo "Lambda function deleted successfully."
        else
            echo "WARNING: Failed to delete Lambda function. You may need to
 delete it manually."
        fi

        echo "Detaching role policy..."
        if aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null; then
            echo "Role policy detached successfully."
        else
            echo "WARNING: Failed to detach role policy. You may need to detach
 it manually."
        fi

        echo "Deleting IAM role..."
        if aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null; then
            echo "IAM role deleted successfully."
        else
            echo "WARNING: Failed to delete IAM role. You may need to delete it
 manually."
        fi
    fi

    # Clean up the JSON file
    rm -f "$DASHBOARD_JSON"

    echo "Cleanup complete."
fi

echo "Script completed successfully!"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy

- CreateFunction
- CreateRole
- DeleteDashboards
- DeleteFunction
- DeleteRole
- DetachRolePolicy
- GetCallerIdentity
- GetDashboard
- Invoke
- ListDashboards
- ListFunctions
- PutDashboard

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione della tua prima funzione serverless

L'esempio di codice seguente mostra come:

- Crea un ruolo IAM per Lambda
- Crea codice funzionale
- Creazione di una funzione Lambda
- Testa la tua funzione Lambda
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash
# AWS Lambda - Create Your First Function
# This script creates a Lambda function, invokes it with a test event,
# views CloudWatch logs, and cleans up all resources.
#
# Source: https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html
#
# Resources created:
#   - IAM role (Lambda execution role with basic logging permissions)
#   - Lambda function (Python 3.13 or Node.js 22.x runtime)
#   - CloudWatch log group (created automatically by Lambda on invocation)

set -eE -o pipefail

################################################################################
# Setup
################################################################################

UNIQUE_ID=$(head -c 8 /dev/urandom | od -An -tx1 | tr -d ' ')
FUNCTION_NAME="my-lambda-function-${UNIQUE_ID}"
ROLE_NAME="lambda-execution-role-${UNIQUE_ID}"
LOG_GROUP_NAME="/aws/lambda/${FUNCTION_NAME}"

TEMP_DIR=$(mktemp -d)
readonly TEMP_DIR
LOG_FILE="${TEMP_DIR}/lambda-gettingstarted.log"

exec > >(tee -a "$LOG_FILE") 2>&1

declare -a CREATED_RESOURCES
```

```
################################################################################
# Helper functions
################################################################################

cleanup_resources() {
    # Disable error trap to prevent recursion during cleanup
    trap - ERR
    set +eE

    echo ""
    echo "Cleaning up resources..."
    echo ""

    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        local RESOURCE="${CREATED_RESOURCES[$i]}"
        local TYPE="${RESOURCE%%:*}"
        local NAME="${RESOURCE#*:}"

        case "$TYPE" in
            log-group)
                echo "Deleting CloudWatch log group: ${NAME}"
                aws logs delete-log-group \
                    --log-group-name "$NAME" 2>&1 || echo "  WARNING: Could not
 delete log group ${NAME}."
                ;;
            lambda-function)
                echo "Deleting Lambda function: ${NAME}"
                aws lambda delete-function \
                    --function-name "$NAME" 2>&1 || echo "  WARNING: Could not
 delete Lambda function ${NAME}."
                echo "  Waiting for function deletion to complete..."
                local DELETE_WAIT=0
                while aws lambda get-function --function-name "$NAME" > /dev/null
 2>&1; do
                    sleep 2
                    DELETE_WAIT=$((DELETE_WAIT + 2))
                    if [ "$DELETE_WAIT" -ge 60 ]; then
                        echo "  WARNING: Timed out waiting for function
 deletion."
                        break
                    fi
                done
                ;;
            iam-role-policy)
```

```
                local ROLE_PART="${NAME%%|*}"
                local POLICY_PART="${NAME#*|}"
                echo "Detaching policy from role: ${ROLE_PART}"
                aws iam detach-role-policy \
                    --role-name "$ROLE_PART" \
                    --policy-arn "$POLICY_PART" 2>&1 || echo "  WARNING: Could
 not detach policy from role ${ROLE_PART}."
                ;;
            iam-role)
                echo "Deleting IAM role: ${NAME}"
                aws iam delete-role \
                    --role-name "$NAME" 2>&1 || echo "  WARNING: Could not delete
 IAM role ${NAME}."
                ;;
        esac
    done

    if [ -d "$TEMP_DIR" ]; then
        rm -rf "$TEMP_DIR"
    fi

    echo ""
    echo "Cleanup complete."
}

handle_error() {
    echo ""
    echo "========================================="
    echo "ERROR: Script failed at $1"
    echo "========================================="
    echo ""
    if [ ${#CREATED_RESOURCES[@]} -gt 0 ]; then
        echo "Attempting to clean up ${#CREATED_RESOURCES[@]} resource(s)..."
        cleanup_resources
    fi
    exit 1
}

trap 'handle_error "line $LINENO"' ERR

wait_for_resource() {
    local DESCRIPTION="$1"
    local COMMAND="$2"
    local TARGET_VALUE="$3"
```

```
    local TIMEOUT=300
    local ELAPSED=0
    local INTERVAL=5

    echo "Waiting for ${DESCRIPTION}..."
    while true; do
        local RESULT
        RESULT=$(eval "$COMMAND" 2>&1) || true
        if echo "$RESULT" | grep -q "$TARGET_VALUE"; then
            echo "  ${DESCRIPTION} is ready."
            return 0
        fi
        if [ "$ELAPSED" -ge "$TIMEOUT" ]; then
            echo "ERROR: Timed out waiting for ${DESCRIPTION} after ${TIMEOUT}
 seconds."
            return 1
        fi
        sleep "$INTERVAL"
        ELAPSED=$((ELAPSED + INTERVAL))
    done
}

validate_input() {
    local input="$1"
    local pattern="$2"
    if ! [[ "$input" =~ $pattern ]]; then
        echo "ERROR: Invalid input: $input"
        return 1
    fi
    return 0
}

################################################################################
# Region pre-check
################################################################################

CONFIGURED_REGION=$(aws configure get region 2>/dev/null || true)
if [ -z "$CONFIGURED_REGION" ] && [ -z "$AWS_DEFAULT_REGION" ] && [ -z
 "$AWS_REGION" ]; then
    echo "ERROR: No AWS region configured."
    echo "Run 'aws configure set region <region>' or export AWS_DEFAULT_REGION."
    exit 1
fi
```

```
################################################################################
# Runtime selection
################################################################################

echo ""
echo "=========================================="
echo "AWS Lambda - Create Your First Function"
echo "=========================================="
echo ""
echo "Select a runtime for your Lambda function:"
echo "  1) Python 3.13"
echo "  2) Node.js 22.x"
echo ""
echo "Using default: Python 3.13"
RUNTIME_CHOICE="1"

case "$RUNTIME_CHOICE" in
    1)
        RUNTIME="python3.13"
        HANDLER="lambda_function.lambda_handler"
        CODE_FILE="lambda_function.py"
        cat > "${TEMP_DIR}/${CODE_FILE}" << 'PYTHON_EOF'
import json
import logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
def lambda_handler(event, context):
    if not isinstance(event, dict) or 'length' not in event or 'width' not in
 event:
        raise ValueError('Event must contain length and width')
    try:
        length = float(event['length'])
        width = float(event['width'])
        if length < 0 or width < 0:
            raise ValueError('Length and width must be non-negative')
        area = calculate_area(length, width)
        print(f'The area is {area}')
        logger.info(f'CloudWatch logs group: {context.log_group_name}')
        return json.dumps({'area': area})
    except (TypeError, ValueError) as e:
        logger.error(f'Error processing input: {str(e)}')
        raise
def calculate_area(length, width):
    return length * width
```

```
PYTHON_EOF
        echo "Selected runtime: Python 3.13"
        ;;
    2)
        RUNTIME="nodejs22.x"
        HANDLER="index.handler"
        CODE_FILE="index.mjs"
        cat > "${TEMP_DIR}/${CODE_FILE}" << 'NODEJS_EOF'
export const handler = async (event, context) => {
  if (!event || typeof event.length !== 'number' || typeof event.width !==
 'number') {
    throw new Error('Event must contain numeric length and width');
  }
  if (event.length < 0 || event.width < 0) {
    throw new Error('Length and width must be non-negative');
  }
  const area = event.length * event.width;
  console.log(`The area is ${area}`);
  console.log('CloudWatch log group: ', context.logGroupName);
  return JSON.stringify({area});
};
NODEJS_EOF
        echo "Selected runtime: Node.js 22.x"
        ;;
    *)
        echo "ERROR: Invalid choice. Please enter 1 or 2."
        exit 1
        ;;
esac

# ============================================================================
# Step 1: Create IAM execution role
# ============================================================================

echo ""
echo "=========================================="
echo "Step 1: Create IAM execution role"
echo "=========================================="
echo ""

TRUST_POLICY='{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'

echo "Creating IAM role: ${ROLE_NAME}"
ROLE_OUTPUT=$(aws iam create-role \
    --role-name "$ROLE_NAME" \
    --assume-role-policy-document "$TRUST_POLICY" \
    --query 'Role.Arn' \
    --output text 2>&1)

if ! validate_input "$ROLE_OUTPUT" "^arn:aws:iam::[0-9]+:role/"; then
    echo "ERROR: Failed to create IAM role"
    exit 1
fi

echo "$ROLE_OUTPUT"
ROLE_ARN="$ROLE_OUTPUT"
CREATED_RESOURCES+=("iam-role:${ROLE_NAME}")
echo "Role ARN: ${ROLE_ARN}"

aws iam tag-role \
    --role-name "$ROLE_NAME" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=lambda-gettingstarted

echo ""
echo "Attaching AWSLambdaBasicExecutionRole policy..."
aws iam attach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>&1
CREATED_RESOURCES+=("iam-role-policy:${ROLE_NAME}|arn:aws:iam::aws:policy/
service-role/AWSLambdaBasicExecutionRole")
echo "Policy attached."

# IAM roles can take a few seconds to propagate
echo "Waiting for IAM role to propagate..."
sleep 10
```

```
################################################################################
# Step 2: Create Lambda function
################################################################################

echo ""
echo "==========================================="
echo "Step 2: Create Lambda function"
echo "==========================================="
echo ""

echo "Creating deployment package..."
ORIGINAL_DIR=$(pwd)
cd "$TEMP_DIR" || exit 1
zip -j function.zip "$CODE_FILE" > /dev/null 2>&1 || {
    echo "ERROR: Failed to create deployment package"
    exit 1
}
cd "$ORIGINAL_DIR" || exit 1

if [ ! -f "${TEMP_DIR}/function.zip" ]; then
    echo "ERROR: Deployment package creation failed"
    exit 1
fi

echo "Creating Lambda function: ${FUNCTION_NAME}"
echo "  Runtime: ${RUNTIME}"
echo "  Handler: ${HANDLER}"
echo ""

CREATE_OUTPUT=$(aws lambda create-function \
    --function-name "$FUNCTION_NAME" \
    --runtime "$RUNTIME" \
    --role "$ROLE_ARN" \
    --handler "$HANDLER" \
    --architectures x86_64 \
    --zip-file "fileb://${TEMP_DIR}/function.zip" \
    --tags project=doc-smith,tutorial=lambda-gettingstarted \
    --query '[FunctionName, FunctionArn, Runtime, State]' \
    --output text 2>&1)

if [ -z "$CREATE_OUTPUT" ]; then
    echo "ERROR: Failed to create Lambda function"
    exit 1
fi
```

```
echo "$CREATE_OUTPUT"
CREATED_RESOURCES+=("lambda-function:${FUNCTION_NAME}")

wait_for_resource "Lambda function to become Active" \
    "aws lambda get-function-configuration --function-name ${FUNCTION_NAME} --
query State --output text" \
    "Active"

################################################################################
# Step 3: Invoke the function
################################################################################

echo ""
echo "============================================"
echo "Step 3: Invoke the function"
echo "============================================"
echo ""

TEST_EVENT='{"length": 6, "width": 7}'
echo "Invoking function with test event: ${TEST_EVENT}"
echo ""

echo "$TEST_EVENT" > "${TEMP_DIR}/test-event.json"

if ! validate_input "$TEST_EVENT" '"length": [0-9]+, "width": [0-9]+'; then
    echo "ERROR: Invalid test event format"
    exit 1
fi

INVOKE_OUTPUT=$(aws lambda invoke \
    --function-name "$FUNCTION_NAME" \
    --payload "fileb://${TEMP_DIR}/test-event.json" \
    --cli-read-timeout 30 \
    "${TEMP_DIR}/response.json" 2>&1)
echo "$INVOKE_OUTPUT"

if [ ! -f "${TEMP_DIR}/response.json" ]; then
    echo "ERROR: No response file generated"
    exit 1
fi

RESPONSE=$(cat "${TEMP_DIR}/response.json")
echo ""
```

```
echo "Function response: ${RESPONSE}"
echo ""

if echo "$INVOKE_OUTPUT" | grep -qi "functionerror"; then
    echo "WARNING: Function returned an error."
fi

################################################################################
# Step 4: View CloudWatch logs
################################################################################

echo ""
echo "========================================="
echo "Step 4: View CloudWatch Logs"
echo "========================================="
echo ""

echo "Log group: ${LOG_GROUP_NAME}"
echo ""

echo "Waiting for CloudWatch logs to be available..."

LOG_STREAMS=""
for i in $(seq 1 6); do
    LOG_STREAMS=$(aws logs describe-log-streams \
        --log-group-name "$LOG_GROUP_NAME" \
        --order-by LastEventTime \
        --descending \
        --query 'logStreams[0].logStreamName' \
        --output text 2>/dev/null) || true
    if [ -n "$LOG_STREAMS" ] && [ "$LOG_STREAMS" != "None" ]; then
        break
    fi
    LOG_STREAMS=""
    sleep 5
done

if [ -n "$LOG_STREAMS" ] && [ "$LOG_STREAMS" != "None" ]; then
    echo "Latest log stream: ${LOG_STREAMS}"
    echo ""
    echo "--- Log events ---"
    LOG_EVENTS=$(aws logs get-log-events \
        --log-group-name "$LOG_GROUP_NAME" \
        --log-stream-name "$LOG_STREAMS" \
```

```
        --query 'events[].message' \
        --output text 2>&1) || true
    echo "$LOG_EVENTS"
    echo "--- End of log events ---"
else
    echo "No log streams found yet. Logs may take a moment to appear."
    echo "You can view them in the CloudWatch console:"
    echo "  Log group: ${LOG_GROUP_NAME}"
fi

CREATED_RESOURCES+=("log-group:${LOG_GROUP_NAME}")

aws logs tag-log-group \
    --log-group-name "$LOG_GROUP_NAME" \
    --tags project=doc-smith,tutorial=lambda-gettingstarted

################################################################################
# Summary and cleanup
################################################################################

echo ""
echo "==========================================="
echo "SUMMARY"
echo "==========================================="
echo ""
echo "Resources created:"
echo "  IAM role:          ${ROLE_NAME}"
echo "  Lambda function:   ${FUNCTION_NAME}"
echo "  CloudWatch logs:   ${LOG_GROUP_NAME}"
echo ""
echo "==========================================="
echo "CLEANUP"
echo "==========================================="
echo ""
echo "Cleaning up all created resources..."
cleanup_resources

echo ""
echo "Done."
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy

- CreateFunction
- CreateRole
- DeleteFunction
- DeleteLogGroup
- DeleteRole
- DescribeLogStreams
- DetachRolePolicy
- GetFunction
- GetFunctionConfiguration
- GetLogEvents
- Invoke

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Inizia a usare i data warehouse serverless utilizzando la CLI

L'esempio di codice seguente mostra come:

- Usa secrets-manager CreateSecret
- Usa secrets-manager DeleteSecret
- Usa secrets-manager GetSecretValue
- Usa redshift CreateNamespace
- Usa redshift CreateWorkgroup
- Usa redshift DeleteNamespace
- Usa iam CreateRole

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon Redshift Serverless Tutorial Script with Secrets Manager (No jq
 dependency)
# This script creates a Redshift Serverless environment, loads sample data, and
 runs queries
# Uses AWS Secrets Manager for secure password management without requiring jq

# Set up logging
LOG_FILE="redshift-serverless-tutorial-v4.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon Redshift Serverless tutorial script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Function to check for errors in command output
check_error() {
  local output=$1
  local cmd=$2

  if echo "$output" | grep -i "error\|exception\|fail" > /dev/null; then
    echo "ERROR: Command failed: $cmd"
    echo "Output: $output"
    cleanup_resources
    exit 1
  fi
}

# Function to generate a secure password that meets Redshift requirements
generate_secure_password() {
  # Redshift password requirements:
```

```
  # - 8-64 characters
  # - At least one uppercase letter
  # - At least one lowercase letter
  # - At least one decimal digit
  # - Can contain printable ASCII characters except /, ", ', \, @, space

  local password=""
  local valid=false
  local attempts=0
  local max_attempts=10

  while [[ "$valid" == false && $attempts -lt $max_attempts ]]; do
    # Generate base password with safe characters
    local base=$(openssl rand -base64 12 | tr -d '/+=' | head -c 12)

    # Ensure we have at least one of each required character type
    local upper=$(echo "ABCDEFGHIJKLMNOPQRSTUVWXYZ" | fold -w1 | shuf -n1)
    local lower=$(echo "abcdefghijklmnopqrstuvwxyz" | fold -w1 | shuf -n1)
    local digit=$(echo "0123456789" | fold -w1 | shuf -n1)
    local special=$(echo "!#$%&*()_+-=[]{}|;:,.<>?" | fold -w1 | shuf -n1)

    # Combine and shuffle
    password="${base}${upper}${lower}${digit}${special}"
    password=$(echo "$password" | fold -w1 | shuf | tr -d '\n')

    # Validate password meets requirements
    if [[ ${#password} -ge 8 && ${#password} -le 64 ]] && \
       [[ "$password" =~ [A-Z] ]] && \
       [[ "$password" =~ [a-z] ]] && \
       [[ "$password" =~ [0-9] ]] && \
       [[ ! "$password" =~ [/\"\'\\@[:space:]] ]]; then
      valid=true
    fi

    ((attempts++))
  done

  if [[ "$valid" == false ]]; then
    echo "ERROR: Failed to generate valid password after $max_attempts attempts"
    exit 1
  fi

  echo "$password"
}
```

```
# Function to create secret in AWS Secrets Manager
create_secret() {
  local secret_name=$1
  local username=$2
  local password=$3
  local description=$4

  echo "Creating secret in AWS Secrets Manager: $secret_name"

  # Create the secret using AWS CLI without jq
  local secret_output=$(aws secretsmanager create-secret \
    --name "$secret_name" \
    --description "$description" \
    --secret-string "{\"username\":\"$username\",\"password\":\"$password\"}" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-serverless
 2>&1)

  if echo "$secret_output" | grep -i "error\|exception\|fail" > /dev/null; then
    echo "ERROR: Failed to create secret: $secret_output"
    return 1
  fi

  echo "Secret created successfully: $secret_name"
  return 0
}

# Function to retrieve password from AWS Secrets Manager
get_password_from_secret() {
  local secret_name=$1

  # Get the secret value and extract password using sed/grep instead of jq
  local secret_value=$(aws secretsmanager get-secret-value \
    --secret-id "$secret_name" \
    --query 'SecretString' \
    --output text 2>/dev/null)

  if [[ $? -eq 0 ]]; then
    # Extract password from JSON using sed
    echo "$secret_value" | sed -n 's/.*"password":"\([^"]*\)".*/\1/p'
  else
    echo ""
  fi
}
```

```
# Function to wait for a resource to be available
wait_for_resource() {
  local resource_type=$1
  local resource_name=$2
  local max_attempts=$3
  local wait_seconds=$4
  local check_cmd=$5

  echo "Waiting for $resource_type $resource_name to be available..."

  for ((i=1; i<=$max_attempts; i++)); do
    local output=$($check_cmd 2>/dev/null)
    local status=$(echo "$output" | grep -o '"Status": "[^"]*' | cut -d'"' -f4 ||
 echo "")

    if [[ "$status" == "AVAILABLE" ]]; then
      echo "$resource_type $resource_name is now available"
      return 0
    fi

    echo "Attempt $i/$max_attempts: $resource_type $resource_name status:
 $status. Waiting $wait_seconds seconds..."
    sleep $wait_seconds
  done

  echo "ERROR: Timed out waiting for $resource_type $resource_name to be
 available"
  return 1
}

# Function to wait for a resource to be deleted
wait_for_resource_deletion() {
  local resource_type=$1
  local resource_name=$2
  local max_attempts=$3
  local wait_seconds=$4
  local check_cmd=$5

  echo "Waiting for $resource_type $resource_name to be deleted..."

  for ((i=1; i<=$max_attempts; i++)); do
    local output=$($check_cmd 2>&1)
```

```
    if echo "$output" | grep -i "not found\|does not exist" > /dev/null; then
      echo "$resource_type $resource_name has been deleted"
      return 0
    fi

    echo "Attempt $i/$max_attempts: $resource_type $resource_name is still being
 deleted. Waiting $wait_seconds seconds..."
    sleep $wait_seconds
  done

  echo "ERROR: Timed out waiting for $resource_type $resource_name to be deleted"
  return 1
}

# Function to clean up resources
cleanup_resources() {
  echo ""
  echo "========================================="
  echo "CLEANUP CONFIRMATION"
  echo "========================================="
  echo "The following resources were created:"
  echo "- Redshift Serverless Workgroup: $WORKGROUP_NAME"
  echo "- Redshift Serverless Namespace: $NAMESPACE_NAME"
  echo "- IAM Role: $ROLE_NAME"
  echo "- Secrets Manager Secret: $SECRET_NAME"
  echo ""
  echo "Do you want to clean up all created resources? (y/n): "
  read -r CLEANUP_CHOICE

  if [[ "${CLEANUP_CHOICE,,}" == "y" ]]; then
    echo "Cleaning up resources..."

    # Delete the workgroup
    echo "Deleting Redshift Serverless workgroup $WORKGROUP_NAME..."
    WORKGROUP_DELETE_OUTPUT=$(aws redshift-serverless delete-workgroup --
workgroup-name "$WORKGROUP_NAME" 2>&1)
    echo "$WORKGROUP_DELETE_OUTPUT"

    # Wait for workgroup to be deleted before deleting namespace
    wait_for_resource_deletion "workgroup" "$WORKGROUP_NAME" 20 30 "aws redshift-
serverless get-workgroup --workgroup-name $WORKGROUP_NAME"

    # Delete the namespace
    echo "Deleting Redshift Serverless namespace $NAMESPACE_NAME..."
```

```
    NAMESPACE_DELETE_OUTPUT=$(aws redshift-serverless delete-namespace --
namespace-name "$NAMESPACE_NAME" 2>&1)
    echo "$NAMESPACE_DELETE_OUTPUT"

    # Wait for namespace to be deleted
    wait_for_resource_deletion "namespace" "$NAMESPACE_NAME" 20 30 "aws redshift-
serverless get-namespace --namespace-name $NAMESPACE_NAME"

    # Delete the IAM role policy
    echo "Deleting IAM role policy..."
    POLICY_DELETE_OUTPUT=$(aws iam delete-role-policy --role-name "$ROLE_NAME" --
policy-name S3Access 2>&1)
    echo "$POLICY_DELETE_OUTPUT"

    # Delete the IAM role
    echo "Deleting IAM role $ROLE_NAME..."
    ROLE_DELETE_OUTPUT=$(aws iam delete-role --role-name "$ROLE_NAME" 2>&1)
    echo "$ROLE_DELETE_OUTPUT"

    # Delete the secret
    echo "Deleting Secrets Manager secret $SECRET_NAME..."
    SECRET_DELETE_OUTPUT=$(aws secretsmanager delete-secret --secret-id
 "$SECRET_NAME" --force-delete-without-recovery 2>&1)
    echo "$SECRET_DELETE_OUTPUT"

    echo "Cleanup completed."
  else
    echo "Cleanup skipped. Resources will remain in your AWS account."
  fi
}

# Check if required tools are available
if ! command -v openssl &> /dev/null; then
    echo "ERROR: openssl is required but not installed. Please install openssl to
 continue."
    exit 1
fi

# Generate unique names for resources
RANDOM_SUFFIX=$(cat /dev/urandom | tr -dc 'a-z0-9' | head -c 6)
NAMESPACE_NAME="rs-namespace-${RANDOM_SUFFIX}"
WORKGROUP_NAME="rs-workgroup-${RANDOM_SUFFIX}"
ROLE_NAME="RedshiftServerlessS3Role-${RANDOM_SUFFIX}"
SECRET_NAME="redshift-serverless-admin-${RANDOM_SUFFIX}"
```

```
DB_NAME="dev"
ADMIN_USERNAME="admin"

# Generate secure password
echo "Generating secure password..."
ADMIN_PASSWORD=$(generate_secure_password)

# Create secret in AWS Secrets Manager
create_secret "$SECRET_NAME" "$ADMIN_USERNAME" "$ADMIN_PASSWORD" "Admin
 credentials for Redshift Serverless namespace $NAMESPACE_NAME"
if [[ $? -ne 0 ]]; then
  echo "ERROR: Failed to create secret in AWS Secrets Manager"
  exit 1
fi

# Track created resources
CREATED_RESOURCES=()

echo "Using the following resource names:"
echo "- Namespace: $NAMESPACE_NAME"
echo "- Workgroup: $WORKGROUP_NAME"
echo "- IAM Role: $ROLE_NAME"
echo "- Secret: $SECRET_NAME"
echo "- Database: $DB_NAME"
echo "- Admin Username: $ADMIN_USERNAME"
echo "- Admin Password: [STORED IN SECRETS MANAGER]"

# Step 1: Create IAM role for S3 access
echo "Creating IAM role for Redshift Serverless S3 access..."

# Create trust policy document
cat > redshift-trust-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift-serverless.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
EOF

# Create S3 access policy document
cat > redshift-s3-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::redshift-downloads",
        "arn:aws:s3:::redshift-downloads/*"
      ]
    }
  ]
}
EOF

# Create IAM role
echo "Creating IAM role $ROLE_NAME..."
ROLE_OUTPUT=$(aws iam create-role --role-name "$ROLE_NAME" --assume-role-policy-
document file://redshift-trust-policy.json 2>&1)
echo "$ROLE_OUTPUT"
check_error "$ROLE_OUTPUT" "aws iam create-role"
aws iam tag-role --role-name "$ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-serverless
CREATED_RESOURCES+=("IAM Role: $ROLE_NAME")

# Attach S3 policy to the role
echo "Attaching S3 access policy to role $ROLE_NAME..."
POLICY_OUTPUT=$(aws iam put-role-policy --role-name "$ROLE_NAME" --policy-name
 S3Access --policy-document file://redshift-s3-policy.json 2>&1)
echo "$POLICY_OUTPUT"
check_error "$POLICY_OUTPUT" "aws iam put-role-policy"

# Get the role ARN
ROLE_ARN=$(aws iam get-role --role-name "$ROLE_NAME" --query 'Role.Arn' --output
 text)
echo "Role ARN: $ROLE_ARN"
```

```
# Step 2: Create a namespace
echo "Creating Redshift Serverless namespace $NAMESPACE_NAME..."
NAMESPACE_OUTPUT=$(aws redshift-serverless create-namespace \
  --namespace-name "$NAMESPACE_NAME" \
  --admin-username "$ADMIN_USERNAME" \
  --admin-user-password "$ADMIN_PASSWORD" \
  --db-name "$DB_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-serverless 2>&1)
echo "$NAMESPACE_OUTPUT"
check_error "$NAMESPACE_OUTPUT" "aws redshift-serverless create-namespace"
CREATED_RESOURCES+=("Redshift Serverless Namespace: $NAMESPACE_NAME")

# Wait for namespace to be available
wait_for_resource "namespace" "$NAMESPACE_NAME" 10 30 "aws redshift-serverless
 get-namespace --namespace-name $NAMESPACE_NAME"

# Associate IAM role with namespace
echo "Associating IAM role with namespace..."
UPDATE_NAMESPACE_OUTPUT=$(aws redshift-serverless update-namespace \
  --namespace-name "$NAMESPACE_NAME" \
  --iam-roles "$ROLE_ARN" 2>&1)
echo "$UPDATE_NAMESPACE_OUTPUT"
check_error "$UPDATE_NAMESPACE_OUTPUT" "aws redshift-serverless update-namespace"

# Step 3: Create a workgroup
echo "Creating Redshift Serverless workgroup $WORKGROUP_NAME..."
WORKGROUP_OUTPUT=$(aws redshift-serverless create-workgroup \
  --workgroup-name "$WORKGROUP_NAME" \
  --namespace-name "$NAMESPACE_NAME" \
  --base-capacity 8 \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-serverless 2>&1)
echo "$WORKGROUP_OUTPUT"
check_error "$WORKGROUP_OUTPUT" "aws redshift-serverless create-workgroup"
CREATED_RESOURCES+=("Redshift Serverless Workgroup: $WORKGROUP_NAME")

# Wait for workgroup to be available
wait_for_resource "workgroup" "$WORKGROUP_NAME" 20 30 "aws redshift-serverless
 get-workgroup --workgroup-name $WORKGROUP_NAME"

# Get workgroup endpoint
WORKGROUP_ENDPOINT=$(aws redshift-serverless get-workgroup \
  --workgroup-name "$WORKGROUP_NAME" \
  --query 'workgroup.endpoint.address' \
  --output text)
```

```
echo "Workgroup endpoint: $WORKGROUP_ENDPOINT"

# Wait additional time for the endpoint to be fully operational
echo "Waiting for endpoint to be fully operational..."
sleep 60

# Step 4: Create tables for sample data
echo "Creating tables for sample data..."

# Create users table
echo "Creating users table..."
USERS_TABLE_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "CREATE TABLE users(
    userid INTEGER NOT NULL DISTKEY SORTKEY,
    username CHAR(8),
    firstname VARCHAR(30),
    lastname VARCHAR(30),
    city VARCHAR(30),
    state CHAR(2),
    email VARCHAR(100),
    phone CHAR(14),
    likesports BOOLEAN,
    liketheatre BOOLEAN,
    likeconcerts BOOLEAN,
    likejazz BOOLEAN,
    likeclassical BOOLEAN,
    likeopera BOOLEAN,
    likerock BOOLEAN,
    likevegas BOOLEAN,
    likebroadway BOOLEAN,
    likemusicals BOOLEAN
  );" 2>&1)
echo "$USERS_TABLE_OUTPUT"
check_error "$USERS_TABLE_OUTPUT" "aws redshift-data execute-statement (users
 table)"
USERS_QUERY_ID=$(echo "$USERS_TABLE_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"'
 -f4)

# Wait for query to complete
echo "Waiting for users table creation to complete..."
sleep 5
```

```
# Create event table
echo "Creating event table..."
EVENT_TABLE_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "CREATE TABLE event(
    eventid INTEGER NOT NULL DISTKEY,
    venueid SMALLINT NOT NULL,
    catid SMALLINT NOT NULL,
    dateid SMALLINT NOT NULL SORTKEY,
    eventname VARCHAR(200),
    starttime TIMESTAMP
  );" 2>&1)
echo "$EVENT_TABLE_OUTPUT"
check_error "$EVENT_TABLE_OUTPUT" "aws redshift-data execute-statement (event
 table)"
EVENT_QUERY_ID=$(echo "$EVENT_TABLE_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"'
 -f4)

# Wait for query to complete
echo "Waiting for event table creation to complete..."
sleep 5

# Create sales table
echo "Creating sales table..."
SALES_TABLE_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "CREATE TABLE sales(
    salesid INTEGER NOT NULL,
    listid INTEGER NOT NULL DISTKEY,
    sellerid INTEGER NOT NULL,
    buyerid INTEGER NOT NULL,
    eventid INTEGER NOT NULL,
    dateid SMALLINT NOT NULL SORTKEY,
    qtysold SMALLINT NOT NULL,
    pricepaid DECIMAL(8,2),
    commission DECIMAL(8,2),
    saletime TIMESTAMP
  );" 2>&1)
echo "$SALES_TABLE_OUTPUT"
check_error "$SALES_TABLE_OUTPUT" "aws redshift-data execute-statement (sales
 table)"
```

```
SALES_QUERY_ID=$(echo "$SALES_TABLE_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"' 
 -f4)

# Wait for tables to be created
echo "Waiting for tables to be created..."
sleep 10

# Step 5: Load sample data from Amazon S3
echo "Loading sample data from Amazon S3..."

# Load data into users table
echo "Loading data into users table..."
USERS_LOAD_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "COPY users
    FROM 's3://redshift-downloads/tickit/allusers_pipe.txt'
    DELIMITER '|'
    TIMEFORMAT 'YYYY-MM-DD HH:MI:SS'
    IGNOREHEADER 1
    IAM_ROLE '$ROLE_ARN';" 2>&1)
echo "$USERS_LOAD_OUTPUT"
check_error "$USERS_LOAD_OUTPUT" "aws redshift-data execute-statement (load
 users)"
USERS_LOAD_QUERY_ID=$(echo "$USERS_LOAD_OUTPUT" | grep -o '"Id": "[^"]*' | cut -
d'"' -f4)

# Wait for data loading to complete
echo "Waiting for users data loading to complete..."
sleep 10

# Load data into event table
echo "Loading data into event table..."
EVENT_LOAD_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "COPY event
    FROM 's3://redshift-downloads/tickit/allevents_pipe.txt'
    DELIMITER '|'
    TIMEFORMAT 'YYYY-MM-DD HH:MI:SS'
    IGNOREHEADER 1
    IAM_ROLE '$ROLE_ARN';" 2>&1)
echo "$EVENT_LOAD_OUTPUT"
```

```
check_error "$EVENT_LOAD_OUTPUT" "aws redshift-data execute-statement (load
 event)"
EVENT_LOAD_QUERY_ID=$(echo "$EVENT_LOAD_OUTPUT" | grep -o '"Id": "[^"]*' | cut -
d'"' -f4)

# Wait for data loading to complete
echo "Waiting for event data loading to complete..."
sleep 10

# Load data into sales table
echo "Loading data into sales table..."
SALES_LOAD_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "COPY sales
    FROM 's3://redshift-downloads/tickit/sales_tab.txt'
    DELIMITER '\t'
    TIMEFORMAT 'MM/DD/YYYY HH:MI:SS'
    IGNOREHEADER 1
    IAM_ROLE '$ROLE_ARN';" 2>&1)
echo "$SALES_LOAD_OUTPUT"
check_error "$SALES_LOAD_OUTPUT" "aws redshift-data execute-statement (load
 sales)"
SALES_LOAD_QUERY_ID=$(echo "$SALES_LOAD_OUTPUT" | grep -o '"Id": "[^"]*' | cut -
d'"' -f4)

# Wait for data loading to complete
echo "Waiting for sales data loading to complete..."
sleep 30

# Step 6: Run sample queries
echo "Running sample queries..."

# Query 1: Find top 10 buyers by quantity
echo "Running query: Find top 10 buyers by quantity..."
QUERY1_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "SELECT firstname, lastname, total_quantity
    FROM (SELECT buyerid, sum(qtysold) total_quantity
          FROM sales
          GROUP BY buyerid
          ORDER BY total_quantity desc limit 10) Q, users
    WHERE Q.buyerid = userid
```

```
    ORDER BY Q.total_quantity desc;" 2>&1)
echo "$QUERY1_OUTPUT"
check_error "$QUERY1_OUTPUT" "aws redshift-data execute-statement (query 1)"
QUERY1_ID=$(echo "$QUERY1_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"' -f4)

# Wait for query to complete
echo "Waiting for query 1 to complete..."
sleep 10

# Get query 1 results
echo "Getting results for query 1..."
QUERY1_STATUS_OUTPUT=$(aws redshift-data describe-statement --id "$QUERY1_ID"
 2>&1)
echo "$QUERY1_STATUS_OUTPUT"
check_error "$QUERY1_STATUS_OUTPUT" "aws redshift-data describe-statement (query
 1)"

QUERY1_STATUS=$(echo "$QUERY1_STATUS_OUTPUT" | grep -o '"Status": "[^"]*' | cut -
d'"' -f4)
if [ "$QUERY1_STATUS" == "FINISHED" ]; then
  QUERY1_RESULTS=$(aws redshift-data get-statement-result --id "$QUERY1_ID" 2>&1)
  echo "Query 1 Results:"
  echo "$QUERY1_RESULTS"
else
  echo "Query 1 is not yet complete. Status: $QUERY1_STATUS"
  echo "Waiting additional time for query to complete..."
  sleep 20

  # Check again
  QUERY1_STATUS_OUTPUT=$(aws redshift-data describe-statement --id "$QUERY1_ID"
 2>&1)
  QUERY1_STATUS=$(echo "$QUERY1_STATUS_OUTPUT" | grep -o '"Status": "[^"]*' | cut
 -d'"' -f4)

  if [ "$QUERY1_STATUS" == "FINISHED" ]; then
    QUERY1_RESULTS=$(aws redshift-data get-statement-result --id "$QUERY1_ID"
 2>&1)
    echo "Query 1 Results:"
    echo "$QUERY1_RESULTS"
  else
    echo "Query 1 is still not complete. Status: $QUERY1_STATUS"
  fi
fi
```

```
# Query 2: Find events in the 99.9 percentile in terms of all time total sales
echo "Running query: Find events in the 99.9 percentile in terms of all time
 total sales..."
QUERY2_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "SELECT eventname, total_price
    FROM (SELECT eventid, total_price, ntile(1000) over(order by total_price
 desc) as percentile
          FROM (SELECT eventid, sum(pricepaid) total_price
                FROM sales
                GROUP BY eventid)) Q, event E
    WHERE Q.eventid = E.eventid
    AND percentile = 1
    ORDER BY total_price desc;" 2>&1)
echo "$QUERY2_OUTPUT"
check_error "$QUERY2_OUTPUT" "aws redshift-data execute-statement (query 2)"
QUERY2_ID=$(echo "$QUERY2_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"' -f4)

# Wait for query to complete
echo "Waiting for query 2 to complete..."
sleep 10

# Get query 2 results
echo "Getting results for query 2..."
QUERY2_STATUS_OUTPUT=$(aws redshift-data describe-statement --id "$QUERY2_ID"
 2>&1)
echo "$QUERY2_STATUS_OUTPUT"
check_error "$QUERY2_STATUS_OUTPUT" "aws redshift-data describe-statement (query
 2)"

QUERY2_STATUS=$(echo "$QUERY2_STATUS_OUTPUT" | grep -o '"Status": "[^"]*' | cut -
d'"' -f4)
if [ "$QUERY2_STATUS" == "FINISHED" ]; then
  QUERY2_RESULTS=$(aws redshift-data get-statement-result --id "$QUERY2_ID" 2>&1)
  echo "Query 2 Results:"
  echo "$QUERY2_RESULTS"
else
  echo "Query 2 is not yet complete. Status: $QUERY2_STATUS"
  echo "Waiting additional time for query to complete..."
  sleep 20

  # Check again
```

```
  QUERY2_STATUS_OUTPUT=$(aws redshift-data describe-statement --id "$QUERY2_ID"
 2>&1)
  QUERY2_STATUS=$(echo "$QUERY2_STATUS_OUTPUT" | grep -o '"Status": "[^"]*' | cut
 -d'"' -f4)

  if [ "$QUERY2_STATUS" == "FINISHED" ]; then
    QUERY2_RESULTS=$(aws redshift-data get-statement-result --id "$QUERY2_ID"
 2>&1)
    echo "Query 2 Results:"
    echo "$QUERY2_RESULTS"
  else
    echo "Query 2 is still not complete. Status: $QUERY2_STATUS"
  fi
fi

# Summary
echo ""
echo "========================================="
echo "TUTORIAL SUMMARY"
echo "========================================="
echo "You have successfully:"
echo "1. Created a Redshift Serverless namespace and workgroup"
echo "2. Created an IAM role with S3 access permissions"
echo "3. Stored admin credentials securely in AWS Secrets Manager"
echo "4. Created tables for sample data"
echo "5. Loaded sample data from Amazon S3"
echo "6. Run sample queries on the data"
echo ""
echo "Redshift Serverless Resources:"
echo "- Namespace: $NAMESPACE_NAME"
echo "- Workgroup: $WORKGROUP_NAME"
echo "- Database: $DB_NAME"
echo "- Endpoint: $WORKGROUP_ENDPOINT"
echo "- Credentials Secret: $SECRET_NAME"
echo ""
echo "To connect to your Redshift Serverless database using SQL tools:"
echo "- Host: $WORKGROUP_ENDPOINT"
echo "- Database: $DB_NAME"
echo "- Username: $ADMIN_USERNAME"
echo "- Password: Retrieve from AWS Secrets Manager secret '$SECRET_NAME'"
echo ""
echo "To retrieve the password from Secrets Manager (without jq):"
echo "aws secretsmanager get-secret-value --secret-id $SECRET_NAME --query
 'SecretString' --output text | sed -n 's/.*\"password\":\"\([^\"]*\)\".*/\1/p'"
```

```
echo ""

# Clean up temporary files
rm -f redshift-trust-policy.json redshift-s3-policy.json

# Clean up resources
cleanup_resources

echo "Tutorial completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - CreateNamespace
  - CreateRole
  - CreateSecret
  - CreateWorkgroup
  - DeleteNamespace
  - DeleteRole
  - DeleteRolePolicy
  - DeleteSecret
  - DeleteWorkgroup
  - GetNamespace
  - GetRole
  - GetSecretValue
  - GetWorkgroup
  - PutRolePolicy
  - UpdateNamespace

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva alla gestione della configurazione

L'esempio di codice seguente mostra come:

- Crea un bucket Amazon S3
- Creare un argomento Amazon SNS.
- Creare un ruolo IAM per Config
- Configurare il registratore di configurazione Config
- Configura il canale di distribuzione Config
- Avvia il registratore di configurazione
- Verifica la configurazione di Config

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS Config Setup Script (v2)
# This script sets up AWS Config with the AWS CLI

# Error handling
set -e
LOGFILE="aws-config-setup-v2.log"
touch $LOGFILE
exec > >(tee -a $LOGFILE)
exec 2>&1

# Function to handle errors
handle_error() {
    echo "ERROR: An error occurred at line $1"
    echo "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}
```

```
# Set trap for error handling
trap 'handle_error $LINENO' ERR

# Function to generate random identifier
generate_random_id() {
    echo $(openssl rand -hex 6)
}

# Function to check if command was successful
check_command() {
    if echo "$1" | grep -i "error" > /dev/null; then
        echo "ERROR: $1"
        return 1
    fi
    return 0
}

# Function to clean up resources
cleanup_resources() {
    if [ -n "$CONFIG_RECORDER_NAME" ]; then
        echo "Stopping configuration recorder..."
        aws configservice stop-configuration-recorder --configuration-recorder-
name "$CONFIG_RECORDER_NAME" 2>/dev/null || true
    fi

    # Check if we created a new delivery channel before trying to delete it
    if [ -n "$DELIVERY_CHANNEL_NAME" ] && [ "$CREATED_NEW_DELIVERY_CHANNEL" =
 "true" ]; then
        echo "Deleting delivery channel..."
        aws configservice delete-delivery-channel --delivery-channel-name
 "$DELIVERY_CHANNEL_NAME" 2>/dev/null || true
    fi

    if [ -n "$CONFIG_RECORDER_NAME" ] && [ "$CREATED_NEW_CONFIG_RECORDER" =
 "true" ]; then
        echo "Deleting configuration recorder..."
        aws configservice delete-configuration-recorder --configuration-recorder-
name "$CONFIG_RECORDER_NAME" 2>/dev/null || true
    fi

    if [ -n "$ROLE_NAME" ]; then
        if [ -n "$POLICY_NAME" ]; then
            echo "Detaching custom policy from role..."
```

```
            aws iam delete-role-policy --role-name "$ROLE_NAME" --policy-name
 "$POLICY_NAME" 2>/dev/null || true
        fi

        if [ -n "$MANAGED_POLICY_ARN" ]; then
            echo "Detaching managed policy from role..."
            aws iam detach-role-policy --role-name "$ROLE_NAME" --policy-arn
 "$MANAGED_POLICY_ARN" 2>/dev/null || true
        fi

        echo "Deleting IAM role..."
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi

    if [ -n "$SNS_TOPIC_ARN" ]; then
        echo "Deleting SNS topic..."
        aws sns delete-topic --topic-arn "$SNS_TOPIC_ARN" 2>/dev/null || true
    fi

    if [ -n "$S3_BUCKET_NAME" ]; then
        echo "Emptying S3 bucket..."
        aws s3 rm "s3://$S3_BUCKET_NAME" --recursive 2>/dev/null || true

        echo "Deleting S3 bucket..."
        if [ "$BUCKET_IS_SHARED" = "false" ]; then
            aws s3api delete-bucket --bucket "$S3_BUCKET_NAME" 2>/dev/null ||
 true
        fi
    fi
}

# Function to display created resources
display_resources() {
    echo ""
    echo "========================================="
    echo "CREATED RESOURCES"
    echo "========================================="
    echo "S3 Bucket: $S3_BUCKET_NAME"
    echo "SNS Topic ARN: $SNS_TOPIC_ARN"
    echo "IAM Role: $ROLE_NAME"
    if [ "$CREATED_NEW_CONFIG_RECORDER" = "true" ]; then
        echo "Configuration Recorder: $CONFIG_RECORDER_NAME (newly created)"
    else
        echo "Configuration Recorder: $CONFIG_RECORDER_NAME (existing)"
```

```
    fi
    if [ "$CREATED_NEW_DELIVERY_CHANNEL" = "true" ]; then
        echo "Delivery Channel: $DELIVERY_CHANNEL_NAME (newly created)"
    else
        echo "Delivery Channel: $DELIVERY_CHANNEL_NAME (existing)"
    fi
    echo "========================================="
}

# Get AWS account ID
echo "Getting AWS account ID..."
ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
if [ -z "$ACCOUNT_ID" ]; then
    echo "ERROR: Failed to get AWS account ID"
    exit 1
fi
echo "AWS Account ID: $ACCOUNT_ID"

# Generate random identifier for resources
RANDOM_ID=$(generate_random_id)
echo "Generated random identifier: $RANDOM_ID"

# Step 1: Create an S3 bucket
# Check for shared prereq bucket
PREREQ_BUCKET=$(aws cloudformation describe-stacks --stack-name tutorial-prereqs-
bucket \
    --query 'Stacks[0].Outputs[?OutputKey==`BucketName`].OutputValue' --output
 text 2>/dev/null)
if [ -n "$PREREQ_BUCKET" ] && [ "$PREREQ_BUCKET" != "None" ]; then
    S3_BUCKET_NAME="$PREREQ_BUCKET"
    BUCKET_IS_SHARED=true
    echo "Using shared bucket: $S3_BUCKET_NAME"
else
    BUCKET_IS_SHARED=false
    S3_BUCKET_NAME="configservice-${RANDOM_ID}"
    echo "Creating S3 bucket: $S3_BUCKET_NAME"
fi

# Get the current region
AWS_REGION=$(aws configure get region)
if [ -z "$AWS_REGION" ]; then
    AWS_REGION="us-east-1"  # Default to us-east-1 if no region is configured
fi
echo "Using AWS Region: $AWS_REGION"
```

```
# Create bucket with appropriate command based on region
if [ "$BUCKET_IS_SHARED" = "false" ]; then
    if [ "$AWS_REGION" = "us-east-1" ]; then
        BUCKET_RESULT=$(aws s3api create-bucket --bucket "$S3_BUCKET_NAME")
    else
        BUCKET_RESULT=$(aws s3api create-bucket --bucket "$S3_BUCKET_NAME" --
create-bucket-configuration LocationConstraint="$AWS_REGION")
    fi
    check_command "$BUCKET_RESULT"
    echo "S3 bucket created: $S3_BUCKET_NAME"

    aws s3api put-bucket-tagging --bucket "$S3_BUCKET_NAME" --tagging
 'TagSet=[{Key=project,Value=doc-smith},{Key=tutorial,Value=aws-config-gs}]'
    echo "Tags applied to S3 bucket"
else
    echo "Using shared bucket: $S3_BUCKET_NAME (skipping creation)"
fi

# Block public access for the bucket
aws s3api put-public-access-block \
    --bucket "$S3_BUCKET_NAME" \
    --public-access-block-configuration
 "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets
echo "Public access blocked for bucket"

# Step 2: Create an SNS topic
TOPIC_NAME="config-topic-${RANDOM_ID}"
echo "Creating SNS topic: $TOPIC_NAME"
SNS_RESULT=$(aws sns create-topic --name "$TOPIC_NAME" --tags
 Key=project,Value=doc-smith Key=tutorial,Value=aws-config-gs)
check_command "$SNS_RESULT"
SNS_TOPIC_ARN=$(echo "$SNS_RESULT" | grep -o 'arn:aws:sns:[^"]*')
echo "SNS topic created: $SNS_TOPIC_ARN"

# Step 3: Create an IAM role for AWS Config
ROLE_NAME="config-role-${RANDOM_ID}"
POLICY_NAME="config-delivery-permissions"
MANAGED_POLICY_ARN="arn:aws:iam::aws:policy/service-role/AWS_ConfigRole"

echo "Creating trust policy document..."
cat > config-trust-policy.json << EOF
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "config.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

echo "Creating IAM role: $ROLE_NAME"
ROLE_RESULT=$(aws iam create-role --role-name "$ROLE_NAME" --assume-role-policy-
document file://config-trust-policy.json)
check_command "$ROLE_RESULT"
ROLE_ARN=$(echo "$ROLE_RESULT" | grep -o 'arn:aws:iam::[^"]*' | head -1)
echo "IAM role created: $ROLE_ARN"

aws iam tag-role --role-name "$ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-config-gs
echo "Tags applied to IAM role"

echo "Attaching AWS managed policy to role..."
ATTACH_RESULT=$(aws iam attach-role-policy --role-name "$ROLE_NAME" --policy-arn
 "$MANAGED_POLICY_ARN")
check_command "$ATTACH_RESULT"
echo "AWS managed policy attached"

echo "Creating custom policy document for S3 and SNS access..."
cat > config-delivery-permissions.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::${S3_BUCKET_NAME}/AWSLogs/${ACCOUNT_ID}/*",
      "Condition": {
        "StringLike": {
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
```

```
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketAcl"
      ],
      "Resource": "arn:aws:s3:::${S3_BUCKET_NAME}"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": "${SNS_TOPIC_ARN}"
    }
  ]
}
EOF

echo "Attaching custom policy to role..."
POLICY_RESULT=$(aws iam put-role-policy --role-name "$ROLE_NAME" --policy-name
 "$POLICY_NAME" --policy-document file://config-delivery-permissions.json)
check_command "$POLICY_RESULT"
echo "Custom policy attached"

# Wait for IAM role to propagate
echo "Waiting for IAM role to propagate (15 seconds)..."
sleep 15

# Step 4: Check if configuration recorder already exists
CONFIG_RECORDER_NAME="default"
CREATED_NEW_CONFIG_RECORDER="false"

echo "Checking for existing configuration recorder..."
EXISTING_RECORDERS=$(aws configservice describe-configuration-recorders 2>/dev/
null || echo "")
if echo "$EXISTING_RECORDERS" | grep -q "name"; then
    echo "Configuration recorder already exists. Will update it."
    # Get the name of the existing recorder
    CONFIG_RECORDER_NAME=$(echo "$EXISTING_RECORDERS" | grep -o '"name": "[^"]*"'
 | head -1 | cut -d'"' -f4)
    echo "Using existing configuration recorder: $CONFIG_RECORDER_NAME"
else
```

```
    echo "No existing configuration recorder found. Will create a new one."
    CREATED_NEW_CONFIG_RECORDER="true"
fi

echo "Creating configuration recorder configuration..."
cat > configurationRecorder.json << EOF
{
  "name": "${CONFIG_RECORDER_NAME}",
  "roleARN": "${ROLE_ARN}",
  "recordingMode": {
    "recordingFrequency": "CONTINUOUS"
  }
}
EOF

echo "Creating recording group configuration..."
cat > recordingGroup.json << EOF
{
  "allSupported": true,
  "includeGlobalResourceTypes": true
}
EOF

echo "Setting up configuration recorder..."
RECORDER_RESULT=$(aws configservice put-configuration-recorder --configuration-
recorder file://configurationRecorder.json --recording-group file://
recordingGroup.json)
check_command "$RECORDER_RESULT"
echo "Configuration recorder set up"

if [ "$CREATED_NEW_CONFIG_RECORDER" = "true" ]; then
    aws configservice tag-resource --resource-arn "arn:aws:config:
${AWS_REGION}:${ACCOUNT_ID}:config-recorder/${CONFIG_RECORDER_NAME}" --tags
 Key=project,Value=doc-smith Key=tutorial,Value=aws-config-gs
    echo "Tags applied to configuration recorder"
fi

# Step 5: Check if delivery channel already exists
DELIVERY_CHANNEL_NAME="default"
CREATED_NEW_DELIVERY_CHANNEL="false"

echo "Checking for existing delivery channel..."
EXISTING_CHANNELS=$(aws configservice describe-delivery-channels 2>/dev/null ||
 echo "")
```

```
if echo "$EXISTING_CHANNELS" | grep -q "name"; then
    echo "Delivery channel already exists."
    # Get the name of the existing channel
    DELIVERY_CHANNEL_NAME=$(echo "$EXISTING_CHANNELS" | grep -o '"name": "[^"]*"'
 | head -1 | cut -d'"' -f4)
    echo "Using existing delivery channel: $DELIVERY_CHANNEL_NAME"

    # Update the existing delivery channel
    echo "Creating delivery channel configuration for update..."
    cat > deliveryChannel.json << EOF
{
  "name": "${DELIVERY_CHANNEL_NAME}",
  "s3BucketName": "${S3_BUCKET_NAME}",
  "snsTopicARN": "${SNS_TOPIC_ARN}",
  "configSnapshotDeliveryProperties": {
    "deliveryFrequency": "Six_Hours"
  }
}
EOF

    echo "Updating delivery channel..."
    CHANNEL_RESULT=$(aws configservice put-delivery-channel --delivery-channel
 file://deliveryChannel.json)
    check_command "$CHANNEL_RESULT"
    echo "Delivery channel updated"
else
    echo "No existing delivery channel found. Will create a new one."
    CREATED_NEW_DELIVERY_CHANNEL="true"

    echo "Creating delivery channel configuration..."
    cat > deliveryChannel.json << EOF
{
  "name": "${DELIVERY_CHANNEL_NAME}",
  "s3BucketName": "${S3_BUCKET_NAME}",
  "snsTopicARN": "${SNS_TOPIC_ARN}",
  "configSnapshotDeliveryProperties": {
    "deliveryFrequency": "Six_Hours"
  }
}
EOF

    echo "Creating delivery channel..."
    CHANNEL_RESULT=$(aws configservice put-delivery-channel --delivery-channel
 file://deliveryChannel.json)
```

```
    check_command "$CHANNEL_RESULT"
    echo "Delivery channel created"

    aws configservice tag-resource --resource-arn "arn:aws:config:
${AWS_REGION}:${ACCOUNT_ID}:delivery-channel/${DELIVERY_CHANNEL_NAME}" --tags
 Key=project,Value=doc-smith Key=tutorial,Value=aws-config-gs
    echo "Tags applied to delivery channel"
fi

# Step 6: Start the configuration recorder
echo "Checking configuration recorder status..."
RECORDER_STATUS=$(aws configservice describe-configuration-recorder-status 2>/
dev/null || echo "")
if echo "$RECORDER_STATUS" | grep -q '"recording": true'; then
    echo "Configuration recorder is already running."
else
    echo "Starting configuration recorder..."
    START_RESULT=$(aws configservice start-configuration-recorder --
configuration-recorder-name "$CONFIG_RECORDER_NAME")
    check_command "$START_RESULT"
    echo "Configuration recorder started"
fi

# Step 7: Verify the AWS Config setup
echo "Verifying delivery channel..."
VERIFY_CHANNEL=$(aws configservice describe-delivery-channels)
check_command "$VERIFY_CHANNEL"
echo "$VERIFY_CHANNEL"

echo "Verifying configuration recorder..."
VERIFY_RECORDER=$(aws configservice describe-configuration-recorders)
check_command "$VERIFY_RECORDER"
echo "$VERIFY_RECORDER"

echo "Verifying configuration recorder status..."
VERIFY_STATUS=$(aws configservice describe-configuration-recorder-status)
check_command "$VERIFY_STATUS"
echo "$VERIFY_STATUS"

# Display created resources
display_resources

# Ask if user wants to clean up resources
echo ""
```

```
echo "========================================="
echo "CLEANUP CONFIRMATION"
echo "========================================="
echo "Do you want to clean up all created resources? (y/n): "
CLEANUP_CHOICE='y'

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    echo "Cleaning up resources..."
    cleanup_resources
    echo "Cleanup completed."
else
    echo "Resources will not be cleaned up. You can manually clean them up
 later."
fi

echo "Script completed successfully!"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateBucket
  - CreateRole
  - CreateTopic
  - DeleteBucket
  - DeleteConfigurationRecorder
  - DeleteDeliveryChannel
  - DeleteRole
  - DeleteRolePolicy
  - DeleteTopic
  - DescribeConfigurationRecorderStatus
  - DescribeConfigurationRecorders
  - DescribeDeliveryChannels
  - DetachRolePolicy
  - GetCallerIdentity
  - PutConfigurationRecorder
  - PutDeliveryChannel

- PutPublicAccessBlock
- PutRolePolicy
- Rm
- StartConfigurationRecorder
- StopConfigurationRecorder

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva alla protezione dei dispositivi tramite Internet of Things

L'esempio di codice seguente mostra come:

- Crea ruoli IAM richiesti
- Abilita i controlli di controllo di IoT Device Defender
- Esegui un audit On-Demand
- Crea un'azione di mitigazione
- Applica azioni di mitigazione ai risultati
- Configura le notifiche SNS (opzionale)
- Abilita la registrazione IoT

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS IoT Device Defender Getting Started Script
```

```
# This script demonstrates how to use AWS IoT Device Defender to enable audit
 checks,
# view audit results, create mitigation actions, and apply them to findings.

set -euo pipefail

# Set up logging
LOG_FILE="iot-device-defender-script-$(date +%Y%m%d%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "================================================="
echo "AWS IoT Device Defender Getting Started Script"
echo "================================================="
echo "Starting script execution at $(date)"
echo ""

# Function to check for errors in command output
check_error() {
    if echo "$1" | grep -iE "An error occurred|Exception|Failed|usage: aws" > /
dev/null; then
        echo "ERROR: Command failed with the following output:"
        echo "$1"
        return 1
    fi
    return 0
}

# Function to safely extract JSON values using jq
extract_json_value() {
    local json="$1"
    local key="$2"
    echo "$json" | jq -r ".${key} // empty" 2>/dev/null || echo ""
}

# Function to validate JSON
validate_json() {
    local json="$1"
    echo "$json" | jq empty 2>/dev/null
}

# Function to check AWS CLI availability
check_aws_cli() {
    if ! command -v aws &> /dev/null; then
        echo "ERROR: AWS CLI is not installed or not in PATH"
```

```
        return 1
    fi
    if ! command -v jq &> /dev/null; then
        echo "ERROR: jq is not installed or not in PATH"
        return 1
    fi
    return 0
}

# Function to get AWS account ID
get_account_id() {
    local account_id
    account_id=$(aws sts get-caller-identity --query 'Account' --output text 2>/
dev/null) || true
    if [ -z "$account_id" ]; then
        echo "ERROR: Could not retrieve AWS account ID"
        return 1
    fi
    echo "$account_id"
    return 0
}

# Function to create IAM roles with retry logic
create_iam_role() {
    local ROLE_NAME=$1
    local TRUST_POLICY=$2
    local MANAGED_POLICY=$3
    local RETRY_COUNT=0
    local MAX_RETRIES=3

    echo "Creating IAM role: $ROLE_NAME"

    # Validate trust policy JSON
    if ! validate_json "$TRUST_POLICY"; then
        echo "ERROR: Invalid trust policy JSON for role $ROLE_NAME"
        return 1
    fi

    # Check if role already exists
    if aws iam get-role --role-name "$ROLE_NAME" >/dev/null 2>&1; then
        echo "Role $ROLE_NAME already exists, skipping creation"
        ROLE_ARN=$(aws iam get-role --role-name "$ROLE_NAME" --query 'Role.Arn'
 --output text 2>/dev/null) || true
        if [ -z "$ROLE_ARN" ]; then
```

```
            echo "ERROR: Could not retrieve ARN for existing role $ROLE_NAME"
            return 1
        fi
        echo "Role ARN: $ROLE_ARN"
        return 0
    fi

    # Create the role with trust policy and retry logic
    while [ $RETRY_COUNT -lt $MAX_RETRIES ]; do
        ROLE_RESULT=$(aws iam create-role \
            --role-name "$ROLE_NAME" \
            --assume-role-policy-document "$TRUST_POLICY" 2>&1) || true

        if check_error "$ROLE_RESULT"; then
            break
        fi

        RETRY_COUNT=$((RETRY_COUNT + 1))
        if [ $RETRY_COUNT -lt $MAX_RETRIES ]; then
            echo "Retrying role creation (attempt $((RETRY_COUNT + 1))/
$MAX_RETRIES)..."
            sleep $((RETRY_COUNT * 2))
        fi
    done

    if ! check_error "$ROLE_RESULT"; then
        echo "Failed to create role $ROLE_NAME after $MAX_RETRIES attempts"
        return 1
    fi

    aws iam tag-role --role-name "$ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-iot-device-defender-gs 2>&1 || true

    # For IoT logging role, create an inline policy instead of using a managed
 policy
    if [[ "$ROLE_NAME" == "AWSIoTLoggingRole" ]]; then
        local LOGGING_POLICY
        LOGGING_POLICY=$(cat <<'EOF'
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:PutMetricFilter",
                "logs:PutRetentionPolicy",
                "logs:GetLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": "arn:aws:logs:*:*:*"
        }
    ]
}
EOF
)

        if ! validate_json "$LOGGING_POLICY"; then
            echo "ERROR: Invalid logging policy JSON"
            return 1
        fi

        POLICY_RESULT=$(aws iam put-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-name "${ROLE_NAME}Policy" \
            --policy-document "$LOGGING_POLICY" 2>&1) || true

        if ! check_error "$POLICY_RESULT"; then
            echo "Failed to attach inline policy to role $ROLE_NAME"
            return 1
        fi
    elif [[ "$ROLE_NAME" == "IoTMitigationActionErrorLoggingRole" ]]; then
        local MITIGATION_POLICY
        MITIGATION_POLICY=$(cat <<'EOF'
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:UpdateCACertificate",
                "iot:UpdateCertificate",
                "iot:SetV2LoggingOptions",
                "iot:SetLoggingOptions",
                "iot:AddThingToThingGroup"
            ],
```

```
            "Resource": "arn:aws:iot:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "iot.amazonaws.com"
                }
            }
        }
    ]
}
EOF
)

        if ! validate_json "$MITIGATION_POLICY"; then
            echo "ERROR: Invalid mitigation policy JSON"
            return 1
        fi

        POLICY_RESULT=$(aws iam put-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-name "${ROLE_NAME}Policy" \
            --policy-document "$MITIGATION_POLICY" 2>&1) || true

        if ! check_error "$POLICY_RESULT"; then
            echo "Failed to attach inline policy to role $ROLE_NAME"
            return 1
        fi
    else
        # Attach managed policy to role if provided
        if [ -n "$MANAGED_POLICY" ]; then
            ATTACH_RESULT=$(aws iam attach-role-policy \
                --role-name "$ROLE_NAME" \
                --policy-arn "$MANAGED_POLICY" 2>&1) || true

            if ! check_error "$ATTACH_RESULT"; then
                echo "Failed to attach policy to role $ROLE_NAME"
                return 1
            fi
        fi
    fi
```

```
    echo "Role $ROLE_NAME created successfully"

    # Get the role ARN with error handling
    ROLE_ARN=$(aws iam get-role --role-name "$ROLE_NAME" --query 'Role.Arn' --
output text 2>/dev/null) || true
    if [ -z "$ROLE_ARN" ]; then
        echo "ERROR: Could not retrieve ARN for newly created role $ROLE_NAME"
        return 1
    fi
    echo "Role ARN: $ROLE_ARN"
    return 0
}

# Array to store created resources for cleanup
declare -a CREATED_RESOURCES

# Validate prerequisites
echo "Validating prerequisites..."
if ! check_aws_cli; then
    echo "ERROR: Prerequisites not met"
    exit 1
fi

ACCOUNT_ID=$(get_account_id) || exit 1
echo "AWS Account ID: $ACCOUNT_ID"
echo ""

# Step 1: Create IAM roles needed for the tutorial
echo "================================================="
echo "Step 1: Creating required IAM roles"
echo "================================================="

# Create IoT Device Defender Audit role
IOT_DEFENDER_AUDIT_TRUST_POLICY=$(cat <<'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
```

```
    }
  ]
}
EOF
)

if ! create_iam_role "AWSIoTDeviceDefenderAuditRole"
 "$IOT_DEFENDER_AUDIT_TRUST_POLICY" "arn:aws:iam::aws:policy/service-role/
AWSIoTDeviceDefenderAudit"; then
    echo "ERROR: Failed to create audit role"
    exit 1
fi
AUDIT_ROLE_ARN=$ROLE_ARN
CREATED_RESOURCES+=("IAM Role: AWSIoTDeviceDefenderAuditRole")

# Create IoT Logging role
IOT_LOGGING_TRUST_POLICY=$(cat <<'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
)

if ! create_iam_role "AWSIoTLoggingRole" "$IOT_LOGGING_TRUST_POLICY" ""; then
    echo "ERROR: Failed to create logging role"
    exit 1
fi
LOGGING_ROLE_ARN=$ROLE_ARN
CREATED_RESOURCES+=("IAM Role: AWSIoTLoggingRole")

# Create IoT Mitigation Action role
IOT_MITIGATION_TRUST_POLICY=$(cat <<'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
)

if ! create_iam_role "IoTMitigationActionErrorLoggingRole"
 "$IOT_MITIGATION_TRUST_POLICY" ""; then
    echo "ERROR: Failed to create mitigation role"
    exit 1
fi
MITIGATION_ROLE_ARN=$ROLE_ARN
CREATED_RESOURCES+=("IAM Role: IoTMitigationActionErrorLoggingRole")

# Wait for IAM role propagation
echo "Waiting for IAM role propagation..."
sleep 5

# Step 2: Enable audit checks
echo ""
echo "=================================================="
echo "Step 2: Enabling AWS IoT Device Defender audit checks"
echo "=================================================="

# Get current audit configuration
echo "Getting current audit configuration..."
CURRENT_CONFIG=$(aws iot describe-account-audit-configuration --output json 2>&1)
 || true
if validate_json "$CURRENT_CONFIG"; then
    echo "$CURRENT_CONFIG" | jq '.' 2>/dev/null || echo "Could not parse current
 configuration"
fi

# Enable specific audit checks with proper JSON escaping
echo "Enabling audit checks..."
AUDIT_CONFIG='{"LOGGING_DISABLED_CHECK":{"enabled":true}}'

if ! validate_json "$AUDIT_CONFIG"; then
```

```
    echo "ERROR: Invalid audit configuration JSON"
    exit 1
fi

UPDATE_RESULT=$(aws iot update-account-audit-configuration \
  --role-arn "$AUDIT_ROLE_ARN" \
  --audit-check-configurations "$AUDIT_CONFIG" 2>&1) || true

if ! check_error "$UPDATE_RESULT"; then
    echo "Failed to update audit configuration"
    exit 1
fi

echo "Audit checks enabled successfully"

# Step 3: Run an on-demand audit
echo ""
echo "=================================================="
echo "Step 3: Running an on-demand audit"
echo "=================================================="

echo "Starting on-demand audit task..."
AUDIT_TASK_RESULT=$(aws iot start-on-demand-audit-task \
  --target-check-names LOGGING_DISABLED_CHECK --output json 2>&1) || true

if ! check_error "$AUDIT_TASK_RESULT"; then
    echo "Failed to start on-demand audit task"
    exit 1
fi

TASK_ID=$(extract_json_value "$AUDIT_TASK_RESULT" "taskId")
if [ -z "$TASK_ID" ]; then
    echo "ERROR: Could not extract task ID from response"
    exit 1
fi

echo "Audit task started with ID: $TASK_ID"
CREATED_RESOURCES+=("Audit Task: $TASK_ID")

# Tag the audit task via IoT service
aws iot tag-resource --resource-arn "arn:aws:iot:$(aws configure get region):
${ACCOUNT_ID}:audittask/${TASK_ID}" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-iot-device-defender-gs 2>&1 || true
```

```
# Wait for the audit task to complete
echo "Waiting for audit task to complete (this may take a few minutes)..."
TASK_STATUS="IN_PROGRESS"
TIMEOUT=0
MAX_TIMEOUT=600
POLL_INTERVAL=15

while [ "$TASK_STATUS" != "COMPLETED" ]; do
    if [ $TIMEOUT -ge $MAX_TIMEOUT ]; then
        echo "WARNING: Audit task did not complete within ${MAX_TIMEOUT} seconds,
 continuing..."
        break
    fi

    sleep "$POLL_INTERVAL"
    TIMEOUT=$((TIMEOUT + POLL_INTERVAL))

    TASK_DETAILS=$(aws iot describe-audit-task --task-id "$TASK_ID" --output json
 2>&1) || true
    if validate_json "$TASK_DETAILS"; then
        TASK_STATUS=$(extract_json_value "$TASK_DETAILS" "taskStatus")
        echo "Current task status: $TASK_STATUS (elapsed: ${TIMEOUT}s)"

        if [ "$TASK_STATUS" = "FAILED" ]; then
            echo "WARNING: Audit task failed, continuing with script..."
            FAILURE_REASON=$(extract_json_value "$TASK_DETAILS"
 "taskStatistics.failedChecksNotApplicable")
            if [ -n "$FAILURE_REASON" ]; then
                echo "Reason: $FAILURE_REASON"
            fi
            break
        fi
    else
        echo "WARNING: Could not parse task details, retrying..."
    fi
done

echo "Audit task processing completed"

# Get audit findings (non-blocking)
echo "Getting audit findings..."
FINDINGS=$(aws iot list-audit-findings \
  --task-id "$TASK_ID" --output json 2>&1) || true
```

```
if validate_json "$FINDINGS"; then
    FINDING_COUNT=$(echo "$FINDINGS" | jq '.findings | length' 2>/dev/null ||
 echo "0")
    echo "Audit findings count: $FINDING_COUNT"
    if [ "$FINDING_COUNT" -gt 0 ]; then
        echo "Sample finding:"
        echo "$FINDINGS" | jq '.findings[0]' 2>/dev/null || echo "Could not parse
 finding"
    fi
else
    echo "WARNING: Could not parse audit findings response"
    FINDINGS='{"findings":[]}'
fi

# Check if we have any non-compliant findings
FINDING_ID=$(extract_json_value "$FINDINGS" "findings[0].findingId")
if [ -n "$FINDING_ID" ]; then
    echo "Found non-compliant finding with ID: $FINDING_ID"
    HAS_FINDINGS=true
else
    echo "No non-compliant findings detected"
    HAS_FINDINGS=false
fi

# Step 4: Create a mitigation action
echo ""
echo "=================================================="
echo "Step 4: Creating a mitigation action"
echo "=================================================="

# Check if mitigation action already exists and delete it
if aws iot describe-mitigation-action --action-name "EnableErrorLoggingAction" >/
dev/null 2>&1; then
    echo "Mitigation action 'EnableErrorLoggingAction' already exists, deleting
 it first..."
    aws iot delete-mitigation-action --action-name "EnableErrorLoggingAction"
 2>&1 || true
    sleep 2
fi

echo "Creating mitigation action to enable AWS IoT logging..."

# Build mitigation action parameters JSON
MITIGATION_PARAMS=$(cat <<EOF
```

```
{
  "enableIoTLoggingParams": {
    "roleArnForLogging": "$LOGGING_ROLE_ARN",
    "logLevel": "ERROR"
  }
}
EOF
)

if ! validate_json "$MITIGATION_PARAMS"; then
    echo "ERROR: Invalid mitigation parameters JSON"
    exit 1
fi

MITIGATION_RESULT=$(aws iot create-mitigation-action \
  --action-name "EnableErrorLoggingAction" \
  --role-arn "$MITIGATION_ROLE_ARN" \
  --action-params "$MITIGATION_PARAMS" --output json 2>&1) || true

if ! check_error "$MITIGATION_RESULT"; then
    echo "Failed to create mitigation action"
    exit 1
fi

if validate_json "$MITIGATION_RESULT"; then
    echo "Mitigation action created successfully"
    MITIGATION_ACTION_ARN=$(extract_json_value "$MITIGATION_RESULT" "actionArn")
    if [ -n "$MITIGATION_ACTION_ARN" ]; then
        echo "Mitigation Action ARN: $MITIGATION_ACTION_ARN"
        aws iot tag-resource --resource-arn "$MITIGATION_ACTION_ARN" --tags
 Key=project,Value=doc-smith Key=tutorial,Value=aws-iot-device-defender-gs 2>&1
 || true
    fi
else
    echo "WARNING: Could not validate mitigation action response, but action may
 have been created"
fi

CREATED_RESOURCES+=("Mitigation Action: EnableErrorLoggingAction")

# Step 5: Apply mitigation action to findings (if any)
if [ "$HAS_FINDINGS" = true ]; then
    echo ""
    echo "================================================="
```

```
    echo "Step 5: Applying mitigation action to findings"
    echo "================================================="

    MITIGATION_TASK_ID="MitigationTask-$(date +%s)"
    echo "Starting mitigation actions task with ID: $MITIGATION_TASK_ID"

    # Build target JSON
    TARGET_JSON=$(cat <<EOF
{
  "findingIds": ["$FINDING_ID"]
}
EOF
)

    if ! validate_json "$TARGET_JSON"; then
        echo "ERROR: Invalid target JSON"
        exit 1
    fi

    # Build audit check to actions mapping JSON
    AUDIT_CHECK_MAPPING=$(cat <<EOF
{
  "LOGGING_DISABLED_CHECK": ["EnableErrorLoggingAction"]
}
EOF
)

    if ! validate_json "$AUDIT_CHECK_MAPPING"; then
        echo "ERROR: Invalid audit check mapping JSON"
        exit 1
    fi

    MITIGATION_TASK_RESULT=$(aws iot start-audit-mitigation-actions-task \
      --task-id "$MITIGATION_TASK_ID" \
      --target "$TARGET_JSON" \
      --audit-check-to-actions-mapping "$AUDIT_CHECK_MAPPING" --output json 2>&1)
 || true

    if ! check_error "$MITIGATION_TASK_RESULT"; then
        echo "WARNING: Failed to start mitigation actions task, continuing..."
    else
        echo "Mitigation actions task started successfully"
        CREATED_RESOURCES+=("Mitigation Task: $MITIGATION_TASK_ID")
    fi
```

```
else
    echo ""
    echo "=================================================="
    echo "Step 5: Skipping mitigation action application (no findings)"
    echo "=================================================="
fi

# Step 6: Set up SNS notifications (optional)
echo ""
echo "=================================================="
echo "Step 6: Setting up SNS notifications"
echo "=================================================="

# Check if SNS topic already exists
SNS_TOPICS=$(aws sns list-topics --output json 2>&1) || true
TOPIC_ARN=""
if validate_json "$SNS_TOPICS"; then
    TOPIC_ARN=$(echo "$SNS_TOPICS" | jq -r '.Topics[] | select(.TopicArn |
 contains("IoTDDNotifications")) | .TopicArn' 2>/dev/null | head -1 || echo "")
fi

if [ -n "$TOPIC_ARN" ]; then
    echo "SNS topic 'IoTDDNotifications' already exists, using existing topic..."
    echo "Topic ARN: $TOPIC_ARN"
else
    echo "Creating SNS topic for notifications..."
    SNS_RESULT=$(aws sns create-topic --name "IoTDDNotifications" --tags
 Key=project,Value=doc-smith Key=tutorial,Value=aws-iot-device-defender-gs --
output json 2>&1) || true

    if ! check_error "$SNS_RESULT"; then
        echo "WARNING: Failed to create SNS topic, continuing..."
        SNS_RESULT=""
    else
        TOPIC_ARN=$(extract_json_value "$SNS_RESULT" "TopicArn")
        if [ -n "$TOPIC_ARN" ]; then
            echo "SNS topic created with ARN: $TOPIC_ARN"
            CREATED_RESOURCES+=("SNS Topic: IoTDDNotifications")
        fi
    fi
fi

if [ -n "$TOPIC_ARN" ]; then
    echo "Updating audit configuration to enable SNS notifications..."
```

```
    # Build SNS notification configuration JSON
    SNS_CONFIG=$(cat <<EOF
{
  "SNS": {
    "targetArn": "$TOPIC_ARN",
    "roleArn": "$AUDIT_ROLE_ARN",
    "enabled": true
  }
}
EOF
)

    if ! validate_json "$SNS_CONFIG"; then
        echo "ERROR: Invalid SNS configuration JSON"
        exit 1
    fi

    SNS_UPDATE_RESULT=$(aws iot update-account-audit-configuration \
      --audit-notification-target-configurations "$SNS_CONFIG" 2>&1) || true

    if ! check_error "$SNS_UPDATE_RESULT"; then
        echo "WARNING: Failed to update audit configuration for SNS
 notifications"
    else
        echo "SNS notifications enabled successfully"
    fi
else
    echo "Skipping SNS configuration due to topic creation failure"
fi

# Step 7: Enable AWS IoT logging
echo ""
echo "=================================================="
echo "Step 7: Enabling AWS IoT logging"
echo "=================================================="

echo "Setting up AWS IoT logging options..."

LOGGING_RESULT=$(aws iot set-v2-logging-options \
  --role-arn "$LOGGING_ROLE_ARN" \
  --default-log-level "ERROR" 2>&1) || true

if ! check_error "$LOGGING_RESULT"; then
```

```
    echo "V2 logging setup failed, trying v1 logging..."

    V1_LOGGING_CONFIG=$(cat <<EOF
{
  "roleArn": "$LOGGING_ROLE_ARN",
  "logLevel": "ERROR"
}
EOF
)

    if ! validate_json "$V1_LOGGING_CONFIG"; then
        echo "ERROR: Invalid v1 logging configuration JSON"
        exit 1
    fi

    LOGGING_RESULT_V1=$(aws iot set-logging-options \
      --logging-options-payload "$V1_LOGGING_CONFIG" 2>&1) || true

    if ! check_error "$LOGGING_RESULT_V1"; then
        echo "WARNING: Failed to set up AWS IoT logging with both v1 and v2
 methods, continuing..."
    else
        echo "AWS IoT v1 logging enabled successfully"
    fi
else
    echo "AWS IoT v2 logging enabled successfully"
fi

# Verify logging is enabled
echo "Verifying logging configuration..."
LOGGING_CONFIG=$(aws iot get-v2-logging-options --output json 2>&1) || true
if [ -n "$LOGGING_CONFIG" ] && ! check_error "$LOGGING_CONFIG" && validate_json
 "$LOGGING_CONFIG"; then
    echo "Logging configuration verified"
    echo "$LOGGING_CONFIG" | jq '.' 2>/dev/null || echo "Configuration retrieved
 but could not display details"
else
    echo "Could not verify logging configuration, but setup completed"
fi

# Script completed successfully
echo ""
echo "================================================="
echo "AWS IoT Device Defender setup completed successfully!"
```

```
echo "================================================="
echo "The following resources were created:"
for resource in "${CREATED_RESOURCES[@]+"${CREATED_RESOURCES[@]}"}"; do
    echo "- $resource"
done
echo ""

# Cleanup phase
echo "=========================================="
echo "CLEANUP"
echo "=========================================="
echo "Starting automatic cleanup of resources..."
echo "Waiting 10 seconds before cleanup to allow resource stabilization..."
sleep 10

# Disable AWS IoT logging
echo "Disabling AWS IoT logging..."

DISABLE_V2_RESULT=$(aws iot set-v2-logging-options \
  --default-log-level "DISABLED" 2>&1) || true

if check_error "$DISABLE_V2_RESULT"; then
    echo "V2 logging disabled successfully"
else
    echo "Attempting v1 logging disable..."

    V1_DISABLE_CONFIG=$(cat <<'EOF'
{
  "logLevel": "DISABLED"
}
EOF
)

    DISABLE_V1_RESULT=$(aws iot set-logging-options \
      --logging-options-payload "$V1_DISABLE_CONFIG" 2>&1) || true

    if check_error "$DISABLE_V1_RESULT"; then
        echo "V1 logging disabled successfully"
    else
        echo "WARNING: Could not disable logging"
    fi
fi

# Delete mitigation action
```

```
echo "Deleting mitigation action..."
aws iot delete-mitigation-action --action-name "EnableErrorLoggingAction" 2>&1 ||
 true

# Reset audit configuration
echo "Resetting IoT Device Defender audit configuration..."
RESET_AUDIT_CONFIG='{"LOGGING_DISABLED_CHECK":{"enabled":false}}'
aws iot update-account-audit-configuration \
  --audit-check-configurations "$RESET_AUDIT_CONFIG" 2>&1 || true

# Delete SNS topic
echo "Deleting SNS topic..."
if [ -n "${TOPIC_ARN:-}" ] && [ "$TOPIC_ARN" != "null" ]; then
    aws sns delete-topic --topic-arn "$TOPIC_ARN" 2>&1 || true
fi

# Clean up IAM roles with improved error handling
echo "Cleaning up IAM roles..."

cleanup_role() {
    local role_name=$1
    echo "Cleaning up role: $role_name"

    if aws iam get-role --role-name "$role_name" >/dev/null 2>&1; then
        ROLE_POLICIES=$(aws iam list-role-policies --role-name "$role_name" --
output json 2>&1 || echo '{"PolicyNames":[]}')
        if validate_json "$ROLE_POLICIES"; then
            while IFS= read -r policy_name; do
                if [ -n "$policy_name" ] && [ "$policy_name" != "null" ]; then
                    echo "  Deleting inline policy: $policy_name"
                    aws iam delete-role-policy \
                        --role-name "$role_name" \
                        --policy-name "$policy_name" 2>&1 || true
                fi
            done < <(echo "$ROLE_POLICIES" | jq -r '.PolicyNames[]' 2>/dev/null
 || echo "")
        fi

        ATTACHED_POLICIES=$(aws iam list-attached-role-policies --role-name
 "$role_name" --output json 2>&1 || echo '{"AttachedPolicies":[]}')
        if validate_json "$ATTACHED_POLICIES"; then
            while IFS= read -r policy_arn; do
                if [ -n "$policy_arn" ] && [ "$policy_arn" != "null" ]; then
                    echo "  Detaching managed policy: $policy_arn"
```

```
                    aws iam detach-role-policy \
                        --role-name "$role_name" \
                        --policy-arn "$policy_arn" 2>&1 || true
                fi
            done < <(echo "$ATTACHED_POLICIES" | jq -r
 '.AttachedPolicies[].PolicyArn' 2>/dev/null || echo "")
        fi

        echo "  Deleting role: $role_name"
        aws iam delete-role --role-name "$role_name" 2>&1 || true
    else
        echo "  Role $role_name does not exist or already deleted"
    fi
}

cleanup_role "AWSIoTDeviceDefenderAuditRole"
cleanup_role "AWSIoTLoggingRole"
cleanup_role "IoTMitigationActionErrorLoggingRole"

echo "Cleanup completed successfully"

echo ""
echo "Script execution completed at $(date)"
echo "Log file: $LOG_FILE"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateMitigationAction
  - CreateRole
  - CreateTopic
  - DeleteAccountAuditConfiguration
  - DeleteMitigationAction
  - DeleteRole
  - DeleteRolePolicy
  - DeleteTopic
  - DescribeAccountAuditConfiguration
  - DescribeAuditTask

- DetachRolePolicy
- GetLoggingOptions
- GetRole
- GetV2LoggingOptions
- ListAuditFindings
- ListAuditMitigationActionsTasks
- ListMitigationActions
- ListRolePolicies
- ListTopics
- PutRolePolicy
- SetLoggingOptions
- SetV2LoggingOptions
- StartAuditMitigationActionsTask
- StartOnDemandAuditTask
- UpdateAccountAuditConfiguration

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva ai feature store di machine learning

L'esempio di codice seguente mostra come:

- Impostare autorizzazioni IAM
- Crea un ruolo di SageMaker esecuzione
- Creazione di gruppi di funzionalità
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon SageMaker Feature Store Tutorial Script - Version 3
# This script demonstrates how to use Amazon SageMaker Feature Store with AWS CLI

# Setup logging
LOG_FILE="sagemaker-featurestore-tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting SageMaker Feature Store tutorial script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"
echo ""

# Track created resources for cleanup
CREATED_RESOURCES=()

# Function to handle errors
handle_error() {
    echo "ERROR: $1"
    echo "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Function to check command status
check_status() {
    if echo "$1" | grep -i "error" > /dev/null; then
        handle_error "$1"
    fi
}
```

```
# Function to wait for feature group to be created
wait_for_feature_group() {
    local feature_group_name=$1
    local status="Creating"

    echo "Waiting for feature group ${feature_group_name} to be created..."

    while [ "$status" = "Creating" ]; do
        sleep 5
        status=$(aws sagemaker describe-feature-group \
            --feature-group-name "${feature_group_name}" \
            --query 'FeatureGroupStatus' \
            --output text)
        echo "Current status: ${status}"

        if [ "$status" = "Failed" ]; then
            handle_error "Feature group ${feature_group_name} creation failed"
        fi
    done

    echo "Feature group ${feature_group_name} is now ${status}"
}

# Function to clean up resources
cleanup_resources() {
    echo "Cleaning up resources..."

    # Clean up in reverse order
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        resource="${CREATED_RESOURCES[$i]}"
        resource_type=$(echo "$resource" | cut -d: -f1)
        resource_name=$(echo "$resource" | cut -d: -f2)

        echo "Deleting $resource_type: $resource_name"

        case "$resource_type" in
            "FeatureGroup")
                aws sagemaker delete-feature-group --feature-group-name
 "$resource_name"
                ;;
            "S3Bucket")
                echo "Emptying S3 bucket: $resource_name"
                aws s3 rm "s3://$resource_name" --recursive 2>/dev/null
                echo "Deleting S3 bucket: $resource_name"
```

```
                aws s3api delete-bucket --bucket "$resource_name" 2>/dev/null
                ;;
            "IAMRole")
                echo "Detaching policies from role: $resource_name"
                aws iam detach-role-policy --role-name "$resource_name" --policy-
arn "arn:aws:iam::aws:policy/AmazonSageMakerFullAccess" 2>/dev/null
                aws iam detach-role-policy --role-name "$resource_name" --policy-
arn "arn:aws:iam::aws:policy/AmazonS3FullAccess" 2>/dev/null
                echo "Deleting IAM role: $resource_name"
                aws iam delete-role --role-name "$resource_name" 2>/dev/null
                ;;
            *)
                echo "Unknown resource type: $resource_type"
                ;;
        esac
    done
}

# Function to create SageMaker execution role
create_sagemaker_role() {
    local role_name="SageMakerFeatureStoreRole-$(openssl rand -hex 4)"

    echo "Creating SageMaker execution role: $role_name" >&2

    # Create trust policy document
    local trust_policy='{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "sagemaker.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    }'

    # Create the role
    local role_result=$(aws iam create-role \
        --role-name "$role_name" \
        --assume-role-policy-document "$trust_policy" \
        --description "SageMaker execution role for Feature Store tutorial" 2>&1)
```

```
    if echo "$role_result" | grep -i "error" > /dev/null; then
        handle_error "Failed to create IAM role: $role_result"
    fi

    echo "Role created successfully" >&2
    CREATED_RESOURCES+=("IAMRole:$role_name")

    # Tag the role
    echo "Tagging IAM role..." >&2
    aws iam tag-role --role-name "$role_name" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=sagemaker-featurestore 2>&1

    # Attach necessary policies
    echo "Attaching policies to role..." >&2

    # SageMaker execution policy
    local policy1_result=$(aws iam attach-role-policy \
        --role-name "$role_name" \
        --policy-arn "arn:aws:iam::aws:policy/AmazonSageMakerFullAccess" 2>&1)

    if echo "$policy1_result" | grep -i "error" > /dev/null; then
        handle_error "Failed to attach SageMaker policy: $policy1_result"
    fi

    # S3 access policy
    local policy2_result=$(aws iam attach-role-policy \
        --role-name "$role_name" \
        --policy-arn "arn:aws:iam::aws:policy/AmazonS3FullAccess" 2>&1)

    if echo "$policy2_result" | grep -i "error" > /dev/null; then
        handle_error "Failed to attach S3 policy: $policy2_result"
    fi

    # Get account ID for role ARN
    local account_id=$(aws sts get-caller-identity --query Account --output text)
    local role_arn="arn:aws:iam::${account_id}:role/${role_name}"

    echo "Role ARN: $role_arn" >&2
    echo "Waiting 10 seconds for role to propagate..." >&2
    sleep 10

    # Return only the role ARN to stdout
    echo "$role_arn"
}
```

```
# Handle SageMaker execution role
ROLE_ARN=""

if [ -z "$1" ]; then
    echo "Creating SageMaker execution role automatically..."
    ROLE_ARN=$(create_sagemaker_role)
    if [ -z "$ROLE_ARN" ]; then
        handle_error "Failed to create SageMaker execution role"
    fi
else
    ROLE_ARN="$1"

    # Validate the role ARN
    ROLE_NAME=$(echo "$ROLE_ARN" | sed 's/.*role\///')
    ROLE_CHECK=$(aws iam get-role --role-name "$ROLE_NAME" 2>&1)
    if echo "$ROLE_CHECK" | grep -i "error" > /dev/null; then
        echo "Creating a new role automatically..."
        ROLE_ARN=$(create_sagemaker_role)
        if [ -z "$ROLE_ARN" ]; then
            handle_error "Failed to create SageMaker execution role"
        fi
    fi
fi

# Handle cleanup option
AUTO_CLEANUP=""
if [ -n "$2" ]; then
    AUTO_CLEANUP="$2"
fi

# Generate a random identifier for resource names
RANDOM_ID=$(openssl rand -hex 4)
echo "Using random identifier: $RANDOM_ID"

# Set variables
ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
check_status "$ACCOUNT_ID"
echo "Account ID: $ACCOUNT_ID"

# Get current region
REGION=$(aws configure get region)
if [ -z "$REGION" ]; then
    REGION="us-east-1"
```

```
    echo "No default region configured, using: $REGION"
else
    echo "Using region: $REGION"
fi
# Check for shared prereq bucket
PREREQ_BUCKET=$(aws cloudformation describe-stacks --stack-name tutorial-prereqs-
bucket \
    --query 'Stacks[0].Outputs[?OutputKey==`BucketName`].OutputValue' --output
 text 2>/dev/null)
if [ -n "$PREREQ_BUCKET" ] && [ "$PREREQ_BUCKET" != "None" ]; then
    S3_BUCKET_NAME="$PREREQ_BUCKET"
    BUCKET_IS_SHARED=true
    echo "Using shared bucket: $S3_BUCKET_NAME"
else
    BUCKET_IS_SHARED=false
    S3_BUCKET_NAME="sagemaker-featurestore-${RANDOM_ID}-${ACCOUNT_ID}"
fi
PREFIX="featurestore-tutorial"
CURRENT_TIME=$(date +%s)

echo "Creating S3 bucket: $S3_BUCKET_NAME"
# Create bucket in current region (skip if using shared bucket)
if [ "$BUCKET_IS_SHARED" = "false" ]; then
if [ "$REGION" = "us-east-1" ]; then
    BUCKET_RESULT=$(aws s3api create-bucket --bucket "$S3_BUCKET_NAME" \
        --region "$REGION" 2>&1)
else
    BUCKET_RESULT=$(aws s3api create-bucket --bucket "$S3_BUCKET_NAME" \
        --region "$REGION" \
        --create-bucket-configuration LocationConstraint="$REGION" 2>&1)
fi

if echo "$BUCKET_RESULT" | grep -i "error" > /dev/null; then
    echo "Failed to create S3 bucket: $BUCKET_RESULT"
    exit 1
fi

echo "$BUCKET_RESULT"
CREATED_RESOURCES+=("S3Bucket:$S3_BUCKET_NAME")

# Tag the S3 bucket
echo "Tagging S3 bucket: $S3_BUCKET_NAME"
```

```
aws s3api put-bucket-tagging --bucket "$S3_BUCKET_NAME" --tagging
 'TagSet=[{Key=project,Value=doc-smith},{Key=tutorial,Value=sagemaker-
featurestore}]' 2>&1

# Block public access to the bucket
BLOCK_RESULT=$(aws s3api put-public-access-block \
    --bucket "$S3_BUCKET_NAME" \
    --public-access-block-configuration
 "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets
 2>&1)

if echo "$BLOCK_RESULT" | grep -i "error" > /dev/null; then
    echo "Failed to block public access to S3 bucket: $BLOCK_RESULT"
    cleanup_resources
    exit 1
fi
else
    echo "Using shared bucket (skipping creation)"
fi

# Create feature groups
echo "Creating feature groups..."

# Create customers feature group
CUSTOMERS_FEATURE_GROUP_NAME="customers-feature-group-${RANDOM_ID}"
echo "Creating customers feature group: $CUSTOMERS_FEATURE_GROUP_NAME"

CUSTOMERS_RESPONSE=$(aws sagemaker create-feature-group \
    --feature-group-name "$CUSTOMERS_FEATURE_GROUP_NAME" \
    --record-identifier-feature-name "customer_id" \
    --event-time-feature-name "EventTime" \
    --feature-definitions '[
        {"FeatureName": "customer_id", "FeatureType": "Integral"},
        {"FeatureName": "name", "FeatureType": "String"},
        {"FeatureName": "age", "FeatureType": "Integral"},
        {"FeatureName": "address", "FeatureType": "String"},
        {"FeatureName": "membership_type", "FeatureType": "String"},
        {"FeatureName": "EventTime", "FeatureType": "Fractional"}
    ]' \
    --online-store-config '{"EnableOnlineStore": true}' \
    --offline-store-config '{
        "S3StorageConfig": {
            "S3Uri": "s3://'${S3_BUCKET_NAME}'/'${PREFIX}'"
        },
```

```
        "DisableGlueTableCreation": false
    }' \
    --role-arn "$ROLE_ARN" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=sagemaker-featurestore
 2>&1)

if echo "$CUSTOMERS_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to create customers feature group: $CUSTOMERS_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$CUSTOMERS_RESPONSE"
CREATED_RESOURCES+=("FeatureGroup:$CUSTOMERS_FEATURE_GROUP_NAME")

# Create orders feature group
ORDERS_FEATURE_GROUP_NAME="orders-feature-group-${RANDOM_ID}"
echo "Creating orders feature group: $ORDERS_FEATURE_GROUP_NAME"

ORDERS_RESPONSE=$(aws sagemaker create-feature-group \
    --feature-group-name "$ORDERS_FEATURE_GROUP_NAME" \
    --record-identifier-feature-name "customer_id" \
    --event-time-feature-name "EventTime" \
    --feature-definitions '[
        {"FeatureName": "customer_id", "FeatureType": "Integral"},
        {"FeatureName": "order_id", "FeatureType": "String"},
        {"FeatureName": "order_date", "FeatureType": "String"},
        {"FeatureName": "product", "FeatureType": "String"},
        {"FeatureName": "quantity", "FeatureType": "Integral"},
        {"FeatureName": "amount", "FeatureType": "Fractional"},
        {"FeatureName": "EventTime", "FeatureType": "Fractional"}
    ]' \
    --online-store-config '{"EnableOnlineStore": true}' \
    --offline-store-config '{
        "S3StorageConfig": {
            "S3Uri": "s3://'${S3_BUCKET_NAME}'/'${PREFIX}'"
        },
        "DisableGlueTableCreation": false
    }' \
    --role-arn "$ROLE_ARN" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=sagemaker-featurestore
 2>&1)

if echo "$ORDERS_RESPONSE" | grep -i "error" > /dev/null; then
```

```
    echo "Failed to create orders feature group: $ORDERS_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$ORDERS_RESPONSE"
CREATED_RESOURCES+=("FeatureGroup:$ORDERS_FEATURE_GROUP_NAME")

# Wait for feature groups to be created
wait_for_feature_group "$CUSTOMERS_FEATURE_GROUP_NAME"
wait_for_feature_group "$ORDERS_FEATURE_GROUP_NAME"

# Ingest data into feature groups
echo "Ingesting data into feature groups..."

# Ingest customer data
echo "Ingesting customer data..."
CUSTOMER1_RESPONSE=$(aws sagemaker-featurestore-runtime put-record \
    --feature-group-name "$CUSTOMERS_FEATURE_GROUP_NAME" \
    --record '[
        {"FeatureName": "customer_id", "ValueAsString": "573291"},
        {"FeatureName": "name", "ValueAsString": "John Doe"},
        {"FeatureName": "age", "ValueAsString": "35"},
        {"FeatureName": "address", "ValueAsString": "123 Main St"},
        {"FeatureName": "membership_type", "ValueAsString": "premium"},
        {"FeatureName": "EventTime", "ValueAsString": "'${CURRENT_TIME}'"}
    ]' 2>&1)

if echo "$CUSTOMER1_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to ingest customer 1 data: $CUSTOMER1_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$CUSTOMER1_RESPONSE"

CUSTOMER2_RESPONSE=$(aws sagemaker-featurestore-runtime put-record \
    --feature-group-name "$CUSTOMERS_FEATURE_GROUP_NAME" \
    --record '[
        {"FeatureName": "customer_id", "ValueAsString": "109382"},
        {"FeatureName": "name", "ValueAsString": "Jane Smith"},
        {"FeatureName": "age", "ValueAsString": "28"},
        {"FeatureName": "address", "ValueAsString": "456 Oak Ave"},
        {"FeatureName": "membership_type", "ValueAsString": "standard"},
```

```
        {"FeatureName": "EventTime", "ValueAsString": "'${CURRENT_TIME}'"}
    ]' 2>&1)

if echo "$CUSTOMER2_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to ingest customer 2 data: $CUSTOMER2_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$CUSTOMER2_RESPONSE"

# Ingest order data
echo "Ingesting order data..."
ORDER1_RESPONSE=$(aws sagemaker-featurestore-runtime put-record \
    --feature-group-name "$ORDERS_FEATURE_GROUP_NAME" \
    --record '[
        {"FeatureName": "customer_id", "ValueAsString": "573291"},
        {"FeatureName": "order_id", "ValueAsString": "ORD-001"},
        {"FeatureName": "order_date", "ValueAsString": "2023-01-15"},
        {"FeatureName": "product", "ValueAsString": "Laptop"},
        {"FeatureName": "quantity", "ValueAsString": "1"},
        {"FeatureName": "amount", "ValueAsString": "1299.99"},
        {"FeatureName": "EventTime", "ValueAsString": "'${CURRENT_TIME}'"}
    ]' 2>&1)

if echo "$ORDER1_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to ingest order 1 data: $ORDER1_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$ORDER1_RESPONSE"

ORDER2_RESPONSE=$(aws sagemaker-featurestore-runtime put-record \
    --feature-group-name "$ORDERS_FEATURE_GROUP_NAME" \
    --record '[
        {"FeatureName": "customer_id", "ValueAsString": "109382"},
        {"FeatureName": "order_id", "ValueAsString": "ORD-002"},
        {"FeatureName": "order_date", "ValueAsString": "2023-01-20"},
        {"FeatureName": "product", "ValueAsString": "Smartphone"},
        {"FeatureName": "quantity", "ValueAsString": "1"},
        {"FeatureName": "amount", "ValueAsString": "899.99"},
        {"FeatureName": "EventTime", "ValueAsString": "'${CURRENT_TIME}'"}
    ]' 2>&1)
```

```
if echo "$ORDER2_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to ingest order 2 data: $ORDER2_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$ORDER2_RESPONSE"

# Retrieve records from feature groups
echo "Retrieving records from feature groups..."

# Get a single customer record
echo "Getting customer record with ID 573291:"
CUSTOMER_RECORD=$(aws sagemaker-featurestore-runtime get-record \
    --feature-group-name "$CUSTOMERS_FEATURE_GROUP_NAME" \
    --record-identifier-value-as-string "573291" 2>&1)

if echo "$CUSTOMER_RECORD" | grep -i "error" > /dev/null; then
    echo "Failed to get customer record: $CUSTOMER_RECORD"
    cleanup_resources
    exit 1
fi

echo "$CUSTOMER_RECORD"

# Get multiple records using batch-get-record
echo "Getting multiple records using batch-get-record:"
BATCH_RECORDS=$(aws sagemaker-featurestore-runtime batch-get-record \
    --identifiers '[
        {
            "FeatureGroupName": "'${CUSTOMERS_FEATURE_GROUP_NAME}'",
            "RecordIdentifiersValueAsString": ["573291", "109382"]
        },
        {
            "FeatureGroupName": "'${ORDERS_FEATURE_GROUP_NAME}'",
            "RecordIdentifiersValueAsString": ["573291", "109382"]
        }
    ]' 2>&1)

if echo "$BATCH_RECORDS" | grep -i "error" > /dev/null && ! echo "$BATCH_RECORDS"
 | grep -i "Records" > /dev/null; then
    echo "Failed to get batch records: $BATCH_RECORDS"
    cleanup_resources
```

```
    exit 1
fi

echo "$BATCH_RECORDS"

# List feature groups
echo "Listing feature groups:"
FEATURE_GROUPS=$(aws sagemaker list-feature-groups 2>&1)

if echo "$FEATURE_GROUPS" | grep -i "error" > /dev/null; then
    echo "Failed to list feature groups: $FEATURE_GROUPS"
    cleanup_resources
    exit 1
fi

echo "$FEATURE_GROUPS"

# Display summary of created resources
echo ""
echo "============================================="
echo "TUTORIAL COMPLETED SUCCESSFULLY!"
echo "============================================="
echo "Resources created:"
echo "- S3 Bucket: $S3_BUCKET_NAME"
echo "- Customers Feature Group: $CUSTOMERS_FEATURE_GROUP_NAME"
echo "- Orders Feature Group: $ORDERS_FEATURE_GROUP_NAME"
if [[ " ${CREATED_RESOURCES[@]} " =~ " IAMRole:" ]]; then
    echo "- IAM Role: $(echo "${CREATED_RESOURCES[@]}" | grep -o 'IAMRole:
[^[:space:]]*' | cut -d: -f2)"
fi
echo ""
echo "You can now:"
echo "1. View your feature groups in the SageMaker console"
echo "2. Query the offline store using Amazon Athena"
echo "3. Use the feature groups in your ML workflows"
echo "============================================="
echo ""

# Handle cleanup
if [ "$AUTO_CLEANUP" = "y" ]; then
    echo "Auto-cleanup enabled. Starting cleanup..."
    cleanup_resources
    echo "Cleanup completed."
elif [ "$AUTO_CLEANUP" = "n" ]; then
```

```
    echo "Auto-cleanup disabled. Resources will remain in your account."
    echo "To clean up later, run this script again with cleanup option 'y'"
else
    echo "========================================="
    echo "CLEANUP CONFIRMATION"
    echo "========================================="
    echo "Do you want to clean up all created resources? (y/n): "
    read -r CLEANUP_CHOICE

    if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
        echo "Starting cleanup..."
        cleanup_resources
        echo "Cleanup completed."
    else
        echo "Skipping cleanup. Resources will remain in your account."
        echo "To clean up later, delete the following resources:"
        echo "- Feature Groups: $CUSTOMERS_FEATURE_GROUP_NAME,
 $ORDERS_FEATURE_GROUP_NAME"
        echo "- S3 Bucket: $S3_BUCKET_NAME"
        if [[ " ${CREATED_RESOURCES[@]} " =~ " IAMRole:" ]]; then
            echo "- IAM Role: $(echo "${CREATED_RESOURCES[@]}" | grep -o
 'IAMRole:[^[:space:]]*' | cut -d: -f2)"
        fi
        echo ""
        echo "Estimated ongoing cost: ~$0.01 per month for online store"
    fi
fi

echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateBucket
  - CreateFeatureGroup
  - CreateRole
  - DeleteBucket
  - DeleteFeatureGroup
  - DeleteRole
  - DescribeFeatureGroup

- DetachRolePolicy
- GetCallerIdentity
- GetRole
- ListFeatureGroups
- PutPublicAccessBlock
- Rm

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva ai cluster Kubernetes gestiti

L'esempio di codice seguente mostra come:

- Crea un VPC per il tuo cluster EKS
- Crea ruoli IAM per il tuo cluster EKS
- Creazione di un cluster EKS
- Configura kubectl per comunicare con il tuo cluster
- Creazione di un gruppo di nodi gestito
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon EKS Cluster Creation Script (v2)
```

```
# This script creates an Amazon EKS cluster with a managed node group using the
 AWS CLI

# Set up logging
LOG_FILE="eks-cluster-creation-v2.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon EKS cluster creation script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Error handling function
handle_error() {
    echo "ERROR: $1"
    echo "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Function to check command success
check_command() {
    if [ $? -ne 0 ] || echo "$1" | grep -i "error" > /dev/null; then
        handle_error "$1"
    fi
}

# Function to check if kubectl is installed
check_kubectl() {
    if ! command -v kubectl &> /dev/null; then
        echo "WARNING: kubectl is not installed or not in your PATH."
        echo ""
        echo "To install kubectl, follow these instructions based on your
 operating system:"
        echo ""
        echo "For Linux:"
        echo "  1. Download the latest release:"
        echo "     curl -LO \"https://dl.k8s.io/release/\$(curl -L -s https://
dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl\""
        echo ""
        echo "  2. Make the kubectl binary executable:"
        echo "     chmod +x ./kubectl"
        echo ""
        echo "  3. Move the binary to your PATH:"
        echo "     sudo mv ./kubectl /usr/local/bin/kubectl"
        echo ""
```

```
        echo "For macOS:"
        echo "  1. Using Homebrew:"
        echo "     brew install kubectl"
        echo "     or"
        echo "  2. Using curl:"
        echo "     curl -LO \"https://dl.k8s.io/release/\$(curl -L -s https://
dl.k8s.io/release/stable.txt)/bin/darwin/amd64/kubectl\""
        echo "     chmod +x ./kubectl"
        echo "     sudo mv ./kubectl /usr/local/bin/kubectl"
        echo ""
        echo "For Windows:"
        echo "  1. Using curl:"
        echo "     curl -LO \"https://dl.k8s.io/release/v1.28.0/bin/windows/
amd64/kubectl.exe\""
        echo "     Add the binary to your PATH"
        echo "     or"
        echo "  2. Using Chocolatey:"
        echo "     choco install kubernetes-cli"
        echo ""
        echo "After installation, verify with: kubectl version --client"
        echo ""
        return 1
    fi
    return 0
}

# Generate a random identifier for resource names
RANDOM_ID=$(LC_ALL=C tr -dc 'a-z0-9' < /dev/urandom | fold -w 6 | head -n 1)
STACK_NAME="eks-vpc-stack-${RANDOM_ID}"
CLUSTER_NAME="eks-cluster-${RANDOM_ID}"
NODEGROUP_NAME="eks-nodegroup-${RANDOM_ID}"
CLUSTER_ROLE_NAME="EKSClusterRole-${RANDOM_ID}"
NODE_ROLE_NAME="EKSNodeRole-${RANDOM_ID}"

echo "Using the following resource names:"
echo "- VPC Stack: $STACK_NAME"
echo "- EKS Cluster: $CLUSTER_NAME"
echo "- Node Group: $NODEGROUP_NAME"
echo "- Cluster IAM Role: $CLUSTER_ROLE_NAME"
echo "- Node IAM Role: $NODE_ROLE_NAME"

# Array to track created resources for cleanup
declare -a CREATED_RESOURCES
```

```
# Function to clean up resources
cleanup_resources() {
    echo "Cleaning up resources in reverse order..."

    # Check if node group exists and delete it
    if aws eks list-nodegroups --cluster-name "$CLUSTER_NAME" --query
 "nodegroups[?contains(@,'$NODEGROUP_NAME')]" --output text 2>/dev/null | grep -q
 "$NODEGROUP_NAME"; then
        echo "Deleting node group: $NODEGROUP_NAME"
        aws eks delete-nodegroup --cluster-name "$CLUSTER_NAME" --nodegroup-name
 "$NODEGROUP_NAME"
        echo "Waiting for node group deletion to complete..."
        aws eks wait nodegroup-deleted --cluster-name "$CLUSTER_NAME" --
nodegroup-name "$NODEGROUP_NAME"
        echo "Node group deleted successfully."
    fi

    # Check if cluster exists and delete it
    if aws eks describe-cluster --name "$CLUSTER_NAME" 2>/dev/null; then
        echo "Deleting cluster: $CLUSTER_NAME"
        aws eks delete-cluster --name "$CLUSTER_NAME"
        echo "Waiting for cluster deletion to complete (this may take several
 minutes)..."
        aws eks wait cluster-deleted --name "$CLUSTER_NAME"
        echo "Cluster deleted successfully."
    fi

    # Check if CloudFormation stack exists and delete it
    if aws cloudformation describe-stacks --stack-name "$STACK_NAME" 2>/dev/null;
 then
        echo "Deleting CloudFormation stack: $STACK_NAME"
        aws cloudformation delete-stack --stack-name "$STACK_NAME"
        echo "Waiting for CloudFormation stack deletion to complete..."
        aws cloudformation wait stack-delete-complete --stack-name "$STACK_NAME"
        echo "CloudFormation stack deleted successfully."
    fi

    # Clean up IAM roles
    if aws iam get-role --role-name "$NODE_ROLE_NAME" 2>/dev/null; then
        echo "Detaching policies from node role: $NODE_ROLE_NAME"
        aws iam detach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSWorkerNodePolicy --role-name "$NODE_ROLE_NAME"
        aws iam detach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEC2ContainerRegistryReadOnly --role-name "$NODE_ROLE_NAME"
```

```
        aws iam detach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKS_CNI_Policy --role-name "$NODE_ROLE_NAME"
        echo "Deleting node role: $NODE_ROLE_NAME"
        aws iam delete-role --role-name "$NODE_ROLE_NAME"
        echo "Node role deleted successfully."
    fi

    if aws iam get-role --role-name "$CLUSTER_ROLE_NAME" 2>/dev/null; then
        echo "Detaching policies from cluster role: $CLUSTER_ROLE_NAME"
        aws iam detach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSClusterPolicy --role-name "$CLUSTER_ROLE_NAME"
        echo "Deleting cluster role: $CLUSTER_ROLE_NAME"
        aws iam delete-role --role-name "$CLUSTER_ROLE_NAME"
        echo "Cluster role deleted successfully."
    fi

    echo "Cleanup complete."
}

# Trap to ensure cleanup on script exit
trap 'echo "Script interrupted. Cleaning up resources..."; cleanup_resources;
 exit 1' SIGINT SIGTERM

# Verify AWS CLI configuration
echo "Verifying AWS CLI configuration..."
AWS_ACCOUNT_INFO=$(aws sts get-caller-identity)
check_command "$AWS_ACCOUNT_INFO"
echo "AWS CLI is properly configured."

# Step 1: Create VPC using CloudFormation
echo "Step 1: Creating VPC with CloudFormation..."
echo "Creating CloudFormation stack: $STACK_NAME"

# Create the CloudFormation stack
CF_CREATE_OUTPUT=$(aws cloudformation create-stack \
  --stack-name "$STACK_NAME" \
  --template-url https://s3.us-west-2.amazonaws.com/amazon-eks/
cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=eks-gs)
check_command "$CF_CREATE_OUTPUT"
CREATED_RESOURCES+=("CloudFormation Stack: $STACK_NAME")

echo "Waiting for CloudFormation stack to complete (this may take a few
 minutes)..."
```

```
aws cloudformation wait stack-create-complete --stack-name "$STACK_NAME"
if [ $? -ne 0 ]; then
    handle_error "CloudFormation stack creation failed"
fi
echo "CloudFormation stack created successfully."

# Step 2: Create IAM roles for EKS
echo "Step 2: Creating IAM roles for EKS..."

# Create cluster role trust policy
echo "Creating cluster role trust policy..."
cat > eks-cluster-role-trust-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create cluster role
echo "Creating cluster IAM role: $CLUSTER_ROLE_NAME"
CLUSTER_ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$CLUSTER_ROLE_NAME" \
  --assume-role-policy-document file://"eks-cluster-role-trust-policy.json")
check_command "$CLUSTER_ROLE_OUTPUT"
aws iam tag-role --role-name "$CLUSTER_ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=eks-gs
CREATED_RESOURCES+=("IAM Role: $CLUSTER_ROLE_NAME")

# Attach policy to cluster role
echo "Attaching EKS cluster policy to role..."
ATTACH_CLUSTER_POLICY_OUTPUT=$(aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy \
  --role-name "$CLUSTER_ROLE_NAME")
check_command "$ATTACH_CLUSTER_POLICY_OUTPUT"

# Create node role trust policy
```

```
echo "Creating node role trust policy..."
cat > node-role-trust-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create node role
echo "Creating node IAM role: $NODE_ROLE_NAME"
NODE_ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$NODE_ROLE_NAME" \
  --assume-role-policy-document file://"node-role-trust-policy.json")
check_command "$NODE_ROLE_OUTPUT"
aws iam tag-role --role-name "$NODE_ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=eks-gs
CREATED_RESOURCES+=("IAM Role: $NODE_ROLE_NAME")

# Attach policies to node role
echo "Attaching EKS node policies to role..."
ATTACH_NODE_POLICY1_OUTPUT=$(aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name "$NODE_ROLE_NAME")
check_command "$ATTACH_NODE_POLICY1_OUTPUT"

ATTACH_NODE_POLICY2_OUTPUT=$(aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name "$NODE_ROLE_NAME")
check_command "$ATTACH_NODE_POLICY2_OUTPUT"

ATTACH_NODE_POLICY3_OUTPUT=$(aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
  --role-name "$NODE_ROLE_NAME")
check_command "$ATTACH_NODE_POLICY3_OUTPUT"

# Step 3: Get VPC and subnet information
```

```
echo "Step 3: Getting VPC and subnet information..."

VPC_ID=$(aws cloudformation describe-stacks \
  --stack-name "$STACK_NAME" \
  --query "Stacks[0].Outputs[?OutputKey=='VpcId'].OutputValue" \
  --output text)
if [ -z "$VPC_ID" ]; then
    handle_error "Failed to get VPC ID from CloudFormation stack"
fi
echo "VPC ID: $VPC_ID"

SUBNET_IDS=$(aws cloudformation describe-stacks \
  --stack-name "$STACK_NAME" \
  --query "Stacks[0].Outputs[?OutputKey=='SubnetIds'].OutputValue" \
  --output text)
if [ -z "$SUBNET_IDS" ]; then
    handle_error "Failed to get Subnet IDs from CloudFormation stack"
fi
echo "Subnet IDs: $SUBNET_IDS"

SECURITY_GROUP_ID=$(aws cloudformation describe-stacks \
  --stack-name "$STACK_NAME" \
  --query "Stacks[0].Outputs[?OutputKey=='SecurityGroups'].OutputValue" \
  --output text)
if [ -z "$SECURITY_GROUP_ID" ]; then
    handle_error "Failed to get Security Group ID from CloudFormation stack"
fi
echo "Security Group ID: $SECURITY_GROUP_ID"

# Step 4: Create EKS cluster
echo "Step 4: Creating EKS cluster: $CLUSTER_NAME"

CLUSTER_ROLE_ARN=$(aws iam get-role --role-name "$CLUSTER_ROLE_NAME" --query
 "Role.Arn" --output text)
if [ -z "$CLUSTER_ROLE_ARN" ]; then
    handle_error "Failed to get Cluster Role ARN"
fi

echo "Creating EKS cluster (this will take 10-15 minutes)..."
CREATE_CLUSTER_OUTPUT=$(aws eks create-cluster \
  --name "$CLUSTER_NAME" \
  --role-arn "$CLUSTER_ROLE_ARN" \
  --resources-vpc-config
 subnetIds="$SUBNET_IDS",securityGroupIds="$SECURITY_GROUP_ID" \
```

```
  --tags Key=project,Value=doc-smith,Key=tutorial,Value=eks-gs)
check_command "$CREATE_CLUSTER_OUTPUT"
CREATED_RESOURCES+=("EKS Cluster: $CLUSTER_NAME")

echo "Waiting for EKS cluster to become active (this may take 10-15 minutes)..."
aws eks wait cluster-active --name "$CLUSTER_NAME"
if [ $? -ne 0 ]; then
    handle_error "Cluster creation failed or timed out"
fi
echo "EKS cluster is now active."

# Step 5: Configure kubectl
echo "Step 5: Configuring kubectl to communicate with the cluster..."

# Check if kubectl is installed
if ! check_kubectl; then
    echo "Will skip kubectl configuration steps but continue with the script."
    echo "You can manually configure kubectl later with: aws eks update-
kubeconfig --name \"$CLUSTER_NAME\""
else
    UPDATE_KUBECONFIG_OUTPUT=$(aws eks update-kubeconfig --name "$CLUSTER_NAME")
    check_command "$UPDATE_KUBECONFIG_OUTPUT"
    echo "kubectl configured successfully."

    # Test kubectl configuration
    echo "Testing kubectl configuration..."
    KUBECTL_TEST_OUTPUT=$(kubectl get svc 2>&1)
    if [ $? -ne 0 ]; then
        echo "Warning: kubectl configuration test failed. This might be due to
 permissions or network issues."
        echo "Error details: $KUBECTL_TEST_OUTPUT"
        echo "Continuing with script execution..."
    else
        echo "$KUBECTL_TEST_OUTPUT"
        echo "kubectl configuration test successful."
    fi
fi

# Step 6: Create managed node group
echo "Step 6: Creating managed node group: $NODEGROUP_NAME"

NODE_ROLE_ARN=$(aws iam get-role --role-name "$NODE_ROLE_NAME" --query "Role.Arn"
 --output text)
if [ -z "$NODE_ROLE_ARN" ]; then
```

```
    handle_error "Failed to get Node Role ARN"
fi

# Convert comma-separated subnet IDs to space-separated for the create-nodegroup
 command
SUBNET_IDS_ARRAY=(${SUBNET_IDS//,/ })

echo "Creating managed node group (this will take 5-10 minutes)..."
CREATE_NODEGROUP_OUTPUT=$(aws eks create-nodegroup \
  --cluster-name "$CLUSTER_NAME" \
  --nodegroup-name "$NODEGROUP_NAME" \
  --node-role "$NODE_ROLE_ARN" \
  --subnets "${SUBNET_IDS_ARRAY[@]}" \
  --tags Key=project,Value=doc-smith,Key=tutorial,Value=eks-gs)
check_command "$CREATE_NODEGROUP_OUTPUT"
CREATED_RESOURCES+=("EKS Node Group: $NODEGROUP_NAME")

echo "Waiting for node group to become active (this may take 5-10 minutes)..."
aws eks wait nodegroup-active --cluster-name "$CLUSTER_NAME" --nodegroup-name
 "$NODEGROUP_NAME"
if [ $? -ne 0 ]; then
    handle_error "Node group creation failed or timed out"
fi
echo "Node group is now active."

# Step 7: Verify nodes
echo "Step 7: Verifying nodes..."
echo "Waiting for nodes to register with the cluster (this may take a few
 minutes)..."
sleep 60  # Give nodes more time to register

# Check if kubectl is installed before attempting to use it
if ! check_kubectl; then
    echo "Cannot verify nodes without kubectl. Skipping this step."
    echo "You can manually verify nodes after installing kubectl with: kubectl
 get nodes"
else
    NODES_OUTPUT=$(kubectl get nodes 2>&1)
    if [ $? -ne 0 ]; then
        echo "Warning: Unable to get nodes. This might be due to permissions or
 the nodes are still registering."
        echo "Error details: $NODES_OUTPUT"
        echo "Continuing with script execution..."
    else
```

```
        echo "$NODES_OUTPUT"
        echo "Nodes verified successfully."
    fi
fi

# Step 8: View resources
echo "Step 8: Viewing cluster resources..."

echo "Cluster information:"
CLUSTER_INFO=$(aws eks describe-cluster --name "$CLUSTER_NAME")
echo "$CLUSTER_INFO"

echo "Node group information:"
NODEGROUP_INFO=$(aws eks describe-nodegroup --cluster-name "$CLUSTER_NAME" --
nodegroup-name "$NODEGROUP_NAME")
echo "$NODEGROUP_INFO"

echo "Kubernetes resources:"
if ! check_kubectl; then
    echo "Cannot list Kubernetes resources without kubectl. Skipping this step."
    echo "You can manually list resources after installing kubectl with: kubectl
 get all --all-namespaces"
else
    KUBE_RESOURCES=$(kubectl get all --all-namespaces 2>&1)
    if [ $? -ne 0 ]; then
        echo "Warning: Unable to get Kubernetes resources. This might be due to
 permissions."
        echo "Error details: $KUBE_RESOURCES"
        echo "Continuing with script execution..."
    else
        echo "$KUBE_RESOURCES"
    fi
fi

# Display summary of created resources
echo ""
echo "==========================================="
echo "RESOURCES CREATED"
echo "==========================================="
for resource in "${CREATED_RESOURCES[@]}"; do
    echo "- $resource"
done
echo "==========================================="
```

```
# Prompt for cleanup
echo ""
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "${CLEANUP_CHOICE,,}" == "y" ]]; then
    cleanup_resources
else
    echo "Resources will not be cleaned up. You can manually clean them up
 later."
    echo "To clean up resources, run the following commands:"
    echo "1. Delete node group: aws eks delete-nodegroup --cluster-name
 $CLUSTER_NAME --nodegroup-name $NODEGROUP_NAME"
    echo "2. Wait for node group deletion: aws eks wait nodegroup-deleted --
cluster-name $CLUSTER_NAME --nodegroup-name $NODEGROUP_NAME"
    echo "3. Delete cluster: aws eks delete-cluster --name $CLUSTER_NAME"
    echo "4. Wait for cluster deletion: aws eks wait cluster-deleted --name
 $CLUSTER_NAME"
    echo "5. Delete CloudFormation stack: aws cloudformation delete-stack --
stack-name $STACK_NAME"
    echo "6. Detach and delete IAM roles for the node group and cluster"
fi

echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateCluster
  - CreateNodegroup
  - CreateRole
  - CreateStack
  - DeleteCluster
  - DeleteNodegroup
  - DeleteRole

- DeleteStack
- DescribeCluster
- DescribeNodegroup
- DescribeStacks
- DetachRolePolicy
- GetCallerIdentity
- GetRole
- ListNodegroups
- UpdateKubeconfig
- Aspetta (CloudFormation)
- Attendi (Amazon EKS)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva allo streaming gestito

L'esempio di codice seguente mostra come:

- Crea un cluster MSK
- Crea autorizzazioni IAM per l'accesso MSK
- Crea una macchina client
- Ottieni broker bootstrap
- Configura la macchina client
- Crea un argomento e produce/consume dei dati
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon MSK Getting Started Tutorial Script - Version 8
# This script automates the steps in the Amazon MSK Getting Started tutorial
# It creates an MSK cluster, sets up IAM permissions, creates a client machine,
# and configures the client to interact with the cluster

# Set up logging
LOG_FILE="msk_tutorial_$(date +%Y%m%d_%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon MSK Getting Started Tutorial Script - Version 8"
echo "Logging to $LOG_FILE"
echo "============================================="

# Function to handle errors
handle_error() {
    echo "ERROR: $1"
    echo "Resources created so far:"
    if [ -n "$CLUSTER_ARN" ]; then echo "- MSK Cluster: $CLUSTER_ARN"; fi
    if [ -n "$POLICY_ARN" ]; then echo "- IAM Policy: $POLICY_ARN"; fi
    if [ -n "$ROLE_NAME" ]; then echo "- IAM Role: $ROLE_NAME"; fi
    if [ -n "$INSTANCE_PROFILE_NAME" ]; then echo "- IAM Instance Profile:
 $INSTANCE_PROFILE_NAME"; fi
    if [ -n "$CLIENT_SG_ID" ]; then echo "- Client Security Group:
 $CLIENT_SG_ID"; fi
    if [ -n "$INSTANCE_ID" ]; then echo "- EC2 Instance: $INSTANCE_ID"; fi
    if [ -n "$KEY_NAME" ]; then echo "- Key Pair: $KEY_NAME"; fi

    echo "Attempting to clean up resources..."
    cleanup_resources
```

```
    exit 1
}

# Function to check if a resource exists
resource_exists() {
    local resource_type="$1"
    local resource_id="$2"

    case "$resource_type" in
        "cluster")
            aws kafka describe-cluster --cluster-arn "$resource_id" &>/dev/null
            ;;
        "policy")
            aws iam get-policy --policy-arn "$resource_id" &>/dev/null
            ;;
        "role")
            aws iam get-role --role-name "$resource_id" &>/dev/null
            ;;
        "instance-profile")
            aws iam get-instance-profile --instance-profile-name "$resource_id"
 &>/dev/null
            ;;
        "security-group")
            aws ec2 describe-security-groups --group-ids "$resource_id" &>/dev/
null
            ;;
        "instance")
            aws ec2 describe-instances --instance-ids "$resource_id" --query
 'Reservations[0].Instances[0].State.Name' --output text | grep -v "terminated"
 &>/dev/null
            ;;
        "key-pair")
            aws ec2 describe-key-pairs --key-names "$resource_id" &>/dev/null
            ;;
    esac
}

# Function to remove security group references
remove_security_group_references() {
    local sg_id="$1"

    if [ -z "$sg_id" ]; then
        echo "No security group ID provided for reference removal"
        return
```

```
    fi

    echo "Removing security group references for $sg_id"

    # Get all security groups in the VPC that might reference our client security
 group
    local vpc_security_groups=$(aws ec2 describe-security-groups \
        --filters "Name=vpc-id,Values=$DEFAULT_VPC_ID" \
        --query 'SecurityGroups[].GroupId' \
        --output text 2>/dev/null)

    if [ -n "$vpc_security_groups" ]; then
        for other_sg in $vpc_security_groups; do
            if [ "$other_sg" != "$sg_id" ]; then
                echo "Checking security group $other_sg for references to $sg_id"

                # Get the security group details in JSON format
                local sg_details=$(aws ec2 describe-security-groups \
                    --group-ids "$other_sg" \
                    --output json 2>/dev/null)

                if [ -n "$sg_details" ]; then
                    # Check if our security group is referenced in inbound rules
                    local has_inbound_ref=$(echo "$sg_details" | grep -o
 "\"GroupId\": \"$sg_id\"" | head -1)

                    if [ -n "$has_inbound_ref" ]; then
                        echo "Found inbound rules in $other_sg referencing
 $sg_id, removing them..."

                        # Try to remove common rule types
                        echo "Attempting to remove all-traffic rule"
                        aws ec2 revoke-security-group-ingress \
                            --group-id "$other_sg" \
                            --protocol all \
                            --source-group "$sg_id" 2>/dev/null || echo "No all-
traffic rule to remove"

                        # Try to remove TCP rules on common ports
                        for port in 22 80 443 9092 9094 9096; do
                            aws ec2 revoke-security-group-ingress \
                                --group-id "$other_sg" \
                                --protocol tcp \
                                --port "$port" \
```

```
                                --source-group "$sg_id" 2>/dev/null || true
                        done

                        # Try to remove UDP rules
                        aws ec2 revoke-security-group-ingress \
                            --group-id "$other_sg" \
                            --protocol udp \
                            --source-group "$sg_id" 2>/dev/null || true
                    fi

                    # Check for outbound rules (less common but possible)
                    local has_outbound_ref=$(echo "$sg_details" | grep -A 20
 "IpPermissionsEgress" | grep -o "\"GroupId\": \"$sg_id\"" | head -1)

                    if [ -n "$has_outbound_ref" ]; then
                        echo "Found outbound rules in $other_sg referencing
 $sg_id, removing them..."

                        aws ec2 revoke-security-group-egress \
                            --group-id "$other_sg" \
                            --protocol all \
                            --source-group "$sg_id" 2>/dev/null || echo "No
 outbound all-traffic rule to remove"
                    fi
                fi
            fi
        done
    fi

    echo "Completed security group reference removal for $sg_id"
}

# Function to clean up resources
cleanup_resources() {
    echo "Cleaning up resources..."

    # Delete EC2 instance if it exists
    if [ -n "$INSTANCE_ID" ] && resource_exists "instance" "$INSTANCE_ID"; then
        echo "Terminating EC2 instance: $INSTANCE_ID"
        aws ec2 terminate-instances --instance-ids "$INSTANCE_ID" || echo "Failed
 to terminate instance"
        echo "Waiting for instance to terminate..."
        aws ec2 wait instance-terminated --instance-ids "$INSTANCE_ID" || echo
 "Failed to wait for instance termination"
```

```
    fi

    # Delete MSK cluster first (to remove dependencies on security group)
    if [ -n "$CLUSTER_ARN" ] && resource_exists "cluster" "$CLUSTER_ARN"; then
        echo "Deleting MSK cluster: $CLUSTER_ARN"
        aws kafka delete-cluster --cluster-arn "$CLUSTER_ARN" || echo "Failed to
delete cluster"

        # Wait a bit for the cluster deletion to start
        echo "Waiting 30 seconds for cluster deletion to begin..."
        sleep 30
    fi

    # Remove security group references before attempting deletion
    if [ -n "$CLIENT_SG_ID" ] && resource_exists "security-group"
"$CLIENT_SG_ID"; then
        remove_security_group_references "$CLIENT_SG_ID"

        echo "Deleting security group: $CLIENT_SG_ID"
        # Try multiple times with longer delays to ensure dependencies are
removed
        for i in {1..10}; do
            if aws ec2 delete-security-group --group-id "$CLIENT_SG_ID"; then
                echo "Security group deleted successfully"
                break
            fi
            echo "Failed to delete security group (attempt $i/10), retrying in 30
seconds..."
            sleep 30
        done
    fi

    # Delete key pair if it exists
    if [ -n "$KEY_NAME" ] && resource_exists "key-pair" "$KEY_NAME"; then
        echo "Deleting key pair: $KEY_NAME"
        aws ec2 delete-key-pair --key-name "$KEY_NAME" || echo "Failed to delete
key pair"
    fi

    # Remove role from instance profile
    if [ -n "$ROLE_NAME" ] && [ -n "$INSTANCE_PROFILE_NAME" ] && resource_exists
"instance-profile" "$INSTANCE_PROFILE_NAME"; then
        echo "Removing role from instance profile"
        aws iam remove-role-from-instance-profile \
```

```
            --instance-profile-name "$INSTANCE_PROFILE_NAME" \
            --role-name "$ROLE_NAME" || echo "Failed to remove role from instance
 profile"
    fi

    # Delete instance profile
    if [ -n "$INSTANCE_PROFILE_NAME" ] && resource_exists "instance-profile"
 "$INSTANCE_PROFILE_NAME"; then
        echo "Deleting instance profile: $INSTANCE_PROFILE_NAME"
        aws iam delete-instance-profile \
            --instance-profile-name "$INSTANCE_PROFILE_NAME" || echo "Failed to
 delete instance profile"
    fi

    # Detach policy from role
    if [ -n "$ROLE_NAME" ] && [ -n "$POLICY_ARN" ] && resource_exists "role"
 "$ROLE_NAME"; then
        echo "Detaching policy from role"
        aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "$POLICY_ARN" || echo "Failed to detach policy"
    fi

    # Delete role
    if [ -n "$ROLE_NAME" ] && resource_exists "role" "$ROLE_NAME"; then
        echo "Deleting role: $ROLE_NAME"
        aws iam delete-role --role-name "$ROLE_NAME" || echo "Failed to delete
 role"
    fi

    # Delete policy
    if [ -n "$POLICY_ARN" ] && resource_exists "policy" "$POLICY_ARN"; then
        echo "Deleting policy: $POLICY_ARN"
        aws iam delete-policy --policy-arn "$POLICY_ARN" || echo "Failed to
 delete policy"
    fi

    echo "Cleanup completed"
}

# Function to find a suitable subnet and instance type combination
find_suitable_subnet_and_instance_type() {
    local vpc_id="$1"
    local -a subnet_array=("${!2}")
```

```
    # List of instance types to try, in order of preference
    local instance_types=("t3.micro" "t2.micro" "t3.small" "t2.small")

    echo "Finding suitable subnet and instance type combination..."

    for instance_type in "${instance_types[@]}"; do
        echo "Trying instance type: $instance_type"

        for subnet_id in "${subnet_array[@]}"; do
            # Get the availability zone for this subnet
            local az=$(aws ec2 describe-subnets \
                --subnet-ids "$subnet_id" \
                --query 'Subnets[0].AvailabilityZone' \
                --output text)

            echo "  Checking subnet $subnet_id in AZ $az"

            # Check if this instance type is available in this AZ
            local available=$(aws ec2 describe-instance-type-offerings \
                --location-type availability-zone \
                --filters "Name=location,Values=$az" "Name=instance-type,Values=
$instance_type" \
                --query 'InstanceTypeOfferings[0].InstanceType' \
                --output text 2>/dev/null)

            if [ "$available" = "$instance_type" ]; then
                echo "  # Found suitable combination: $instance_type in $az
 (subnet: $subnet_id)"
                SELECTED_SUBNET_ID="$subnet_id"
                SELECTED_INSTANCE_TYPE="$instance_type"
                return 0
            else
                echo "  # $instance_type not available in $az"
            fi
        done
    done

    echo "ERROR: Could not find any suitable subnet and instance type
 combination"
    return 1
}

# Generate unique identifiers
```

```
RANDOM_SUFFIX=$(LC_ALL=C tr -dc 'a-z0-9' < /dev/urandom | fold -w 8 | head -n 1)
CLUSTER_NAME="MSKTutorialCluster-${RANDOM_SUFFIX}"
POLICY_NAME="msk-tutorial-policy-${RANDOM_SUFFIX}"
ROLE_NAME="msk-tutorial-role-${RANDOM_SUFFIX}"
INSTANCE_PROFILE_NAME="msk-tutorial-profile-${RANDOM_SUFFIX}"
SG_NAME="MSKClientSecurityGroup-${RANDOM_SUFFIX}"

echo "Using the following resource names:"
echo "- Cluster Name: $CLUSTER_NAME"
echo "- Policy Name: $POLICY_NAME"
echo "- Role Name: $ROLE_NAME"
echo "- Instance Profile Name: $INSTANCE_PROFILE_NAME"
echo "- Security Group Name: $SG_NAME"
echo "================================================"

# Step 1: Create an MSK Provisioned cluster
echo "Step 1: Creating MSK Provisioned cluster"

# Get the default VPC ID first
echo "Getting default VPC..."
DEFAULT_VPC_ID=$(aws ec2 describe-vpcs \
    --filters "Name=is-default,Values=true" \
    --query "Vpcs[0].VpcId" \
    --output text)

if [ -z "$DEFAULT_VPC_ID" ] || [ "$DEFAULT_VPC_ID" = "None" ]; then
    handle_error "Could not find default VPC. Please ensure you have a default
 VPC in your region."
fi

echo "Default VPC ID: $DEFAULT_VPC_ID"

# Get available subnets in the default VPC
echo "Getting available subnets in the default VPC..."
SUBNETS=$(aws ec2 describe-subnets \
    --filters "Name=vpc-id,Values=$DEFAULT_VPC_ID" "Name=default-for-
az,Values=true" \
    --query "Subnets[0:3].SubnetId" \
    --output text)

# Convert space-separated subnet IDs to an array
read -r -a SUBNET_ARRAY <<< "$SUBNETS"

if [ ${#SUBNET_ARRAY[@]} -lt 3 ]; then
```

```
    handle_error "Not enough subnets available in the default VPC. Need at least
 3 subnets, found ${#SUBNET_ARRAY[@]}."
fi

# Get default security group for the default VPC
echo "Getting default security group for the default VPC..."
DEFAULT_SG=$(aws ec2 describe-security-groups \
    --filters "Name=group-name,Values=default" "Name=vpc-id,Values=
$DEFAULT_VPC_ID" \
    --query "SecurityGroups[0].GroupId" \
    --output text)

if [ -z "$DEFAULT_SG" ] || [ "$DEFAULT_SG" = "None" ]; then
    handle_error "Could not find default security group for VPC $DEFAULT_VPC_ID"
fi

echo "Creating MSK cluster: $CLUSTER_NAME"
echo "Using VPC: $DEFAULT_VPC_ID"
echo "Using subnets: ${SUBNET_ARRAY[0]} ${SUBNET_ARRAY[1]} ${SUBNET_ARRAY[2]}"
echo "Using security group: $DEFAULT_SG"

# Create the MSK cluster with proper error handling
CLUSTER_RESPONSE=$(aws kafka create-cluster \
    --cluster-name "$CLUSTER_NAME" \
    --broker-node-group-info "{\"InstanceType\": \"kafka.t3.small\",
 \"ClientSubnets\": [\"${SUBNET_ARRAY[0]}\", \"${SUBNET_ARRAY[1]}\",
 \"${SUBNET_ARRAY[2]}\"], \"SecurityGroups\": [\"$DEFAULT_SG\"]}" \
    --kafka-version "3.6.0" \
    --number-of-broker-nodes 3 \
    --encryption-info "{\"EncryptionInTransit\": {\"InCluster\": true,
 \"ClientBroker\": \"TLS\"}}" \
    --tags project=doc-smith,tutorial=amazon-managed-streaming-for-apache-kafka-
gs 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create MSK cluster: $CLUSTER_RESPONSE"
fi

# Extract the cluster ARN using grep
CLUSTER_ARN=$(echo "$CLUSTER_RESPONSE" | grep -o '"ClusterArn": "[^"]*' | cut -
d'"' -f4)

if [ -z "$CLUSTER_ARN" ]; then
```

```
    handle_error "Failed to extract cluster ARN from response: $CLUSTER_RESPONSE"
fi

echo "MSK cluster creation initiated. ARN: $CLUSTER_ARN"
echo "Waiting for cluster to become active (this may take 15-20 minutes)..."

# Wait for the cluster to become active
while true; do
    CLUSTER_STATUS=$(aws kafka describe-cluster --cluster-arn "$CLUSTER_ARN" --
query "ClusterInfo.State" --output text 2>/dev/null)

    if [ $? -ne 0 ]; then
        echo "Failed to get cluster status. Retrying in 30 seconds..."
        sleep 30
        continue
    fi

    echo "Current cluster status: $CLUSTER_STATUS"

    if [ "$CLUSTER_STATUS" = "ACTIVE" ]; then
        echo "Cluster is now active!"
        break
    elif [ "$CLUSTER_STATUS" = "FAILED" ]; then
        handle_error "Cluster creation failed"
    fi

    echo "Still waiting for cluster to become active... (checking again in 60
 seconds)"
    sleep 60
done

echo "============================================="
# Step 2: Create an IAM role granting access to create topics on the Amazon MSK
 cluster
echo "Step 2: Creating IAM policy and role"

# Get account ID and region
ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
REGION=$(aws configure get region)
if [ -z "$REGION" ]; then
    REGION=$(aws ec2 describe-availability-zones --query
 'AvailabilityZones[0].RegionName' --output text)
fi
```

```
if [ -z "$ACCOUNT_ID" ] || [ -z "$REGION" ]; then
    handle_error "Could not determine AWS account ID or region"
fi

echo "Account ID: $ACCOUNT_ID"
echo "Region: $REGION"

# Create IAM policy
echo "Creating IAM policy: $POLICY_NAME"
POLICY_DOCUMENT="{
    \"Version\": \"2012-10-17\",
    \"Statement\": [
        {
            \"Effect\": \"Allow\",
            \"Action\": [
                \"kafka-cluster:Connect\",
                \"kafka-cluster:AlterCluster\",
                \"kafka-cluster:DescribeCluster\"
            ],
            \"Resource\": [
                \"$CLUSTER_ARN\"
            ]
        },
        {
            \"Effect\": \"Allow\",
            \"Action\": [
                \"kafka-cluster:*Topic*\",
                \"kafka-cluster:WriteData\",
                \"kafka-cluster:ReadData\"
            ],
            \"Resource\": [
                \"arn:aws:kafka:$REGION:$ACCOUNT_ID:topic/$CLUSTER_NAME/*\"
            ]
        },
        {
            \"Effect\": \"Allow\",
            \"Action\": [
                \"kafka-cluster:AlterGroup\",
                \"kafka-cluster:DescribeGroup\"
            ],
            \"Resource\": [
                \"arn:aws:kafka:$REGION:$ACCOUNT_ID:group/$CLUSTER_NAME/*\"
            ]
        }
```

```
    ]
}"

POLICY_RESPONSE=$(aws iam create-policy \
    --policy-name "$POLICY_NAME" \
    --policy-document "$POLICY_DOCUMENT" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create IAM policy: $POLICY_RESPONSE"
fi

# Extract the policy ARN using grep
POLICY_ARN=$(echo "$POLICY_RESPONSE" | grep -o '"Arn": "[^"]*' | cut -d'"' -f4)

if [ -z "$POLICY_ARN" ]; then
    handle_error "Failed to extract policy ARN from response: $POLICY_RESPONSE"
fi

aws iam tag-policy --policy-arn "$POLICY_ARN" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=amazon-managed-
streaming-for-apache-kafka-gs

echo "IAM policy created. ARN: $POLICY_ARN"

# Create IAM role for EC2
echo "Creating IAM role: $ROLE_NAME"
TRUST_POLICY="{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"ec2.amazonaws.com\"},\"Action\":\"sts:AssumeRole
\"}]}"

ROLE_RESPONSE=$(aws iam create-role \
    --role-name "$ROLE_NAME" \
    --assume-role-policy-document "$TRUST_POLICY" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create IAM role: $ROLE_RESPONSE"
fi

echo "IAM role created: $ROLE_NAME"
aws iam tag-role --role-name "$ROLE_NAME" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=amazon-managed-
streaming-for-apache-kafka-gs
```

```
# Attach policy to role
echo "Attaching policy to role"
ATTACH_RESPONSE=$(aws iam attach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "$POLICY_ARN" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to attach policy to role: $ATTACH_RESPONSE"
fi

echo "Policy attached to role"

# Create instance profile and add role to it
echo "Creating instance profile: $INSTANCE_PROFILE_NAME"
PROFILE_RESPONSE=$(aws iam create-instance-profile \
    --instance-profile-name "$INSTANCE_PROFILE_NAME" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create instance profile: $PROFILE_RESPONSE"
fi

echo "Instance profile created"

echo "Adding role to instance profile"
ADD_ROLE_RESPONSE=$(aws iam add-role-to-instance-profile \
    --instance-profile-name "$INSTANCE_PROFILE_NAME" \
    --role-name "$ROLE_NAME" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to add role to instance profile: $ADD_ROLE_RESPONSE"
fi

echo "Role added to instance profile"

# Wait a moment for IAM propagation
echo "Waiting 10 seconds for IAM propagation..."
sleep 10

echo "============================================="
```

```
# Step 3: Create a client machine
echo "Step 3: Creating client machine"

# Find a suitable subnet and instance type combination
if ! find_suitable_subnet_and_instance_type "$DEFAULT_VPC_ID" SUBNET_ARRAY[@];
 then
    handle_error "Could not find a suitable subnet and instance type combination"
fi

echo "Selected subnet: $SELECTED_SUBNET_ID"
echo "Selected instance type: $SELECTED_INSTANCE_TYPE"

# Verify the subnet is in the same VPC we're using
SUBNET_VPC_ID=$(aws ec2 describe-subnets \
    --subnet-ids "$SELECTED_SUBNET_ID" \
    --query 'Subnets[0].VpcId' \
    --output text)

if [ "$SUBNET_VPC_ID" != "$DEFAULT_VPC_ID" ]; then
    handle_error "Subnet VPC ($SUBNET_VPC_ID) does not match default VPC
 ($DEFAULT_VPC_ID)"
fi

echo "VPC ID: $SUBNET_VPC_ID"

# Get security group ID from the MSK cluster
echo "Getting security group ID from the MSK cluster"
MSK_SG_ID=$(aws kafka describe-cluster \
    --cluster-arn "$CLUSTER_ARN" \
    --query 'ClusterInfo.BrokerNodeGroupInfo.SecurityGroups[0]' \
    --output text)

if [ -z "$MSK_SG_ID" ] || [ "$MSK_SG_ID" = "None" ]; then
    handle_error "Failed to get security group ID from cluster"
fi

echo "MSK security group ID: $MSK_SG_ID"

# Create security group for client
echo "Creating security group for client: $SG_NAME"
CLIENT_SG_RESPONSE=$(aws ec2 create-security-group \
    --group-name "$SG_NAME" \
    --description "Security group for MSK client" \
    --vpc-id "$DEFAULT_VPC_ID" \
```

```
    --tag-specifications 'ResourceType=security-
group,Tags=[{Key=project,Value=doc-smith},{Key=tutorial,Value=amazon-managed-
streaming-for-apache-kafka-gs}]' 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create security group: $CLIENT_SG_RESPONSE"
fi

# Extract the security group ID using grep
CLIENT_SG_ID=$(echo "$CLIENT_SG_RESPONSE" | grep -o '"GroupId": "[^"]*' | cut -
d'"' -f4)

if [ -z "$CLIENT_SG_ID" ]; then
    handle_error "Failed to extract security group ID from response:
 $CLIENT_SG_RESPONSE"
fi

echo "Client security group created. ID: $CLIENT_SG_ID"

# Allow SSH access to client from your IP only
echo "Getting your public IP address"
MY_IP=$(curl -s https://checkip.amazonaws.com 2>/dev/null)

if [ -z "$MY_IP" ]; then
    echo "Warning: Could not determine your IP address. Using 0.0.0.0/0 (not
 recommended for production)"
    MY_IP="0.0.0.0/0"
else
    MY_IP="$MY_IP/32"
    echo "Your IP address: $MY_IP"
fi

echo "Adding SSH ingress rule to client security group"
SSH_RULE_RESPONSE=$(aws ec2 authorize-security-group-ingress \
    --group-id "$CLIENT_SG_ID" \
    --protocol tcp \
    --port 22 \
    --cidr "$MY_IP" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    echo "Warning: Failed to add SSH ingress rule: $SSH_RULE_RESPONSE"
```

```
    echo "You may need to manually add SSH access to security group
 $CLIENT_SG_ID"
fi

echo "SSH ingress rule added"

# Update MSK security group to allow traffic from client security group
echo "Adding ingress rule to MSK security group to allow traffic from client"
MSK_RULE_RESPONSE=$(aws ec2 authorize-security-group-ingress \
    --group-id "$MSK_SG_ID" \
    --protocol all \
    --source-group "$CLIENT_SG_ID" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    echo "Warning: Failed to add ingress rule to MSK security group:
 $MSK_RULE_RESPONSE"
    echo "You may need to manually add ingress rule to security group $MSK_SG_ID"
fi

echo "Ingress rule added to MSK security group"

# Create key pair
KEY_NAME="MSKKeyPair-${RANDOM_SUFFIX}"
echo "Creating key pair: $KEY_NAME"
KEY_RESPONSE=$(aws ec2 create-key-pair --key-name "$KEY_NAME" --tag-
specifications 'ResourceType=key-pair,Tags=[{Key=project,Value=doc-smith},
{Key=tutorial,Value=amazon-managed-streaming-for-apache-kafka-gs}]' --query
 'KeyMaterial' --output text 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create key pair: $KEY_RESPONSE"
fi

# Save the private key to a file
KEY_FILE="${KEY_NAME}.pem"
echo "$KEY_RESPONSE" > "$KEY_FILE"
chmod 400 "$KEY_FILE"

echo "Key pair created and saved to $KEY_FILE"

# Get the latest Amazon Linux 2 AMI
echo "Getting latest Amazon Linux 2 AMI ID"
```

```
AMI_ID=$(aws ec2 describe-images \
    --owners amazon \
    --filters "Name=name,Values=amzn2-ami-hvm-*-x86_64-gp2"
 "Name=state,Values=available" \
    --query "sort_by(Images, &CreationDate)[-1].ImageId" \
    --output text 2>/dev/null)

if [ -z "$AMI_ID" ] || [ "$AMI_ID" = "None" ]; then
    handle_error "Failed to get Amazon Linux 2 AMI ID"
fi

echo "Using AMI ID: $AMI_ID"

# Launch EC2 instance with the selected subnet and instance type
echo "Launching EC2 instance"
echo "Instance type: $SELECTED_INSTANCE_TYPE"
echo "Subnet: $SELECTED_SUBNET_ID"

INSTANCE_RESPONSE=$(aws ec2 run-instances \
    --image-id "$AMI_ID" \
    --instance-type "$SELECTED_INSTANCE_TYPE" \
    --key-name "$KEY_NAME" \
    --security-group-ids "$CLIENT_SG_ID" \
    --subnet-id "$SELECTED_SUBNET_ID" \
    --iam-instance-profile "Name=$INSTANCE_PROFILE_NAME" \
    --tag-specifications
 "ResourceType=instance,Tags=[{Key=Name,Value=MSKTutorialClient-
${RANDOM_SUFFIX}},{Key=project,Value=doc-smith},{Key=tutorial,Value=amazon-
managed-streaming-for-apache-kafka-gs}]" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to launch EC2 instance: $INSTANCE_RESPONSE"
fi

# Extract the instance ID using grep
INSTANCE_ID=$(echo "$INSTANCE_RESPONSE" | grep -o '"InstanceId": "[^"]*' | head
 -1 | cut -d'"' -f4)

if [ -z "$INSTANCE_ID" ]; then
    handle_error "Failed to extract instance ID from response:
 $INSTANCE_RESPONSE"
fi
```

```
echo "EC2 instance launched successfully. ID: $INSTANCE_ID"
echo "Waiting for instance to be running..."

# Wait for the instance to be running
aws ec2 wait instance-running --instance-ids "$INSTANCE_ID"

if [ $? -ne 0 ]; then
    handle_error "Instance failed to reach running state"
fi

# Wait a bit more for the instance to initialize
echo "Instance is running. Waiting 30 seconds for initialization..."
sleep 30

# Get public DNS name of instance
CLIENT_DNS=$(aws ec2 describe-instances \
    --instance-ids "$INSTANCE_ID" \
    --query 'Reservations[0].Instances[0].PublicDnsName' \
    --output text)

if [ -z "$CLIENT_DNS" ] || [ "$CLIENT_DNS" = "None" ]; then
    echo "Warning: Could not get public DNS name for instance. Trying public
 IP..."
    CLIENT_DNS=$(aws ec2 describe-instances \
        --instance-ids "$INSTANCE_ID" \
        --query 'Reservations[0].Instances[0].PublicIpAddress' \
        --output text)

    if [ -z "$CLIENT_DNS" ] || [ "$CLIENT_DNS" = "None" ]; then
        handle_error "Failed to get public DNS name or IP address for instance"
    fi
fi

echo "Client instance DNS/IP: $CLIENT_DNS"
echo "============================================="
# Get bootstrap brokers with improved logic
echo "Getting bootstrap brokers"
MAX_RETRIES=10
RETRY_COUNT=0
BOOTSTRAP_BROKERS=""
AUTH_METHOD=""

while [ -z "$BOOTSTRAP_BROKERS" ] || [ "$BOOTSTRAP_BROKERS" = "None" ]; do
    # Get the full bootstrap brokers response
```

```
    BOOTSTRAP_RESPONSE=$(aws kafka get-bootstrap-brokers \
        --cluster-arn "$CLUSTER_ARN" 2>/dev/null)

    if [ $? -eq 0 ] && [ -n "$BOOTSTRAP_RESPONSE" ]; then
        # Try to get IAM authentication brokers first using grep
        BOOTSTRAP_BROKERS=$(echo "$BOOTSTRAP_RESPONSE" | grep -o
 '"BootstrapBrokerStringSaslIam": "[^"]*' | cut -d'"' -f4)
        if [ -n "$BOOTSTRAP_BROKERS" ]; then
            AUTH_METHOD="IAM"
        else
            # Fall back to TLS authentication
            BOOTSTRAP_BROKERS=$(echo "$BOOTSTRAP_RESPONSE" | grep -o
 '"BootstrapBrokerStringTls": "[^"]*' | cut -d'"' -f4)
            if [ -n "$BOOTSTRAP_BROKERS" ]; then
                AUTH_METHOD="TLS"
            fi
        fi
    fi

    RETRY_COUNT=$((RETRY_COUNT + 1))

    if [ "$RETRY_COUNT" -ge "$MAX_RETRIES" ]; then
        echo "Warning: Could not get bootstrap brokers after $MAX_RETRIES
 attempts."
        echo "You may need to manually retrieve them later using:"
        echo "aws kafka get-bootstrap-brokers --cluster-arn $CLUSTER_ARN"
        BOOTSTRAP_BROKERS="BOOTSTRAP_BROKERS_NOT_AVAILABLE"
        AUTH_METHOD="UNKNOWN"
        break
    fi

    if [ -z "$BOOTSTRAP_BROKERS" ] || [ "$BOOTSTRAP_BROKERS" = "None" ]; then
        echo "Bootstrap brokers not available yet. Retrying in 30 seconds...
 (Attempt $RETRY_COUNT/$MAX_RETRIES)"
        sleep 30
    fi
done

echo "Bootstrap brokers: $BOOTSTRAP_BROKERS"
echo "Authentication method: $AUTH_METHOD"
echo "============================================="

# Create setup script for the client machine
echo "Creating setup script for the client machine"
```

```
cat > setup_client.sh << 'EOF'
#!/bin/bash

# Set up logging
LOG_FILE="client_setup_$(date +%Y%m%d_%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting client setup"
echo "============================================="

# Install Java
echo "Installing Java"
sudo yum -y install java-11

# Set environment variables
echo "Setting up environment variables"
export KAFKA_VERSION="3.6.0"
echo "KAFKA_VERSION=$KAFKA_VERSION"

# Download and extract Apache Kafka
echo "Downloading Apache Kafka"
wget https://archive.apache.org/dist/kafka/$KAFKA_VERSION/kafka_2.13-
$KAFKA_VERSION.tgz
if [ $? -ne 0 ]; then
    echo "Failed to download Kafka. Trying alternative mirror..."
    wget https://www.apache.org/dyn/closer.cgi?path=/kafka/$KAFKA_VERSION/
kafka_2.13-$KAFKA_VERSION.tgz
fi

echo "Extracting Kafka"
tar -xzf kafka_2.13-$KAFKA_VERSION.tgz
export KAFKA_ROOT=$(pwd)/kafka_2.13-$KAFKA_VERSION
echo "KAFKA_ROOT=$KAFKA_ROOT"

# Download the MSK IAM authentication package (needed for both IAM and TLS)
echo "Downloading MSK IAM authentication package"
cd $KAFKA_ROOT/libs
wget https://github.com/aws/aws-msk-iam-auth/releases/latest/download/aws-msk-
iam-auth-1.1.6-all.jar
if [ $? -ne 0 ]; then
    echo "Failed to download specific version. Trying to get latest version..."
    LATEST_VERSION=$(curl -s https://api.github.com/repos/aws/aws-msk-iam-auth/
releases/latest | grep -o '"tag_name": "[^"]*' | cut -d'"' -f4)
```

```
    wget https://github.com/aws/aws-msk-iam-auth/releases/download/
$LATEST_VERSION/aws-msk-iam-auth-$LATEST_VERSION-all.jar
    if [ $? -ne 0 ]; then
        echo "Failed to download IAM auth package. Please check the URL and try
 again."
        exit 1
    fi
    export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-$LATEST_VERSION-all.jar
else
    export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-1.1.6-all.jar
fi
echo "CLASSPATH=$CLASSPATH"

# Create client properties file based on authentication method
echo "Creating client properties file"
cd $KAFKA_ROOT/config

# The AUTH_METHOD_PLACEHOLDER will be replaced by the script
AUTH_METHOD="AUTH_METHOD_PLACEHOLDER"

if [ "$AUTH_METHOD" = "IAM" ]; then
    echo "Configuring for IAM authentication"
    cat > client.properties << 'EOT'
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required;
sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandler
EOT
elif [ "$AUTH_METHOD" = "TLS" ]; then
    echo "Configuring for TLS authentication"
    cat > client.properties << 'EOT'
security.protocol=SSL
EOT
else
    echo "Unknown authentication method. Creating basic TLS configuration."
    cat > client.properties << 'EOT'
security.protocol=SSL
EOT
fi

echo "Client setup completed"
echo "============================================="

# Create a script to set up environment variables
```

```
cat > ~/setup_env.sh << 'EOT'
#!/bin/bash
export KAFKA_VERSION="3.6.0"
export KAFKA_ROOT=~/kafka_2.13-$KAFKA_VERSION
export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-1.1.6-all.jar
export BOOTSTRAP_SERVER="BOOTSTRAP_SERVER_PLACEHOLDER"
export AUTH_METHOD="AUTH_METHOD_PLACEHOLDER"

echo "Environment variables set:"
echo "KAFKA_VERSION=$KAFKA_VERSION"
echo "KAFKA_ROOT=$KAFKA_ROOT"
echo "CLASSPATH=$CLASSPATH"
echo "BOOTSTRAP_SERVER=$BOOTSTRAP_SERVER"
echo "AUTH_METHOD=$AUTH_METHOD"
EOT

chmod +x ~/setup_env.sh

echo "Created environment setup script: ~/setup_env.sh"
echo "Run 'source ~/setup_env.sh' to set up your environment"
EOF

# Replace placeholders in the setup script
if [ -n "$BOOTSTRAP_BROKERS" ] && [ "$BOOTSTRAP_BROKERS" != "None" ] &&
 [ "$BOOTSTRAP_BROKERS" != "BOOTSTRAP_BROKERS_NOT_AVAILABLE" ]; then
    sed -i "s|BOOTSTRAP_SERVER_PLACEHOLDER|$BOOTSTRAP_BROKERS|g" setup_client.sh
else
    # If bootstrap brokers are not available, provide instructions to get them
    sed -i "s|BOOTSTRAP_SERVER_PLACEHOLDER|\$(aws kafka get-bootstrap-brokers
 --cluster-arn $CLUSTER_ARN --query 'BootstrapBrokerStringTls' --output text)|g"
 setup_client.sh
fi

# Replace auth method placeholder
sed -i "s|AUTH_METHOD_PLACEHOLDER|$AUTH_METHOD|g" setup_client.sh

echo "Setup script created"
echo "============================================="

# Display summary of created resources
echo ""
echo "============================================="
echo "RESOURCE SUMMARY"
echo "============================================="
```

```
echo "MSK Cluster ARN: $CLUSTER_ARN"
echo "MSK Cluster Name: $CLUSTER_NAME"
echo "Authentication Method: $AUTH_METHOD"
echo "IAM Policy ARN: $POLICY_ARN"
echo "IAM Role Name: $ROLE_NAME"
echo "IAM Instance Profile: $INSTANCE_PROFILE_NAME"
echo "Client Security Group: $CLIENT_SG_ID"
echo "EC2 Instance ID: $INSTANCE_ID"
echo "EC2 Instance Type: $SELECTED_INSTANCE_TYPE"
echo "EC2 Instance DNS: $CLIENT_DNS"
echo "Key Pair: $KEY_NAME (saved to $KEY_FILE)"
echo "Bootstrap Brokers: $BOOTSTRAP_BROKERS"
echo "================================================"
echo ""

# Instructions for connecting to the instance and setting up the client
echo "NEXT STEPS:"
echo "1. Connect to your EC2 instance:"
echo "   ssh -i $KEY_FILE ec2-user@$CLIENT_DNS"
echo ""
echo "2. Upload the setup script to your instance:"
echo "   scp -i $KEY_FILE setup_client.sh ec2-user@$CLIENT_DNS:~/"
echo ""
echo "3. Run the setup script on your instance:"
echo "   ssh -i $KEY_FILE ec2-user@$CLIENT_DNS 'chmod +x ~/setup_client.sh && ~/
setup_client.sh'"
echo ""
echo "4. Source the environment setup script:"
echo "   source ~/setup_env.sh"
echo ""

# Provide different instructions based on authentication method
if [ "$AUTH_METHOD" = "IAM" ]; then
    echo "5. Create a Kafka topic (using IAM authentication):"
    echo "   \$KAFKA_ROOT/bin/kafka-topics.sh --create \\"
    echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
    echo "     --command-config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --replication-factor 3 \\"
    echo "     --partitions 1 \\"
    echo "     --topic MSKTutorialTopic"
    echo ""
    echo "6. Start a producer:"
    echo "   \$KAFKA_ROOT/bin/kafka-console-producer.sh \\"
    echo "     --broker-list \$BOOTSTRAP_SERVER \\"
```

```
    echo "     --producer.config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --topic MSKTutorialTopic"
    echo ""
    echo "7. Start a consumer:"
    echo "   \$KAFKA_ROOT/bin/kafka-console-consumer.sh \\"
    echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
    echo "     --consumer.config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --topic MSKTutorialTopic \\"
    echo "     --from-beginning"
elif [ "$AUTH_METHOD" = "TLS" ]; then
    echo "5. Create a Kafka topic (using TLS authentication):"
    echo "   \$KAFKA_ROOT/bin/kafka-topics.sh --create \\"
    echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
    echo "     --command-config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --replication-factor 3 \\"
    echo "     --partitions 1 \\"
    echo "     --topic MSKTutorialTopic"
    echo ""
    echo "6. Start a producer:"
    echo "   \$KAFKA_ROOT/bin/kafka-console-producer.sh \\"
    echo "     --broker-list \$BOOTSTRAP_SERVER \\"
    echo "     --producer.config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --topic MSKTutorialTopic"
    echo ""
    echo "7. Start a consumer:"
    echo "   \$KAFKA_ROOT/bin/kafka-console-consumer.sh \\"
    echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
    echo "     --consumer.config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --topic MSKTutorialTopic \\"
    echo "     --from-beginning"
else
    echo "5. Manually retrieve bootstrap brokers and configure authentication:"
    echo "   aws kafka get-bootstrap-brokers --cluster-arn $CLUSTER_ARN"
fi

echo ""
echo "8. Verify the topic was created:"
echo "   \$KAFKA_ROOT/bin/kafka-topics.sh --list \\"
echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
echo "     --command-config \$KAFKA_ROOT/config/client.properties"
echo "============================================="
echo ""

# Ask if user wants to clean up resources
```

```
echo ""
echo "============================================="
echo "CLEANUP CONFIRMATION"
echo "============================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ $CLEANUP_CHOICE =~ ^[Yy]$ ]]; then
    cleanup_resources
    echo "All resources have been cleaned up."
else
    echo "Resources will not be cleaned up. You can manually clean them up
 later."
    echo "To clean up resources, run this script again and choose 'y' when
 prompted."
fi

echo "Script completed successfully!"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AddRoleToInstanceProfile
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress
  - CreateCluster
  - CreateInstanceProfile
  - CreateKeyPair
  - CreatePolicy
  - CreateRole
  - CreateSecurityGroup
  - DeleteCluster
  - DeleteInstanceProfile
  - DeleteKeyPair
  - DeletePolicy
  - DeleteRole
  - DeleteSecurityGroup

- DescribeAvailabilityZones
- DescribeCluster
- DescribeImages
- DescribeInstanceTypeOfferings
- DescribeInstances
- DescribeKeyPairs
- DescribeSecurityGroups
- DescribeSubnets
- DescribeVpcs
- DetachRolePolicy
- GetBootstrapBrokers
- GetCallerIdentity
- GetInstanceProfile
- GetPolicy
- GetRole
- RemoveRoleFromInstanceProfile
- RevokeSecurityGroupEgress
- RevokeSecurityGroupIngress
- RunInstances
- TerminateInstances
- Attendi

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva ai cluster di data warehouse con provisioning

L'esempio di codice seguente mostra come:

- Crea un cluster Redshift
- Crea un ruolo IAM per l'accesso a S3
- Crea tabelle e carica dati

- Esegui query di esempio
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon Redshift Provisioned Cluster Tutorial Script
# This script creates a Redshift cluster, loads sample data, runs queries, and
 cleans up resources
# Version 4: Security improvements and best practices

set -euo pipefail

# Set up logging
LOG_FILE="redshift_tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon Redshift tutorial script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Function to handle errors
handle_error() {
    echo "ERROR: $1" >&2
    echo "Resources created so far:"
    if [ -n "${CLUSTER_ID:-}" ]; then echo "- Redshift Cluster: $CLUSTER_ID"; fi
    if [ -n "${ROLE_NAME:-}" ]; then echo "- IAM Role: $ROLE_NAME"; fi

    echo "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}
```

```
# Function to clean up resources
cleanup_resources() {
    echo "Cleaning up resources..."

    # Delete the cluster if it exists
    if [ -n "${CLUSTER_ID:-}" ]; then
        echo "Deleting Redshift cluster: $CLUSTER_ID"
        aws redshift delete-cluster --cluster-identifier "$CLUSTER_ID" --skip-
final-cluster-snapshot 2>/dev/null || true
        echo "Waiting for cluster deletion to complete..."
        aws redshift wait cluster-deleted --cluster-identifier "$CLUSTER_ID" 2>/
dev/null || true
        echo "Cluster deleted successfully."
    fi

    # Delete the IAM role if it exists
    if [ -n "${ROLE_NAME:-}" ]; then
        echo "Removing IAM role policy..."
        aws iam delete-role-policy --role-name "$ROLE_NAME" --policy-name
 RedshiftS3Access 2>/dev/null || true

        echo "Deleting IAM role: $ROLE_NAME"
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi

    # Clean up temporary files
    rm -f redshift-trust-policy.json redshift-s3-policy.json

    echo "Cleanup completed."
}

# Trap errors and cleanup
trap 'handle_error "Script interrupted"' INT TERM

# Function to wait for SQL statement to complete
wait_for_statement() {
    local statement_id=$1
    local max_attempts=30
    local attempt=1
    local status=""

    echo "Waiting for statement $statement_id to complete..."
```

```
    while [ $attempt -le $max_attempts ]; do
        status=$(aws redshift-data describe-statement --id "$statement_id" --
query 'Status' --output text 2>/dev/null || echo "")

        if [ "$status" == "FINISHED" ]; then
            echo "Statement completed successfully."
            return 0
        elif [ "$status" == "FAILED" ]; then
            local error=$(aws redshift-data describe-statement --id
 "$statement_id" --query 'Error' --output text 2>/dev/null || echo "Unknown
 error")
            echo "Statement failed with error: $error" >&2
            return 1
        elif [ "$status" == "ABORTED" ]; then
            echo "Statement was aborted." >&2
            return 1
        fi

        echo "Statement status: $status. Waiting... (Attempt $attempt/
$max_attempts)"
        sleep 10
        ((attempt++))
    done

    echo "Timed out waiting for statement to complete." >&2
    return 1
}

# Function to check if IAM role is attached to cluster
check_role_attached() {
    local role_arn=$1
    local max_attempts=10
    local attempt=1

    echo "Checking if IAM role is attached to the cluster..."

    while [ $attempt -le $max_attempts ]; do
        local status=$(aws redshift describe-clusters \
            --cluster-identifier "$CLUSTER_ID" \
            --query "Clusters[0].IamRoles[?IamRoleArn=='$role_arn'].ApplyStatus"
 \
            --output text 2>/dev/null || echo "")

        if [ "$status" == "in-sync" ]; then
```

```
            echo "IAM role is successfully attached to the cluster."
            return 0
        fi

        echo "IAM role status: $status. Waiting... (Attempt $attempt/
$max_attempts)"
        sleep 30
        ((attempt++))
    done

    echo "Timed out waiting for IAM role to be attached." >&2
    return 1
}

# Validate required commands
for cmd in aws jq; do
    if ! command -v "$cmd" &> /dev/null; then
        handle_error "Required command '$cmd' not found. Please install it and
 try again."
    fi
done

# Validate AWS credentials
if ! aws sts get-caller-identity &>/dev/null; then
    handle_error "AWS credentials not configured or invalid"
fi

# Variables to track created resources
CLUSTER_ID="examplecluster"
ROLE_NAME="RedshiftS3Role-$(date +%s)"
DB_NAME="dev"
DB_USER="awsuser"

# Generate secure password using AWS Secrets Manager or random string
if command -v openssl &> /dev/null; then
    DB_PASSWORD=$(openssl rand -base64 32 | tr -d "=+/" | cut -c1-20)
else
    DB_PASSWORD="TempPass$(date +%s | md5sum | cut -c1-20)"
fi

# Validate password meets requirements
if [ ${#DB_PASSWORD} -lt 8 ]; then
    handle_error "Generated password does not meet minimum length requirement"
fi
```

```
# Store password securely (optional: use AWS Secrets Manager in production)
echo "Generated database password (store securely): $DB_PASSWORD"

echo "=== Step 1: Creating Amazon Redshift Cluster ==="

# Create the Redshift cluster with encryption and audit logging enabled
echo "Creating Redshift cluster: $CLUSTER_ID"
CLUSTER_RESULT=$(aws redshift create-cluster \
  --cluster-identifier "$CLUSTER_ID" \
  --node-type ra3.xlplus \
  --number-of-nodes 2 \
  --master-username "$DB_USER" \
  --master-user-password "$DB_PASSWORD" \
  --db-name "$DB_NAME" \
  --port 5439 \
  --encrypted \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-provisioned \
  2>&1) || handle_error "Failed to create Redshift cluster"

echo "$CLUSTER_RESULT"
echo "Waiting for cluster to become available..."

# Wait for the cluster to be available
aws redshift wait cluster-available --cluster-identifier "$CLUSTER_ID" ||
 handle_error "Timeout waiting for cluster to become available"

# Get cluster status to confirm
CLUSTER_STATUS=$(aws redshift describe-clusters \
  --cluster-identifier "$CLUSTER_ID" \
  --query 'Clusters[0].ClusterStatus' \
  --output text)

echo "Cluster status: $CLUSTER_STATUS"

echo "=== Step 2: Creating IAM Role for S3 Access ==="

# Create trust policy file with restricted permissions
echo "Creating trust policy for Redshift"
cat > redshift-trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

chmod 600 redshift-trust-policy.json

# Create IAM role
echo "Creating IAM role: $ROLE_NAME"
ROLE_RESULT=$(aws iam create-role \
  --role-name "$ROLE_NAME" \
  --assume-role-policy-document file://redshift-trust-policy.json 2>&1) ||
 handle_error "Failed to create IAM role"

echo "$ROLE_RESULT"

# Get the role ARN
ROLE_ARN=$(aws iam get-role --role-name "$ROLE_NAME" --query 'Role.Arn' --output
 text)
echo "Role ARN: $ROLE_ARN"

# Tag the IAM role
echo "Tagging IAM role: $ROLE_NAME"
aws iam tag-role --role-name "$ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=redshift-provisioned

# Create policy document for S3 access with principle of least privilege
echo "Creating S3 access policy"
cat > redshift-s3-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
```

```
        "arn:aws:s3:::redshift-downloads",
        "arn:aws:s3:::redshift-downloads/*"
      ]
    }
  ]
}
EOF

chmod 600 redshift-s3-policy.json

# Attach policy to role
echo "Attaching S3 access policy to role"
POLICY_RESULT=$(aws iam put-role-policy \
  --role-name "$ROLE_NAME" \
  --policy-name RedshiftS3Access \
  --policy-document file://redshift-s3-policy.json 2>&1) || handle_error "Failed
 to attach policy to role"

echo "$POLICY_RESULT"

# Attach role to cluster
echo "Attaching IAM role to Redshift cluster"
ATTACH_ROLE_RESULT=$(aws redshift modify-cluster-iam-roles \
  --cluster-identifier "$CLUSTER_ID" \
  --add-iam-roles "$ROLE_ARN" 2>&1) || handle_error "Failed to attach role to
 cluster"

echo "$ATTACH_ROLE_RESULT"

# Wait for the role to be attached
echo "Waiting for IAM role to be attached to the cluster..."
if ! check_role_attached "$ROLE_ARN"; then
    handle_error "Failed to attach IAM role to cluster"
fi

echo "=== Step 3: Getting Cluster Connection Information ==="

# Get cluster endpoint
CLUSTER_INFO=$(aws redshift describe-clusters \
  --cluster-identifier "$CLUSTER_ID" \
  --query 'Clusters[0].Endpoint.{Address:Address,Port:Port}' \
  --output json)

echo "Cluster endpoint information:"
```

```
echo "$CLUSTER_INFO"

echo "=== Step 4: Creating Tables and Loading Data ==="

echo "Creating sales table"
SALES_TABLE_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "DROP TABLE IF EXISTS sales; CREATE TABLE sales(salesid integer not null,
 listid integer not null distkey, sellerid integer not null, buyerid integer
 not null, eventid integer not null, dateid smallint not null sortkey, qtysold
 smallint not null, pricepaid decimal(8,2), commission decimal(8,2), saletime
 timestamp);" \
  --query 'Id' --output text)

echo "Sales table creation statement ID: $SALES_TABLE_ID"

# Wait for statement to complete
if ! wait_for_statement "$SALES_TABLE_ID"; then
    handle_error "Failed to create sales table"
fi

echo "Creating date table"
DATE_TABLE_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "DROP TABLE IF EXISTS date; CREATE TABLE date(dateid smallint not null
 distkey sortkey, caldate date not null, day character(3) not null, week smallint
 not null, month character(5) not null, qtr character(5) not null, year smallint
 not null, holiday boolean default('N'));" \
  --query 'Id' --output text)

echo "Date table creation statement ID: $DATE_TABLE_ID"

# Wait for statement to complete
if ! wait_for_statement "$DATE_TABLE_ID"; then
    handle_error "Failed to create date table"
fi

echo "Loading data into sales table"
SALES_LOAD_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
```

```
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "COPY sales FROM 's3://redshift-downloads/tickit/sales_tab.txt' DELIMITER
 '\t' TIMEFORMAT 'MM/DD/YYYY HH:MI:SS' REGION 'us-east-1' IAM_ROLE '$ROLE_ARN';"
 \
  --query 'Id' --output text)

echo "Sales data load statement ID: $SALES_LOAD_ID"

# Wait for statement to complete
if ! wait_for_statement "$SALES_LOAD_ID"; then
    handle_error "Failed to load data into sales table"
fi

echo "Loading data into date table"
DATE_LOAD_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "COPY date FROM 's3://redshift-downloads/tickit/date2008_pipe.txt'
 DELIMITER '|' REGION 'us-east-1' IAM_ROLE '$ROLE_ARN';" \
  --query 'Id' --output text)

echo "Date data load statement ID: $DATE_LOAD_ID"

# Wait for statement to complete
if ! wait_for_statement "$DATE_LOAD_ID"; then
    handle_error "Failed to load data into date table"
fi

echo "=== Step 5: Running Example Queries ==="

echo "Running query: Get definition for the sales table"
QUERY1_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "SELECT * FROM pg_table_def WHERE tablename = 'sales';" \
  --query 'Id' --output text)

echo "Query 1 statement ID: $QUERY1_ID"

# Wait for statement to complete
if ! wait_for_statement "$QUERY1_ID"; then
```

```
    handle_error "Query 1 failed"
fi

# Get and display results
echo "Query 1 results (first 10 rows):"
aws redshift-data get-statement-result --id "$QUERY1_ID" --max-items 10

echo "Running query: Find total sales on a given calendar date"
QUERY2_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "SELECT sum(qtysold) FROM sales, date WHERE sales.dateid = date.dateid
 AND caldate = '2008-01-05';" \
  --query 'Id' --output text)

echo "Query 2 statement ID: $QUERY2_ID"

# Wait for statement to complete
if ! wait_for_statement "$QUERY2_ID"; then
    handle_error "Query 2 failed"
fi

# Get and display results
echo "Query 2 results:"
aws redshift-data get-statement-result --id "$QUERY2_ID"

echo "=== Tutorial Complete ==="
echo "The following resources were created:"
echo "- Redshift Cluster: $CLUSTER_ID"
echo "- IAM Role: $ROLE_NAME"

echo ""
echo "========================================="
echo "CLEANUP CONFIRMATION"
echo "========================================="
echo "Cleaning up all created resources..."
cleanup_resources
echo "All resources have been cleaned up."

# Securely clear password from memory
DB_PASSWORD=""
```

```
echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - CreateCluster
  - CreateRole
  - DeleteCluster
  - DeleteRole
  - DeleteRolePolicy
  - DescribeClusters
  - GetRole
  - ModifyClusterIamRoles
  - PutRolePolicy
  - Attendi

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva all'orchestrazione del flusso di lavoro

L'esempio di codice seguente mostra come:

- Creare un ruolo IAM per Step Functions
- Crea la tua prima macchina a stati
- Avvia l'esecuzione della tua macchina a stati
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS Step Functions Getting Started Tutorial Script
# This script creates and runs a Step Functions state machine based on the AWS
 Step Functions Getting Started tutorial

set -euo pipefail

# Security: Restrict umask to prevent unintended file permissions
umask 077

# Parse command line arguments
AUTO_CLEANUP=true
while [[ $# -gt 0 ]]; do
    case $1 in
        --auto-cleanup)
            AUTO_CLEANUP=true
            shift
            ;;
        -h|--help)
            echo "Usage: $0 [--auto-cleanup] [--help]"
            echo "  --auto-cleanup: Automatically clean up resources without
 prompting"
            echo "  --help: Show this help message"
            exit 0
            ;;
        *)
            echo "Unknown option: $1"
            echo "Use --help for usage information"
            exit 1
            ;;
```

```
    esac
done

# Set up logging with secure permissions
LOG_FILE="step-functions-tutorial.log"
touch "$LOG_FILE"
chmod 600 "$LOG_FILE"

# Security: Use process substitution with explicit FD cleanup
exec 3>&1 4>&2
exec > >(tee -a "$LOG_FILE") 2>&1
trap 'exec 1>&3 2>&4 3>&- 4>&-' EXIT

echo "Starting AWS Step Functions Getting Started Tutorial..."
echo "Logging to $LOG_FILE"

# Verify AWS CLI is installed and configured
if ! command -v aws &> /dev/null; then
    echo "ERROR: AWS CLI is not installed"
    exit 1
fi

# Verify AWS credentials are configured
if ! aws sts get-caller-identity &> /dev/null; then
    echo "ERROR: AWS credentials are not configured or invalid"
    exit 1
fi

# Check if jq is available for better JSON parsing
if ! command -v jq &> /dev/null; then
    echo "WARNING: jq is not installed. Using basic JSON parsing which may be
 less reliable."
    echo "Consider installing jq for better error handling: brew install jq
 (macOS) or apt-get install jq (Ubuntu)"
    USE_JQ=false
else
    USE_JQ=true
fi

# Use fixed region that supports Amazon Comprehend
CURRENT_REGION="us-west-2"
echo "Using fixed AWS region: $CURRENT_REGION (supports Amazon Comprehend)"

# Set AWS CLI to use the fixed region for all commands
```

```
export AWS_DEFAULT_REGION="$CURRENT_REGION"
export AWS_REGION="$CURRENT_REGION"

# Amazon Comprehend is available in us-west-2, so we can always enable it
echo "Amazon Comprehend is available in region $CURRENT_REGION"
SKIP_COMPREHEND=false

# Security: Initialize all resource variables
STATE_MACHINE_ARN=""
ROLE_NAME=""
ROLE_ARN=""
POLICY_ARN=""
STEPFUNCTIONS_POLICY_ARN=""
EXECUTION_ARN=""
EXECUTION2_ARN=""
EXECUTION3_ARN=""

# Performance: Cache for AWS API calls to reduce redundant requests
declare -A API_CACHE

# Function to make cached AWS CLI calls
aws_call_cached() {
    local cache_key="$1"
    shift

    if [[ -v API_CACHE["$cache_key"] ]]; then
        echo "${API_CACHE[$cache_key]}"
        return 0
    fi

    local result
    result=$(aws "$@" 2>&1) || return $?
    API_CACHE["$cache_key"]="$result"
    echo "$result"
}

# Function to check for API errors in JSON response with optimized jq usage
check_api_error() {
    local response="$1"
    local operation="$2"

    if [[ "$USE_JQ" == "true" ]]; then
        # Use jq for more reliable JSON parsing with efficient error detection
```

```
        if echo "$response" | jq -e '.Error // .error // empty' > /dev/null 2>&1;
 then
            local error_message=$(echo "$response" | jq -r
 '.Error.Message // .Error.Code // .error // "Unknown error"' 2>/dev/null)
            handle_error "$operation failed: $error_message"
        fi
    else
        # Fallback to grep-based detection with optimized pattern
        if echo "$response" | grep -qE '"[Ee]rror":|"error":'; then
            handle_error "$operation failed: $response"
        fi
    fi
}

# Function to extract JSON field efficiently
extract_json_field() {
    local json="$1"
    local field="$2"

    if [[ "$USE_JQ" == "true" ]]; then
        echo "$json" | jq -r "$field" 2>/dev/null
    else
        echo "$json" | grep -oP "\"${field}\":\s*\"\K[^\"]+|\"${field}\":\s*
\K[^,}]+" | head -1
    fi
}

# Function to securely wait for resource propagation with exponential backoff
wait_for_propagation() {
    local resource_type="$1"
    local wait_time="${2:-10}"

    # Validate wait_time is a positive integer
    if ! [[ "$wait_time" =~ ^[0-9]+$ ]] || [ "$wait_time" -lt 1 ] ||
 [ "$wait_time" -gt 300 ]; then
        echo "WARNING: Invalid wait time $wait_time, using default 10 seconds"
        wait_time=10
    fi

    echo "Waiting for $resource_type to propagate ($wait_time seconds)..."
    sleep "$wait_time"
}

# Function to validate JSON file efficiently
```

```
validate_json_file() {
    local file="$1"

    if [[ "$USE_JQ" == "true" ]]; then
        if ! jq empty "$file" 2>/dev/null; then
            handle_error "Invalid JSON in $file"
        fi
    fi
}

# Function to handle errors
handle_error() {
    echo "ERROR: $1"
    echo "Resources created:"
    if [ -n "${STATE_MACHINE_ARN:-}" ]; then
        echo "- State Machine: $STATE_MACHINE_ARN"
    fi
    if [ -n "${ROLE_NAME:-}" ]; then
        echo "- IAM Role: $ROLE_NAME"
    fi
    if [ -n "${POLICY_ARN:-}" ]; then
        echo "- IAM Policy: $POLICY_ARN"
    fi
    if [ -n "${STEPFUNCTIONS_POLICY_ARN:-}" ]; then
        echo "- Step Functions Policy: $STEPFUNCTIONS_POLICY_ARN"
    fi

    echo "Attempting to clean up resources..."
    cleanup
    exit 1
}

# Function to securely clean up resources with parallel deletion
cleanup() {
    echo "Cleaning up resources..."

    # Delete state machine if it exists
    if [ -n "${STATE_MACHINE_ARN:-}" ]; then
        echo "Deleting state machine: $STATE_MACHINE_ARN"
        aws stepfunctions delete-state-machine --state-machine-arn
 "$STATE_MACHINE_ARN" 2>/dev/null &
    fi

    # Detach and delete policies if they exist
```

```
    if [ -n "${POLICY_ARN:-}" ] && [ -n "${ROLE_NAME:-}" ]; then
        echo "Detaching Comprehend policy $POLICY_ARN from role $ROLE_NAME"
        aws iam detach-role-policy --role-name "$ROLE_NAME" --policy-arn
 "$POLICY_ARN" 2>/dev/null &
    fi

    if [ -n "${STEPFUNCTIONS_POLICY_ARN:-}" ] && [ -n "${ROLE_NAME:-}" ]; then
        echo "Detaching Step Functions policy $STEPFUNCTIONS_POLICY_ARN from role
 $ROLE_NAME"
        aws iam detach-role-policy --role-name "$ROLE_NAME" --policy-arn
 "$STEPFUNCTIONS_POLICY_ARN" 2>/dev/null &
    fi

    # Wait for detach operations to complete
    wait 2>/dev/null || true

    # Delete custom policies if they exist
    if [ -n "${POLICY_ARN:-}" ]; then
        echo "Deleting Comprehend policy: $POLICY_ARN"
        aws iam delete-policy --policy-arn "$POLICY_ARN" 2>/dev/null &
    fi

    if [ -n "${STEPFUNCTIONS_POLICY_ARN:-}" ]; then
        echo "Deleting Step Functions policy: $STEPFUNCTIONS_POLICY_ARN"
        aws iam delete-policy --policy-arn "$STEPFUNCTIONS_POLICY_ARN" 2>/dev/
null &
    fi

    # Wait for policy deletion to complete
    wait 2>/dev/null || true

    # Delete role if it exists
    if [ -n "${ROLE_NAME:-}" ]; then
        echo "Deleting role: $ROLE_NAME"
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || echo "Failed
 to delete role"
    fi

    # Remove temporary files securely
    echo "Removing temporary files"
    local temp_files=(
        "hello-world.json"
        "updated-hello-world.json"
        "sentiment-hello-world.json"
```

```
        "step-functions-trust-policy.json"
        "comprehend-policy.json"
        "stepfunctions-policy.json"
        "input.json"
        "sentiment-input.json"
    )

    for file in "${temp_files[@]}"; do
        if [ -f "$file" ]; then
            if command -v shred &> /dev/null; then
                shred -vfz -n 3 "$file" 2>/dev/null || rm -f "$file"
            else
                rm -f "$file"
            fi
        fi
    done
}

# Security: Set trap to cleanup on script exit
trap cleanup EXIT

# Generate a secure random identifier for resource names
RANDOM_ID=$(openssl rand -hex 4)
ROLE_NAME="StepFunctionsHelloWorldRole-${RANDOM_ID}"
POLICY_NAME="DetectSentimentPolicy-${RANDOM_ID}"
STATE_MACHINE_NAME="MyFirstStateMachine-${RANDOM_ID}"

echo "Using random identifier: $RANDOM_ID"
echo "Role name: $ROLE_NAME"
echo "Policy name: $POLICY_NAME"
echo "State machine name: $STATE_MACHINE_NAME"

# Step 1: Create the state machine definition
echo "Creating state machine definition..."
cat > hello-world.json << 'EOF'
{
  "Comment": "A Hello World example of the Amazon States Language using a Pass
 state",
  "StartAt": "SetVariables",
  "States": {
    "SetVariables": {
      "Type": "Pass",
      "Result": {
        "IsHelloWorldExample": true,
```

```
        "ExecutionWaitTimeInSeconds": 10
      },
      "Next": "IsHelloWorldExample"
    },
    "IsHelloWorldExample": {
      "Type": "Choice",
      "Choices": [
        {
          "Variable": "$.IsHelloWorldExample",
          "BooleanEquals": true,
          "Next": "WaitState"
        }
      ],
      "Default": "FailState"
    },
    "WaitState": {
      "Type": "Wait",
      "SecondsPath": "$.ExecutionWaitTimeInSeconds",
      "Next": "ParallelProcessing"
    },
    "ParallelProcessing": {
      "Type": "Parallel",
      "Branches": [
        {
          "StartAt": "Process1",
          "States": {
            "Process1": {
              "Type": "Pass",
              "Result": {
                "message": "Processing task 1"
              },
              "End": true
            }
          }
        },
        {
          "StartAt": "Process2",
          "States": {
            "Process2": {
              "Type": "Pass",
              "Result": {
                "message": "Processing task 2"
              },
              "End": true
```

```
            }
          }
        }
      ],
      "Next": "CheckpointState"
    },
    "CheckpointState": {
      "Type": "Pass",
      "Result": {
        "CheckpointMessage": "Workflow completed successfully!"
      },
      "Next": "SuccessState"
    },
    "SuccessState": {
      "Type": "Succeed"
    },
    "FailState": {
      "Type": "Fail",
      "Error": "NotHelloWorldExample",
      "Cause": "The IsHelloWorldExample value was false"
    }
  }
}
EOF

# Create IAM role trust policy
echo "Creating IAM role trust policy..."
cat > step-functions-trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "states.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create IAM role
echo "Creating IAM role: $ROLE_NAME"
```

```
ROLE_RESULT=$(aws iam create-role \
  --role-name "$ROLE_NAME" \
  --assume-role-policy-document file://step-functions-trust-policy.json 2>&1)

check_api_error "$ROLE_RESULT" "Create IAM role"
echo "Role created successfully"
aws iam tag-role --role-name "$ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-step-functions-gs

# Get the role ARN
ROLE_ARN=$(extract_json_field "$ROLE_RESULT" ".Role.Arn")

if [ -z "$ROLE_ARN" ]; then
    handle_error "Failed to extract role ARN"
fi
echo "Role ARN: $ROLE_ARN"

# Create a custom policy for Step Functions with least privilege
echo "Creating custom policy for Step Functions..."
cat > stepfunctions-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "states:StartExecution",
        "states:DescribeExecution",
        "states:StopExecution"
      ],
      "Resource": "arn:aws:states:*:*:stateMachine:*"
    }
  ]
}
EOF

# Create the policy
echo "Creating Step Functions policy..."
STEPFUNCTIONS_POLICY_RESULT=$(aws iam create-policy \
  --policy-name "StepFunctionsPolicy-${RANDOM_ID}" \
  --policy-document file://stepfunctions-policy.json 2>&1)

check_api_error "$STEPFUNCTIONS_POLICY_RESULT" "Create Step Functions policy"
echo "Step Functions policy created successfully"
```

```
# Get the policy ARN
STEPFUNCTIONS_POLICY_ARN=$(extract_json_field "$STEPFUNCTIONS_POLICY_RESULT"
 ".Policy.Arn")

if [ -z "$STEPFUNCTIONS_POLICY_ARN" ]; then
    handle_error "Failed to extract Step Functions policy ARN"
fi
echo "Step Functions policy ARN: $STEPFUNCTIONS_POLICY_ARN"

# Attach policy to the role
echo "Attaching Step Functions policy to role..."
ATTACH_RESULT=$(aws iam attach-role-policy \
  --role-name "$ROLE_NAME" \
  --policy-arn "$STEPFUNCTIONS_POLICY_ARN" 2>&1)

if [ $? -ne 0 ]; then
    handle_error "Failed to attach Step Functions policy to role: $ATTACH_RESULT"
fi

# Wait for role to propagate (IAM changes can take time to propagate)
wait_for_propagation "IAM role" 8

# Create state machine
echo "Creating state machine: $STATE_MACHINE_NAME"
SM_RESULT=$(aws stepfunctions create-state-machine \
  --name "$STATE_MACHINE_NAME" \
  --definition file://hello-world.json \
  --role-arn "$ROLE_ARN" \
  --type STANDARD \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=aws-step-functions-gs
 2>&1)

check_api_error "$SM_RESULT" "Create state machine"
echo "State machine created successfully"

# Get the state machine ARN
STATE_MACHINE_ARN=$(extract_json_field "$SM_RESULT" ".stateMachineArn")

if [ -z "$STATE_MACHINE_ARN" ]; then
    handle_error "Failed to extract state machine ARN"
fi
echo "State machine ARN: $STATE_MACHINE_ARN"
```

```
# Step 2: Start the state machine execution
echo "Starting state machine execution..."
EXEC_RESULT=$(aws stepfunctions start-execution \
  --state-machine-arn "$STATE_MACHINE_ARN" \
  --name "hello001-${RANDOM_ID}" 2>&1)

check_api_error "$EXEC_RESULT" "Start execution"
echo "Execution started successfully"

# Get the execution ARN
EXECUTION_ARN=$(extract_json_field "$EXEC_RESULT" ".executionArn")

if [ -z "$EXECUTION_ARN" ]; then
    handle_error "Failed to extract execution ARN"
fi
echo "Execution ARN: $EXECUTION_ARN"

# Wait for execution to complete (the workflow has a 10-second wait state)
echo "Waiting for execution to complete (12 seconds)..."
sleep 12

# Check execution status
echo "Checking execution status..."
EXEC_STATUS=$(aws stepfunctions describe-execution \
  --execution-arn "$EXECUTION_ARN" 2>&1)

echo "Execution status: $EXEC_STATUS"

# Step 3: Update state machine to process external input
echo "Updating state machine to process external input..."
cat > updated-hello-world.json << 'EOF'
{
  "Comment": "A Hello World example of the Amazon States Language using a Pass
 state",
  "StartAt": "SetVariables",
  "States": {
    "SetVariables": {
      "Type": "Pass",
      "Parameters": {
        "IsHelloWorldExample.$": "$.hello_world",
        "ExecutionWaitTimeInSeconds.$": "$.wait"
      },
      "Next": "IsHelloWorldExample"
    },
```

```
    "IsHelloWorldExample": {
      "Type": "Choice",
      "Choices": [
        {
          "Variable": "$.IsHelloWorldExample",
          "BooleanEquals": true,
          "Next": "WaitState"
        }
      ],
      "Default": "FailState"
    },
    "WaitState": {
      "Type": "Wait",
      "SecondsPath": "$.ExecutionWaitTimeInSeconds",
      "Next": "ParallelProcessing"
    },
    "ParallelProcessing": {
      "Type": "Parallel",
      "Branches": [
        {
          "StartAt": "Process1",
          "States": {
            "Process1": {
              "Type": "Pass",
              "Result": {
                "message": "Processing task 1"
              },
              "End": true
            }
          }
        },
        {
          "StartAt": "Process2",
          "States": {
            "Process2": {
              "Type": "Pass",
              "Result": {
                "message": "Processing task 2"
              },
              "End": true
            }
          }
        }
      ],
```

```
      "Next": "CheckpointState"
    },
    "CheckpointState": {
      "Type": "Pass",
      "Result": {
        "CheckpointMessage": "Workflow completed successfully!"
      },
      "Next": "SuccessState"
    },
    "SuccessState": {
      "Type": "Succeed"
    },
    "FailState": {
      "Type": "Fail",
      "Error": "NotHelloWorldExample",
      "Cause": "The IsHelloWorldExample value was false"
    }
  }
}
EOF

# Update state machine
echo "Updating state machine..."
UPDATE_RESULT=$(aws stepfunctions update-state-machine \
  --state-machine-arn "$STATE_MACHINE_ARN" \
  --definition file://updated-hello-world.json \
  --role-arn "$ROLE_ARN" 2>&1)

check_api_error "$UPDATE_RESULT" "Update state machine"
echo "State machine updated successfully"

# Create input file with strict validation
echo "Creating input file..."
cat > input.json << 'EOF'
{
  "wait": 5,
  "hello_world": true
}
EOF

# Validate input JSON
validate_json_file "input.json"

# Start execution with input
```

```
echo "Starting execution with input..."
EXEC2_RESULT=$(aws stepfunctions start-execution \
  --state-machine-arn "$STATE_MACHINE_ARN" \
  --name "hello002-${RANDOM_ID}" \
  --input file://input.json 2>&1)

check_api_error "$EXEC2_RESULT" "Start execution with input"
echo "Execution with input started successfully"

# Get the execution ARN
EXECUTION2_ARN=$(extract_json_field "$EXEC2_RESULT" ".executionArn")

if [ -z "$EXECUTION2_ARN" ]; then
    handle_error "Failed to extract execution ARN"
fi
echo "Execution ARN: $EXECUTION2_ARN"

# Wait for execution to complete (the workflow has a 5-second wait state)
echo "Waiting for execution to complete (8 seconds)..."
sleep 8

# Check execution status
echo "Checking execution status..."
EXEC2_STATUS=$(aws stepfunctions describe-execution \
  --execution-arn "$EXECUTION2_ARN" 2>&1)

echo "Execution status: $EXEC2_STATUS"

# Step 4: Integrate Amazon Comprehend for sentiment analysis (if available)
if [[ "$SKIP_COMPREHEND" == "false" ]]; then
    echo "Creating policy for Amazon Comprehend access with least privilege..."
    cat > comprehend-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "comprehend:DetectSentiment"
      ],
      "Resource": "*"
    }
  ]
}
```

```
EOF

    # Create policy
    echo "Creating IAM policy: $POLICY_NAME"
    POLICY_RESULT=$(aws iam create-policy \
      --policy-name "$POLICY_NAME" \
      --policy-document file://comprehend-policy.json 2>&1)

    check_api_error "$POLICY_RESULT" "Create Comprehend policy"
    echo "Comprehend policy created successfully"

    # Get policy ARN
    POLICY_ARN=$(extract_json_field "$POLICY_RESULT" ".Policy.Arn")

    if [ -z "$POLICY_ARN" ]; then
        handle_error "Failed to extract policy ARN"
    fi
    echo "Policy ARN: $POLICY_ARN"

    # Attach policy to role
    echo "Attaching policy to role..."
    ATTACH2_RESULT=$(aws iam attach-role-policy \
      --role-name "$ROLE_NAME" \
      --policy-arn "$POLICY_ARN" 2>&1)

    if [ $? -ne 0 ]; then
        handle_error "Failed to attach policy to role: $ATTACH2_RESULT"
    fi

    # Create updated state machine definition with sentiment analysis
    echo "Creating updated state machine definition with sentiment analysis..."
    cat > sentiment-hello-world.json << 'EOF'
{
  "Comment": "A Hello World example with sentiment analysis",
  "StartAt": "SetVariables",
  "States": {
    "SetVariables": {
      "Type": "Pass",
      "Parameters": {
        "IsHelloWorldExample.$": "$.hello_world",
        "ExecutionWaitTimeInSeconds.$": "$.wait",
        "FeedbackComment.$": "$.feedback_comment"
      },
      "Next": "IsHelloWorldExample"
```

```
    },
    "IsHelloWorldExample": {
      "Type": "Choice",
      "Choices": [
        {
          "Variable": "$.IsHelloWorldExample",
          "BooleanEquals": true,
          "Next": "WaitState"
        }
      ],
      "Default": "DetectSentiment"
    },
    "WaitState": {
      "Type": "Wait",
      "SecondsPath": "$.ExecutionWaitTimeInSeconds",
      "Next": "ParallelProcessing"
    },
    "ParallelProcessing": {
      "Type": "Parallel",
      "Branches": [
        {
          "StartAt": "Process1",
          "States": {
            "Process1": {
              "Type": "Pass",
              "Result": {
                "message": "Processing task 1"
              },
              "End": true
            }
          }
        },
        {
          "StartAt": "Process2",
          "States": {
            "Process2": {
              "Type": "Pass",
              "Result": {
                "message": "Processing task 2"
              },
              "End": true
            }
          }
        }
```

```
      ],
      "Next": "CheckpointState"
    },
    "CheckpointState": {
      "Type": "Pass",
      "Result": {
        "CheckpointMessage": "Workflow completed successfully!"
      },
      "Next": "SuccessState"
    },
    "DetectSentiment": {
      "Type": "Task",
      "Resource": "arn:aws:states:::aws-sdk:comprehend:detectSentiment",
      "Parameters": {
        "LanguageCode": "en",
        "Text.$": "$.FeedbackComment"
      },
      "Next": "SuccessState"
    },
    "SuccessState": {
      "Type": "Succeed"
    }
  }
}
EOF

    # Validate sentiment state machine JSON
    validate_json_file "sentiment-hello-world.json"

    # Wait for IAM changes to propagate
    wait_for_propagation "IAM changes" 8

    # Update state machine
    echo "Updating state machine with sentiment analysis..."
    UPDATE2_RESULT=$(aws stepfunctions update-state-machine \
      --state-machine-arn "$STATE_MACHINE_ARN" \
      --definition file://sentiment-hello-world.json \
      --role-arn "$ROLE_ARN" 2>&1)

    check_api_error "$UPDATE2_RESULT" "Update state machine with sentiment
 analysis"
    echo "State machine updated with sentiment analysis successfully"

    # Create input file with feedback comment
```

```
    echo "Creating input file with feedback comment..."
    cat > sentiment-input.json << 'EOF'
{
  "hello_world": false,
  "wait": 5,
  "feedback_comment": "This getting started with Step Functions workshop is a
 challenge!"
}
EOF

    # Validate sentiment input JSON
    validate_json_file "sentiment-input.json"

    # Start execution with sentiment analysis input
    echo "Starting execution with sentiment analysis input..."
    EXEC3_RESULT=$(aws stepfunctions start-execution \
      --state-machine-arn "$STATE_MACHINE_ARN" \
      --name "hello003-${RANDOM_ID}" \
      --input file://sentiment-input.json 2>&1)

    check_api_error "$EXEC3_RESULT" "Start execution with sentiment analysis"
    echo "Execution with sentiment analysis started successfully"

    # Get the execution ARN
    EXECUTION3_ARN=$(extract_json_field "$EXEC3_RESULT" ".executionArn")

    if [ -z "$EXECUTION3_ARN" ]; then
        handle_error "Failed to extract execution ARN"
    fi
    echo "Execution ARN: $EXECUTION3_ARN"

    # Wait for execution to complete
    echo "Waiting for execution to complete (3 seconds)..."
    sleep 3

    # Check execution status
    echo "Checking execution status..."
    EXEC3_STATUS=$(aws stepfunctions describe-execution \
      --execution-arn "$EXECUTION3_ARN" 2>&1)

    echo "Execution status: $EXEC3_STATUS"
else
    echo "Skipping Amazon Comprehend integration (not available in
 $CURRENT_REGION)"
```

```
    EXECUTION3_ARN=""
fi

# Display summary of resources created
echo ""
echo "========================================="
echo "RESOURCES CREATED"
echo "========================================="
echo "State Machine: $STATE_MACHINE_ARN"
echo "IAM Role: $ROLE_NAME"
echo "Step Functions Policy: StepFunctionsPolicy-${RANDOM_ID}
 ($STEPFUNCTIONS_POLICY_ARN)"
if [[ "$SKIP_COMPREHEND" == "false" ]]; then
    echo "Comprehend Policy: $POLICY_NAME ($POLICY_ARN)"
fi
echo "Executions:"
echo "  - hello001-${RANDOM_ID}: $EXECUTION_ARN"
echo "  - hello002-${RANDOM_ID}: $EXECUTION2_ARN"
if [[ "$SKIP_COMPREHEND" == "false" ]]; then
    echo "  - hello003-${RANDOM_ID}: $EXECUTION3_ARN"
fi
echo "========================================="

# Cleanup
echo ""
echo "========================================="
echo "CLEANUP"
echo "========================================="
echo "Auto-cleanup enabled. Cleaning up resources..."

echo "All resources have been cleaned up."

echo "Script completed successfully!"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreatePolicy
  - CreateRole
  - CreateStateMachine
  - DeletePolicy

- DeleteRole
- DeleteStateMachine
- DescribeExecution
- DetachRolePolicy
- StartExecution
- UpdateStateMachine

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci le chiavi di accesso IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come gestire le chiavi di accesso.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come Centro identità AWS IAM.

- Creare ed elencare le chiavi di accesso.
- Scoprire come e quando una chiave di accesso è stata utilizzata per ultima.
- Aggiornare ed eliminare le chiavi di accesso.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni delle chiavi di accesso IAM.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

iam = boto3.resource("iam")

def list_keys(user_name):
    """
    Lists the keys owned by the specified user.

    :param user_name: The name of the user.
    :return: The list of keys owned by the user.
    """
    try:
        keys = list(iam.User(user_name).access_keys.all())
        logger.info("Got %s access keys for %s.", len(keys), user_name)
    except ClientError:
        logger.exception("Couldn't get access keys for %s.", user_name)
        raise
    else:
        return keys



def create_key(user_name):
    """
    Creates an access key for the specified user. Each user can have a
    maximum of two keys.

    :param user_name: The name of the user.
    :return: The created access key.
    """
    try:
        key_pair = iam.User(user_name).create_access_key_pair()
        logger.info(
            "Created access key pair for %s. Key ID is %s.",
            key_pair.user_name,
            key_pair.id,
        )
    except ClientError:
        logger.exception("Couldn't create access key pair for %s.", user_name)
```

```
        raise
    else:
        return key_pair



def get_last_use(key_id):
    """
    Gets information about when and how a key was last used.

    :param key_id: The ID of the key to look up.
    :return: Information about the key's last use.
    """
    try:
        response = iam.meta.client.get_access_key_last_used(AccessKeyId=key_id)
        last_used_date = response["AccessKeyLastUsed"].get("LastUsedDate", None)
        last_service = response["AccessKeyLastUsed"].get("ServiceName", None)
        logger.info(
            "Key %s was last used by %s on %s to access %s.",
            key_id,
            response["UserName"],
            last_used_date,
            last_service,
        )
    except ClientError:
        logger.exception("Couldn't get last use of key %s.", key_id)
        raise
    else:
        return response



def update_key(user_name, key_id, activate):
    """
    Updates the status of a key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to update.
    :param activate: When True, the key is activated. Otherwise, the key is
 deactivated.
    """

    try:
        key = iam.User(user_name).AccessKey(key_id)
```

```
        if activate:
            key.activate()
        else:
            key.deactivate()
        logger.info("%s key %s.", "Activated" if activate else "Deactivated",
 key_id)
    except ClientError:
        logger.exception(
            "Couldn't %s key %s.", "Activate" if activate else "Deactivate",
 key_id
        )
        raise



def delete_key(user_name, key_id):
    """
    Deletes a user's access key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to delete.
    """

    try:
        key = iam.AccessKey(user_name, key_id)
        key.delete()
        logger.info("Deleted access key %s for %s.", key.id, key.user_name)
    except ClientError:
        logger.exception("Couldn't delete key %s for %s", key_id, user_name)
        raise
```

Utilizza le funzioni di wrapping per eseguire operazioni sulle chiavi di accesso per l'utente corrente.

```
def usage_demo():
    """Shows how to create and manage access keys."""

    def print_keys():
        """Gets and prints the current keys for a user."""
        current_keys = list_keys(current_user_name)
```

```
        print("The current user's keys are now:")
        print(*[f"{key.id}: {key.status}" for key in current_keys], sep="\n")

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management access key demo.")
    print("-" * 88)
    current_user_name = iam.CurrentUser().user_name
    print(
        f"This demo creates an access key for the current user "
        f"({current_user_name}), manipulates the key in a few ways, and then "
        f"deletes it."
    )
    all_keys = list_keys(current_user_name)
    if len(all_keys) == 2:
        print(
            "The current user already has the maximum of 2 access keys. To run "
            "this demo, either delete one of the access keys or use a user "
            "that has only 1 access key."
        )
    else:
        new_key = create_key(current_user_name)
        print(f"Created a new key with id {new_key.id} and secret
{new_key.secret}.")
        print_keys()
        existing_key = next(key for key in all_keys if key != new_key)
        last_use = get_last_use(existing_key.id)["AccessKeyLastUsed"]
        print(
            f"Key {all_keys[0].id} was last used to access
{last_use['ServiceName']} "
            f"on {last_use['LastUsedDate']}"
        )
        update_key(current_user_name, new_key.id, False)
        print(f"Key {new_key.id} is now deactivated.")
        print_keys()
        delete_key(current_user_name, new_key.id)
        print_keys()
        print("Thanks for watching!")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - CreateAccessKey
  - DeleteAccessKey
  - GetAccessKeyLastUsed
  - ListAccessKeys
  - UpdateAccessKey

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci le policy IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Creare ed elencare le policy.
- Creare ed ottenere le versioni della policy.
- Ripristinare una policy a una versione precedente.
- Eliminare le policy.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni delle policy IAM.

```
import json
import logging
import operator
import pprint
```

```
import time

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def create_policy(name, description, actions, resource_arn):
    """
    Creates a policy that contains a single statement.

    :param name: The name of the policy to create.
    :param description: The description of the policy.
    :param actions: The actions allowed by the policy. These typically take the
                    form of service:action, such as s3:PutObject.
    :param resource_arn: The Amazon Resource Name (ARN) of the resource this
 policy
                         applies to. This ARN can contain wildcards, such as
                         'arn:aws:s3:::amzn-s3-demo-bucket/*' to allow actions on
 all objects
                         in the bucket named 'amzn-s3-demo-bucket'.
    :return: The newly created policy.
    """
    policy_doc = {
        "Version":"2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
 resource_arn}],
    }
    try:
        policy = iam.create_policy(
            PolicyName=name,
            Description=description,
            PolicyDocument=json.dumps(policy_doc),
        )
        logger.info("Created policy %s.", policy.arn)
    except ClientError:
        logger.exception("Couldn't create policy %s.", name)
        raise
    else:
        return policy
```

```
def list_policies(scope):
    """
    Lists the policies in the current account.

    :param scope: Limits the kinds of policies that are returned. For example,
                  'Local' specifies that only locally managed policies are
 returned.
    :return: The list of policies.
    """
    try:
        policies = list(iam.policies.filter(Scope=scope))
        logger.info("Got %s policies in scope '%s'.", len(policies), scope)
    except ClientError:
        logger.exception("Couldn't get policies for scope '%s'.", scope)
        raise
    else:
        return policies



def create_policy_version(policy_arn, actions, resource_arn, set_as_default):
    """
    Creates a policy version. Policies can have up to five versions. The default
    version is the one that is used for all resources that reference the policy.

    :param policy_arn: The ARN of the policy.
    :param actions: The actions to allow in the policy version.
    :param resource_arn: The ARN of the resource this policy version applies to.
    :param set_as_default: When True, this policy version is set as the default
                           version for the policy. Otherwise, the default
                           is not changed.
    :return: The newly created policy version.
    """
    policy_doc = {
        "Version":"2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
 resource_arn}],
    }
    try:
        policy = iam.Policy(policy_arn)
        policy_version = policy.create_version(
            PolicyDocument=json.dumps(policy_doc), SetAsDefault=set_as_default
        )
        logger.info(
```

```
            "Created policy version %s for policy %s.",
            policy_version.version_id,
            policy_version.arn,
        )
    except ClientError:
        logger.exception("Couldn't create a policy version for %s.", policy_arn)
        raise
    else:
        return policy_version



def get_default_policy_statement(policy_arn):
    """
    Gets the statement of the default version of the specified policy.

    :param policy_arn: The ARN of the policy to look up.
    :return: The statement of the default policy version.
    """
    try:
        policy = iam.Policy(policy_arn)
        # To get an attribute of a policy, the SDK first calls get_policy.
        policy_doc = policy.default_version.document
        policy_statement = policy_doc.get("Statement", None)
        logger.info("Got default policy doc for %s.", policy.policy_name)
        logger.info(policy_doc)
    except ClientError:
        logger.exception("Couldn't get default policy statement for %s.",
 policy_arn)
        raise
    else:
        return policy_statement



def rollback_policy_version(policy_arn):
    """
    Rolls back to the previous default policy, if it exists.

    1. Gets the list of policy versions in order by date.
    2. Finds the default.
    3. Makes the previous policy the default.
    4. Deletes the old default version.
```

```
    :param policy_arn: The ARN of the policy to roll back.
    :return: The default version of the policy after the rollback.
    """
    try:
        policy_versions = sorted(
            iam.Policy(policy_arn).versions.all(),
            key=operator.attrgetter("create_date"),
        )
        logger.info("Got %s versions for %s.", len(policy_versions), policy_arn)
    except ClientError:
        logger.exception("Couldn't get versions for %s.", policy_arn)
        raise

    default_version = None
    rollback_version = None
    try:
        while default_version is None:
            ver = policy_versions.pop()
            if ver.is_default_version:
                default_version = ver
        rollback_version = policy_versions.pop()
        rollback_version.set_as_default()
        logger.info("Set %s as the default version.",
rollback_version.version_id)
        default_version.delete()
        logger.info("Deleted original default version %s.",
default_version.version_id)
    except IndexError:
        if default_version is None:
            logger.warning("No default version found for %s.", policy_arn)
        elif rollback_version is None:
            logger.warning(
                "Default version %s found for %s, but no previous version exists,
so "
                "nothing to roll back to.",
                default_version.version_id,
                policy_arn,
            )
    except ClientError:
        logger.exception("Couldn't roll back version for %s.", policy_arn)
        raise
    else:
        return rollback_version
```

```
def delete_policy(policy_arn):
    """
    Deletes a policy.

    :param policy_arn: The ARN of the policy to delete.
    """
    try:
        iam.Policy(policy_arn).delete()
        logger.info("Deleted policy %s.", policy_arn)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_arn)
        raise
```

Utilizza le funzioni di wrapping per creare policy, aggiornare le versioni e ottenere informazioni su di esse.

```
def usage_demo():
    """Shows how to use the policy functions."""
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management policy demo.")
    print("-" * 88)
    print(
        "Policies let you define sets of permissions that can be attached to "
        "other IAM resources, like users and roles."
    )
    bucket_arn = f"arn:aws:s3:::amzn-s3-demo-bucket"
    policy = create_policy(
        "demo-iam-policy",
        "Policy for IAM demonstration.",
        ["s3:ListObjects"],
        bucket_arn,
    )
    print(f"Created policy {policy.policy_name}.")
    policies = list_policies("Local")
    print(f"Your account has {len(policies)} managed policies:")
    print(*[pol.policy_name for pol in policies], sep=", ")
    time.sleep(1)
```

```
    policy_version = create_policy_version(
        policy.arn, ["s3:PutObject"], bucket_arn, True
    )
    print(
        f"Added policy version {policy_version.version_id} to policy "
        f"{policy.policy_name}."
    )
    default_statement = get_default_policy_statement(policy.arn)
    print(f"The default policy statement for {policy.policy_name} is:")
    pprint.pprint(default_statement)
    rollback_version = rollback_policy_version(policy.arn)
    print(
        f"Rolled back to version {rollback_version.version_id} for "
        f"{policy.policy_name}."
    )
    default_statement = get_default_policy_statement(policy.arn)
    print(f"The default policy statement for {policy.policy_name} is now:")
    pprint.pprint(default_statement)
    delete_policy(policy.arn)
    print(f"Deleted policy {policy.policy_name}.")
    print("Thanks for watching!")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - CreatePolicy
  - CreatePolicyVersion
  - DeletePolicy
  - DeletePolicyVersion
  - GetPolicyVersion
  - ListPolicies
  - ListPolicyVersions
  - SetDefaultPolicyVersion

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci i ruoli IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Crea un ruolo IAM.
- Collegamento e scollegamento delle policy per un ruolo
- Elimina un ruolo.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni dei ruoli IAM.

```
import json
import logging
import pprint

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def create_role(role_name, allowed_services):
    """
    Creates a role that lets a list of specified services assume the role.

    :param role_name: The name of the role.
    :param allowed_services: The services that can assume the role.
    :return: The newly created role.
    """
    trust_policy = {
        "Version":"2012-10-17",
        "Statement": [
```

```
            {
                "Effect": "Allow",
                "Principal": {"Service": service},
                "Action": "sts:AssumeRole",
            }
            for service in allowed_services
        ],
    }

    try:
        role = iam.create_role(
            RoleName=role_name, AssumeRolePolicyDocument=json.dumps(trust_policy)
        )
        logger.info("Created role %s.", role.name)
    except ClientError:
        logger.exception("Couldn't create role %s.", role_name)
        raise
    else:
        return role



def attach_policy(role_name, policy_arn):
    """
    Attaches a policy to a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Role(role_name).attach_policy(PolicyArn=policy_arn)
        logger.info("Attached policy %s to role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to role %s.", policy_arn,
 role_name)
        raise



def detach_policy(role_name, policy_arn):
    """
    Detaches a policy from a role.
```

```
    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Role(role_name).detach_policy(PolicyArn=policy_arn)
        logger.info("Detached policy %s from role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from role %s.", policy_arn, role_name
        )
        raise



def delete_role(role_name):
    """
    Deletes a role.

    :param role_name: The name of the role to delete.
    """
    try:
        iam.Role(role_name).delete()
        logger.info("Deleted role %s.", role_name)
    except ClientError:
        logger.exception("Couldn't delete role %s.", role_name)
        raise
```

Utilizza le funzioni di wrapping per creare un ruolo, per poi collegare e scollegare una policy.

```
def usage_demo():
    """Shows how to use the role functions."""
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management role demo.")
    print("-" * 88)
    print(
        "Roles let you define sets of permissions and can be assumed by "
        "other entities, like users and services."
    )
```

```
    print("The first 10 roles currently in your account are:")
    roles = list_roles(10)
    print(f"The inline policies for role {roles[0].name} are:")
    list_policies(roles[0].name)
    role = create_role(
        "demo-iam-role", ["lambda.amazonaws.com",
"batchoperations.s3.amazonaws.com"]
    )
    print(f"Created role {role.name}, with trust policy:")
    pprint.pprint(role.assume_role_policy_document)
    policy_arn = "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess"
    attach_policy(role.name, policy_arn)
    print(f"Attached policy {policy_arn} to {role.name}.")
    print(f"Policies attached to role {role.name} are:")
    list_attached_policies(role.name)
    detach_policy(role.name, policy_arn)
    print(f"Detached policy {policy_arn} from {role.name}.")
    delete_role(role.name)
    print(f"Deleted {role.name}.")
    print("Thanks for watching!")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - AttachRolePolicy
  - CreateRole
  - DeleteRole
  - DetachRolePolicy

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci il tuo account IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Ottenere e aggiornare l'alias dell'account.

- Generare un report degli utenti e delle loro credenziali.
- Ottenere un riepilogo dell'utilizzo dell'account.
- Ottenere informazioni su tutti gli utenti, gruppi, ruoli e policy nell'account, comprese le relazioni reciproche.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni dell'account IAM.

```
import logging
import pprint
import sys
import time
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def list_aliases():
    """
    Gets the list of aliases for the current account. An account has at most one
 alias.

    :return: The list of aliases for the account.
    """
    try:
        response = iam.meta.client.list_account_aliases()
        aliases = response["AccountAliases"]
        if len(aliases) > 0:
            logger.info("Got aliases for your account: %s.", ",".join(aliases))
        else:
            logger.info("Got no aliases for your account.")
```

```
    except ClientError:
        logger.exception("Couldn't list aliases for your account.")
        raise
    else:
        return response["AccountAliases"]



def create_alias(alias):
    """
    Creates an alias for the current account. The alias can be used in place of
 the
    account ID in the sign-in URL. An account can have only one alias. When a new
    alias is created, it replaces any existing alias.

    :param alias: The alias to assign to the account.
    """

    try:
        iam.create_account_alias(AccountAlias=alias)
        logger.info("Created an alias '%s' for your account.", alias)
    except ClientError:
        logger.exception("Couldn't create alias '%s' for your account.", alias)
        raise



def delete_alias(alias):
    """
    Removes the alias from the current account.

    :param alias: The alias to remove.
    """
    try:
        iam.meta.client.delete_account_alias(AccountAlias=alias)
        logger.info("Removed alias '%s' from your account.", alias)
    except ClientError:
        logger.exception("Couldn't remove alias '%s' from your account.", alias)
        raise



def generate_credential_report():
    """
```

```
    Starts generation of a credentials report about the current account. After
    calling this function to generate the report, call get_credential_report
    to get the latest report. A new report can be generated a minimum of four
 hours
    after the last one was generated.
    """
    try:
        response = iam.meta.client.generate_credential_report()
        logger.info(
            "Generating credentials report for your account. " "Current state is
 %s.",
            response["State"],
        )
    except ClientError:
        logger.exception("Couldn't generate a credentials report for your
 account.")
        raise
    else:
        return response



def get_credential_report():
    """
    Gets the most recently generated credentials report about the current
 account.

    :return: The credentials report.
    """
    try:
        response = iam.meta.client.get_credential_report()
        logger.debug(response["Content"])
    except ClientError:
        logger.exception("Couldn't get credentials report.")
        raise
    else:
        return response["Content"]



def get_summary():
    """
    Gets a summary of account usage.
```

```
    :return: The summary of account usage.
    """
    try:
        summary = iam.AccountSummary()
        logger.debug(summary.summary_map)
    except ClientError:
        logger.exception("Couldn't get a summary for your account.")
        raise
    else:
        return summary.summary_map



def get_authorization_details(response_filter):
    """
    Gets an authorization detail report for the current account.

    :param response_filter: A list of resource types to include in the report,
 such
                            as users or roles. When not specified, all resources
                            are included.
    :return: The authorization detail report.
    """
    try:
        account_details = iam.meta.client.get_account_authorization_details(
            Filter=response_filter
        )
        logger.debug(account_details)
    except ClientError:
        logger.exception("Couldn't get details for your account.")
        raise
    else:
        return account_details
```

Utilizza le funzioni di wrapping per modificare l'alias dell'account e recuperare report sull'account.

```
def usage_demo():
    """Shows how to use the account functions."""
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
```

```
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management account demo.")
    print("-" * 88)
    print(
        "Setting an account alias lets you use the alias in your sign-in URL "
        "instead of your account number."
    )
    old_aliases = list_aliases()
    if len(old_aliases) > 0:
        print(f"Your account currently uses '{old_aliases[0]}' as its alias.")
    else:
        print("Your account currently has no alias.")
    for index in range(1, 3):
        new_alias = f"alias-{index}-{time.time_ns()}"
        print(f"Setting your account alias to {new_alias}")
        create_alias(new_alias)
    current_aliases = list_aliases()
    print(f"Your account alias is now {current_aliases}.")
    delete_alias(current_aliases[0])
    print(f"Your account now has no alias.")
    if len(old_aliases) > 0:
        print(f"Restoring your original alias back to {old_aliases[0]}...")
        create_alias(old_aliases[0])

    print("-" * 88)
    print("You can get various reports about your account.")
    print("Let's generate a credentials report...")
    report_state = None
    while report_state != "COMPLETE":
        cred_report_response = generate_credential_report()
        old_report_state = report_state
        report_state = cred_report_response["State"]
        if report_state != old_report_state:
            print(report_state, sep="")
        else:
            print(".", sep="")
        sys.stdout.flush()
        time.sleep(1)
    print()
    cred_report = get_credential_report()
    col_count = 3
    print(f"Got credentials report. Showing only the first {col_count} columns.")
    cred_lines = [
```

```
        line.split(",")[:col_count] for line in
 cred_report.decode("utf-8").split("\n")
    ]
    col_width = max([len(item) for line in cred_lines for item in line]) + 2
    for line in cred_report.decode("utf-8").split("\n"):
        print(
            "".join(element.ljust(col_width) for element in line.split(",")
[:col_count])
        )

    print("-" * 88)
    print("Let's get an account summary.")
    summary = get_summary()
    print("Here's your summary:")
    pprint.pprint(summary)

    print("-" * 88)
    print("Let's get authorization details!")
    details = get_authorization_details([])
    see_details = input("These are pretty long, do you want to see them (y/n)? ")
    if see_details.lower() == "y":
        pprint.pprint(details)

    print("-" * 88)
    pw_policy_created = None
    see_pw_policy = input("Want to see the password policy for the account (y/n)?
 ")
    if see_pw_policy.lower() == "y":
        while True:
            if print_password_policy():
                break
            else:
                answer = input(
                    "Do you want to create a default password policy (y/n)? "
                )
                if answer.lower() == "y":
                    pw_policy_created = iam.create_account_password_policy()
                else:
                    break
    if pw_policy_created is not None:
        answer = input("Do you want to delete the password policy (y/n)? ")
        if answer.lower() == "y":
            pw_policy_created.delete()
            print("Password policy deleted.")
```

```
    print("The SAML providers for your account are:")
    list_saml_providers(10)

    print("-" * 88)
    print("Thanks for watching.")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - CreateAccountAlias
  - DeleteAccountAlias
  - GenerateCredentialReport
  - GetAccountAuthorizationDetails
  - GetAccountSummary
  - GetCredentialReport
  - ListAccountAliases

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Spostamento di segreti codificati per proteggere l'archiviazione segreta

L'esempio di codice seguente mostra come:

- Creazione di ruoli IAM
- Creazione di un segreto in Secrets Manager
- Aggiorna il codice dell'applicazione
- Aggiorna il segreto
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Script to move hardcoded secrets to AWS Secrets Manager
# This script demonstrates how to create IAM roles, store a secret in AWS Secrets
 Manager,
# and set up appropriate permissions

set -euo pipefail

# Set up logging
LOG_FILE="secrets_manager_tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting AWS Secrets Manager tutorial script at $(date)"
echo "============================================================"

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2

    if echo "$output" | grep -qi "error"; then
        echo "ERROR: Command failed: $cmd"
        echo "$output"
        cleanup_resources
        exit 1
    fi
}

# Function to generate a random identifier using secure method
generate_random_id() {
```

```
    python3 -c "import secrets; print('sm' + secrets.token_hex(4))"
}

# Function to safely clean up resources
cleanup_resources() {
    echo ""
    echo "========================================="
    echo "RESOURCES CREATED"
    echo "========================================="

    if [ -n "${SECRET_NAME:-}" ]; then
        echo "Secret: $SECRET_NAME"
    fi

    if [ -n "${RUNTIME_ROLE_NAME:-}" ]; then
        echo "IAM Role: $RUNTIME_ROLE_NAME"
    fi

    if [ -n "${ADMIN_ROLE_NAME:-}" ]; then
        echo "IAM Role: $ADMIN_ROLE_NAME"
    fi

    echo ""
    echo "========================================="
    echo "CLEANUP CONFIRMATION"
    echo "========================================="
    echo "Cleaning up all created resources..."

    # Delete secret if it exists
    if [ -n "${SECRET_NAME:-}" ]; then
        echo "Deleting secret: $SECRET_NAME"
        aws secretsmanager delete-secret --secret-id "$SECRET_NAME" --force-
delete-without-recovery 2>/dev/null || true
    fi

    # Detach policies and delete runtime role if it exists
    if [ -n "${RUNTIME_ROLE_NAME:-}" ]; then
        echo "Deleting inline policies from runtime role: $RUNTIME_ROLE_NAME"
        for policy in $(aws iam list-role-policies --role-name
 "$RUNTIME_ROLE_NAME" --query 'PolicyNames[]' --output text 2>/dev/null || true);
 do
            aws iam delete-role-policy --role-name "$RUNTIME_ROLE_NAME" --policy-
name "$policy" 2>/dev/null || true
        done
```

```
        echo "Deleting IAM role: $RUNTIME_ROLE_NAME"
        aws iam delete-role --role-name "$RUNTIME_ROLE_NAME" 2>/dev/null || true
    fi

    # Detach policies and delete admin role if it exists
    if [ -n "${ADMIN_ROLE_NAME:-}" ]; then
        echo "Detaching policy from role: $ADMIN_ROLE_NAME"
        aws iam detach-role-policy --role-name "$ADMIN_ROLE_NAME" --policy-arn
 "arn:aws:iam::aws:policy/SecretsManagerReadWrite" 2>/dev/null || true

        for policy in $(aws iam list-role-policies --role-name "$ADMIN_ROLE_NAME"
 --query 'PolicyNames[]' --output text 2>/dev/null || true); do
            aws iam delete-role-policy --role-name "$ADMIN_ROLE_NAME" --policy-
name "$policy" 2>/dev/null || true
        done

        echo "Deleting IAM role: $ADMIN_ROLE_NAME"
        aws iam delete-role --role-name "$ADMIN_ROLE_NAME" 2>/dev/null || true
    fi

    echo "Cleanup completed."
}

# Trap to ensure cleanup on script exit
trap 'echo "Script interrupted. Running cleanup..."; cleanup_resources' INT TERM
 EXIT

# Generate random identifiers for resources
ADMIN_ROLE_NAME="SecretsManagerAdmin-$(generate_random_id)"
RUNTIME_ROLE_NAME="RoleToRetrieveSecretAtRuntime-$(generate_random_id)"
SECRET_NAME="MyAPIKey-$(generate_random_id)"

echo "Using the following resource names:"
echo "Admin Role: $ADMIN_ROLE_NAME"
echo "Runtime Role: $RUNTIME_ROLE_NAME"
echo "Secret Name: $SECRET_NAME"
echo ""

# Step 1: Create IAM roles
echo "Creating IAM roles..."

# Create assume role policy document
ASSUME_ROLE_POLICY=$(cat <<'EOF'
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
)

# Create the SecretsManagerAdmin role
echo "Creating admin role: $ADMIN_ROLE_NAME"
ADMIN_ROLE_OUTPUT=$(aws iam create-role \
    --role-name "$ADMIN_ROLE_NAME" \
    --assume-role-policy-document "$ASSUME_ROLE_POLICY" 2>&1)

check_error "$ADMIN_ROLE_OUTPUT" "create-role for admin"
echo "$ADMIN_ROLE_OUTPUT"
aws iam tag-role --role-name "$ADMIN_ROLE_NAME" --tags Key=project,Value=doc-
smith Key=tutorial,Value=aws-secrets-manager-gs

# Attach the SecretsManagerReadWrite policy to the admin role
echo "Attaching SecretsManagerReadWrite policy to admin role"
ATTACH_POLICY_OUTPUT=$(aws iam attach-role-policy \
    --role-name "$ADMIN_ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/SecretsManagerReadWrite" 2>&1)

check_error "$ATTACH_POLICY_OUTPUT" "attach-role-policy for admin"
echo "Policy attached successfully"

# Create the RoleToRetrieveSecretAtRuntime role
echo "Creating runtime role: $RUNTIME_ROLE_NAME"
RUNTIME_ROLE_OUTPUT=$(aws iam create-role \
    --role-name "$RUNTIME_ROLE_NAME" \
    --assume-role-policy-document "$ASSUME_ROLE_POLICY" 2>&1)

check_error "$RUNTIME_ROLE_OUTPUT" "create-role for runtime"
echo "$RUNTIME_ROLE_OUTPUT"
aws iam tag-role --role-name "$RUNTIME_ROLE_NAME" --tags Key=project,Value=doc-
smith Key=tutorial,Value=aws-secrets-manager-gs
```

```
# Wait for roles to be fully created
echo "Waiting for IAM roles to be fully created..."
sleep 10

# Step 2: Create a secret in AWS Secrets Manager
echo "Creating secret in AWS Secrets Manager..."

# Generate secure secret value using environment variable or secure method
# WARNING: In production, use secure methods to inject secrets (AWS CodeBuild,
 parameter store, etc.)
if [ -z "${TUTORIAL_SECRET_VALUE:-}" ]; then
    SECRET_VALUE=$(python3 -c "import json;
 print(json.dumps({'ClientID':'my_client_id','ClientSecret':__import__('secrets').token_u
else
    SECRET_VALUE="$TUTORIAL_SECRET_VALUE"
fi

CREATE_SECRET_OUTPUT=$(aws secretsmanager create-secret \
    --name "$SECRET_NAME" \
    --description "API key for my application" \
    --secret-string "$SECRET_VALUE" \
    --add-replica-regions 'Region=us-east-1' \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=aws-secrets-manager-gs
 2>&1)

check_error "$CREATE_SECRET_OUTPUT" "create-secret"
echo "$CREATE_SECRET_OUTPUT"

# Get AWS account ID
echo "Getting AWS account ID..."
ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text 2>&1)
check_error "$ACCOUNT_ID" "get-caller-identity"
echo "Account ID: $ACCOUNT_ID"

# Get secret ARN for precise resource policy
echo "Getting secret ARN..."
SECRET_ARN=$(aws secretsmanager describe-secret \
    --secret-id "$SECRET_NAME" \
    --query 'ARN' \
    --output text 2>&1)
check_error "$SECRET_ARN" "describe-secret"

# Add resource policy to the secret with least privilege
```

```
echo "Adding resource policy to secret..."
RESOURCE_POLICY=$(cat <<EOF
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRuntimeRoleReadSecret",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::$ACCOUNT_ID:role/$RUNTIME_ROLE_NAME"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "$SECRET_ARN",
            "Condition": {
                "StringEquals": {
                    "secretsmanager:VersionStage": "AWSCURRENT"
                }
            }
        }
    ]
}
EOF
)

PUT_POLICY_OUTPUT=$(aws secretsmanager put-resource-policy \
    --secret-id "$SECRET_NAME" \
    --resource-policy "$RESOURCE_POLICY" \
    --block-public-policy 2>&1)

check_error "$PUT_POLICY_OUTPUT" "put-resource-policy"
echo "Resource policy added successfully"

# Enable rotation policy recommendation
echo "Enabling secret metadata tags for rotation tracking..."
aws secretsmanager tag-resource \
    --secret-id "$SECRET_NAME" \
    --tags Key=Purpose,Value=Tutorial Key=AutoRotation,Value=Recommended 2>/dev/
null || true

# Step 3: Demonstrate retrieving the secret
echo "Retrieving the secret value (for demonstration purposes)..."
GET_SECRET_OUTPUT=$(aws secretsmanager get-secret-value \
    --secret-id "$SECRET_NAME" 2>&1)
```

```
check_error "$GET_SECRET_OUTPUT" "get-secret-value"
echo "Secret retrieved successfully. Secret metadata:"
echo "$GET_SECRET_OUTPUT" | jq '{ARN: .ARN, Name: .Name,
 LastUpdatedDate: .LastUpdatedDate, VersionIdsToStages: .VersionIdsToStages}' 2>/
dev/null || echo "Secret metadata retrieved (jq not available)"

# Step 4: Update the secret with new values
echo "Updating the secret with new values..."
UPDATE_SECRET_VALUE=$(python3 -c "import json;
 print(json.dumps({'ClientID':'my_new_client_id','ClientSecret':__import__('secrets').tok

UPDATE_SECRET_OUTPUT=$(aws secretsmanager update-secret \
    --secret-id "$SECRET_NAME" \
    --secret-string "$UPDATE_SECRET_VALUE" 2>&1)

check_error "$UPDATE_SECRET_OUTPUT" "update-secret"
echo "Secret updated successfully"

# Step 5: Verify the updated secret
echo "Verifying the updated secret..."
VERIFY_SECRET_OUTPUT=$(aws secretsmanager get-secret-value \
    --secret-id "$SECRET_NAME" 2>&1)

check_error "$VERIFY_SECRET_OUTPUT" "get-secret-value for verification"
echo "Updated secret retrieved successfully. Secret metadata:"
echo "$VERIFY_SECRET_OUTPUT" | jq '{ARN: .ARN, Name: .Name,
 LastUpdatedDate: .LastUpdatedDate, VersionIdsToStages: .VersionIdsToStages}' 2>/
dev/null || echo "Secret metadata retrieved (jq not available)"

# Step 6: Display rotation recommendations
echo ""
echo "Rotation Configuration Recommendations:"
echo "========================================"
DESCRIBE_OUTPUT=$(aws secretsmanager describe-secret --secret-id "$SECRET_NAME"
 2>&1)
if echo "$DESCRIBE_OUTPUT" | grep -q "RotationRules"; then
    echo "Current rotation configuration:"
    echo "$DESCRIBE_OUTPUT" | jq '.RotationRules' 2>/dev/null || echo "Rotation
 rules available"
else
    echo "No automatic rotation configured. Consider enabling rotation with:"
    echo "aws secretsmanager rotate-secret --secret-id $SECRET_NAME --rotation-
lambda-arn arn:aws:lambda:REGION:ACCOUNT:function:FUNCTION_NAME --rotation-rules
 AutomaticallyAfterDays=30"
```

```
fi

echo ""
echo "============================================================"
echo "Tutorial completed successfully!"
echo ""
echo "Summary of what we did:"
echo "1. Created IAM roles for managing and retrieving secrets"
echo "2. Created a secret in AWS Secrets Manager with secure generation"
echo "3. Added a least-privilege resource policy to control access to the secret"
echo "4. Retrieved the secret value (simulating application access)"
echo "5. Updated the secret with cryptographically secure values"
echo "6. Verified the updated secret"
echo ""
echo "Security best practices applied:"
echo "- Used cryptographically secure random ID generation"
echo "- Applied least-privilege resource policies with version stages"
echo "- Tagged resources for rotation tracking"
echo "- Blocked public access to secrets"
echo "- Used ARN-specific permissions instead of wildcards"
echo ""
echo "Next steps you might want to consider:"
echo "- Enable automatic secret rotation with AWS Lambda"
echo "- Implement secret caching in your application"
echo "- Set up CloudTrail logging for secret access auditing"
echo "- Use AWS CodeGuru Reviewer to find hardcoded secrets in your code"
echo "- For multi-region applications, replicate your secrets across regions"
echo "- Configure VPC endpoints for private access to Secrets Manager"
echo ""

echo "Script completed at $(date)"
exit 0
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateRole
  - CreateSecret
  - DeleteRole
  - DeleteSecret
  - DetachRolePolicy

- GetCallerIdentity
- GetSecretValue
- PutResourcePolicy
- UpdateSecret

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Permette AWS Compute Optimizer Funzionalità di automazione per applicare le azioni consigliate

Il seguente esempio di codice mostra come: Questa politica basata sulle autorizzazioni consente alla funzionalità di AWS Compute Optimizer automazione di applicare le azioni consigliate

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "aco-automation.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Politica per abilitare l'automazione in tutta l'organizzazione

Il seguente esempio di codice mostra come questa politica basata sulle autorizzazioni consente l'automazione in tutta l'organizzazione

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/aco-
automation.amazonaws.com/AWSServiceRoleForComputeOptimizerAutomation",
            "Condition": {"StringLike": {"iam:AWSServiceName": "aco-
automation.amazonaws.com"}}
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PutRolePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/aco-
automation.amazonaws.com/AWSServiceRoleForComputeOptimizerAutomation"
        },
        {
            "Effect": "Allow",
            "Action": "aco-automation:UpdateEnrollmentConfiguration",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "aco-automation:AssociateAccounts",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "aco-automation:DisassociateAccounts",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "aco-automation:ListAccounts",
            "Resource": "*"
        }
    ]
```

```
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Politica per abilitare l'automazione per il tuo account

Il seguente esempio di codice mostra come questa politica basata sulle autorizzazioni abilita l'automazione per il tuo account

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/aco-
automation.amazonaws.com/AWSServiceRoleForComputeOptimizerAutomation",
            "Condition": {"StringLike": {"iam:AWSServiceName": "aco-
automation.amazonaws.com"}}
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PutRolePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/aco-
automation.amazonaws.com/AWSServiceRoleForComputeOptimizerAutomation"
        },
        {
            "Effect": "Allow",
            "Action": "aco-automation:UpdateEnrollmentConfiguration",
            "Resource": "*"
        }
    ]
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Politica per concedere l'accesso completo a Compute Optimizer Automation per un account di gestione di un'organizzazione

Il seguente esempio di codice mostra come: Questa politica basata sulle autorizzazioni concede l'accesso completo a Compute Optimizer Automation per un account di gestione di un'organizzazione.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "aco-automation:*",
               "ec2:DescribeVolumes",
               "organizations:ListAccounts",
               "organizations:DescribeOrganization",
               "organizations:DescribeAccount",
               "organizations:EnableAWSServiceAccess",
               "organizations:ListDelegatedAdministrators",
               "organizations:RegisterDelegatedAdministrator",
               "organizations:DeregisterDelegatedAdministrator"
            ],
            "Resource": "*"
        }
    ]
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Politica per concedere l'accesso completo a Compute Optimizer Automation per la versione standalone AWS account

Il seguente esempio di codice mostra come questa politica basata sulle autorizzazioni concede l'accesso completo a Compute Optimizer Automation per gli account autonomi. AWS

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "aco-automation:*",
            "ec2:DescribeVolumes"
            ],
            "Resource": "*"
        }
    ]
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Politica per concedere l'accesso in sola lettura a Compute Optimizer Automation per un account di gestione di un'organizzazione

Il seguente esempio di codice mostra come: Questa politica basata sulle autorizzazioni concede l'accesso in sola lettura a Compute Optimizer Automation per un account di gestione di un'organizzazione.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
               "aco-automation:GetEnrollmentConfiguration",
               "aco-automation:GetAutomationEvent",
               "aco-automation:GetAutomationRule",
               "aco-automation:ListAccounts",
               "aco-automation:ListAutomationEvents",
               "aco-automation:ListAutomationEventSteps",
               "aco-automation:ListAutomationEventSummaries",
               "aco-automation:ListAutomationRules",
               "aco-automation:ListAutomationRulePreview",
               "aco-automation:ListAutomationRulePreviewSummaries",
               "aco-automation:ListRecommendedActions",
               "aco-automation:ListRecommendedActionSummaries",
               "aco-automation:ListTagsForResource",
               "ec2:DescribeVolumes"
            ],
            "Resource": "*"
        }
    ]
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Politica per concedere l'accesso in sola lettura a Compute Optimizer Automation per uso autonomo AWS account

Il seguente esempio di codice mostra come: Questa politica basata sulle autorizzazioni concede l'accesso in sola lettura a Compute Optimizer Automation per gli account autonomi. AWS

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "aco-automation:GetEnrollmentConfiguration",
```

```
                "aco-automation:GetAutomationEvent",
                "aco-automation:GetAutomationRule",
                "aco-automation:ListAutomationEvents",
                "aco-automation:ListAutomationEventSteps",
                "aco-automation:ListAutomationEventSummaries",
                "aco-automation:ListAutomationRules",
                "aco-automation:ListAutomationRulePreview",
                "aco-automation:ListAutomationRulePreviewSummaries",
                "aco-automation:ListRecommendedActions",
                "aco-automation:ListRecommendedActionSummaries",
                "aco-automation:ListTagsForResource",
                "ec2:DescribeVolumes"
            ],
            "Resource": "*"
        }
    ]
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Politica per la concessione di autorizzazioni di ruolo collegate ai servizi per Compute Optimization Automation

Il seguente esempio di codice mostra come questa politica basata sulle autorizzazioni concede le autorizzazioni di ruolo collegate ai servizi per Compute Optimization Automation

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/aco-
automation.amazonaws.com/AWSServiceRoleForComputeOptimizerAutomation",
            "Condition": {"StringLike": {"iam:AWSServiceName": "aco-
automation.amazonaws.com"}}
        },
```

```
        {
            "Effect": "Allow",
            "Action": "iam:PutRolePolicy",
            "Resource": "arn:aws:iam::*:role/aws-service-role/aco-
automation.amazonaws.com/AWSServiceRoleForComputeOptimizerAutomation"
        }
    ]
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Ripristina una versione della policy IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Ottieni l'elenco delle versioni delle policy in ordine di data.
- Individua la versione predefinita della policy.
- Rendi predefinita la versione precedente della policy.
- Elimina la vecchia versione predefinita.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
def rollback_policy_version(policy_arn):
    """
    Rolls back to the previous default policy, if it exists.

    1. Gets the list of policy versions in order by date.
    2. Finds the default.
    3. Makes the previous policy the default.
```

```
    4. Deletes the old default version.

    :param policy_arn: The ARN of the policy to roll back.
    :return: The default version of the policy after the rollback.
    """
    try:
        policy_versions = sorted(
            iam.Policy(policy_arn).versions.all(),
            key=operator.attrgetter("create_date"),
        )
        logger.info("Got %s versions for %s.", len(policy_versions), policy_arn)
    except ClientError:
        logger.exception("Couldn't get versions for %s.", policy_arn)
        raise

    default_version = None
    rollback_version = None
    try:
        while default_version is None:
            ver = policy_versions.pop()
            if ver.is_default_version:
                default_version = ver
        rollback_version = policy_versions.pop()
        rollback_version.set_as_default()
        logger.info("Set %s as the default version.",
rollback_version.version_id)
        default_version.delete()
        logger.info("Deleted original default version %s.",
default_version.version_id)
    except IndexError:
        if default_version is None:
            logger.warning("No default version found for %s.", policy_arn)
        elif rollback_version is None:
            logger.warning(
                "Default version %s found for %s, but no previous version exists,
so "
                "nothing to roll back to.",
                default_version.version_id,
                policy_arn,
            )
    except ClientError:
        logger.exception("Couldn't roll back version for %s.", policy_arn)
        raise
    else:
```

```
        return rollback_version
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - DeletePolicyVersion
  - ListPolicyVersions
  - SetDefaultPolicyVersion

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

L'esempio di codice seguente mostra come:

- Creazione di ruoli IAM
- Crea un allarme CloudWatch
- Crea un modello di esperimento
- Esegui l'esperimento
- Verifica i risultati
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS FIS CPU Stress Test Tutorial Script
# This script automates the steps in the AWS FIS CPU stress test tutorial

#    approach using epoch time calculations that work across all Linux
 distributions

# Set up logging
LOG_FILE="fis-tutorial-$(date +%Y%m%d-%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting AWS FIS CPU Stress Test Tutorial Script"
echo "Logging to $LOG_FILE"
echo "============================================="

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2

    if echo "$output" | grep -i "error" > /dev/null; then
        # Ignore specific expected errors
        if [[ "$cmd" == *"aws fis get-experiment"* ]] && [[ "$output" ==
 *"ConfigurationFailure"* ]]; then
            echo "Note: Experiment failed due to configuration issue. This is
 expected in some cases."
            return 0
        fi

        echo "ERROR: Command failed: $cmd"
        echo "Output: $output"
        cleanup_on_error
        exit 1
    fi
}

# Function to clean up resources on error
cleanup_on_error() {
    echo "Error encountered. Cleaning up resources..."

    if [ -n "$EXPERIMENT_ID" ]; then
        echo "Stopping experiment $EXPERIMENT_ID if running..."
```

```
        aws fis stop-experiment --id "$EXPERIMENT_ID" 2>/dev/null || true
    fi

    if [ -n "$TEMPLATE_ID" ]; then
        echo "Deleting experiment template $TEMPLATE_ID..."
        aws fis delete-experiment-template --id "$TEMPLATE_ID" || true
    fi

    if [ -n "$INSTANCE_ID" ]; then
        echo "Terminating EC2 instance $INSTANCE_ID..."
        aws ec2 terminate-instances --instance-ids "$INSTANCE_ID" || true
    fi

    if [ -n "$ALARM_NAME" ]; then
        echo "Deleting CloudWatch alarm $ALARM_NAME..."
        aws cloudwatch delete-alarms --alarm-names "$ALARM_NAME" || true
    fi

    if [ -n "$INSTANCE_PROFILE_NAME" ]; then
        echo "Removing role from instance profile..."
        aws iam remove-role-from-instance-profile --instance-profile-name
"$INSTANCE_PROFILE_NAME" --role-name "$EC2_ROLE_NAME" || true

        echo "Deleting instance profile..."
        aws iam delete-instance-profile --instance-profile-name
"$INSTANCE_PROFILE_NAME" || true
    fi

    if [ -n "$FIS_ROLE_NAME" ]; then
        echo "Deleting FIS role policy..."
        aws iam delete-role-policy --role-name "$FIS_ROLE_NAME" --policy-name
"$FIS_POLICY_NAME" || true

        echo "Deleting FIS role..."
        aws iam delete-role --role-name "$FIS_ROLE_NAME" || true
    fi

    if [ -n "$EC2_ROLE_NAME" ]; then
        echo "Detaching policy from EC2 role..."
        aws iam detach-role-policy --role-name "$EC2_ROLE_NAME" --policy-arn
"arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" || true

        echo "Deleting EC2 role..."
        aws iam delete-role --role-name "$EC2_ROLE_NAME" || true
```

```
    fi

    echo "Cleanup completed."
}

# Generate unique identifiers for resources
TIMESTAMP=$(date +%Y%m%d%H%M%S)
FIS_ROLE_NAME="FISRole-${TIMESTAMP}"
FIS_POLICY_NAME="FISPolicy-${TIMESTAMP}"
EC2_ROLE_NAME="EC2SSMRole-${TIMESTAMP}"
INSTANCE_PROFILE_NAME="EC2SSMProfile-${TIMESTAMP}"
ALARM_NAME="FIS-CPU-Alarm-${TIMESTAMP}"

# Track created resources
CREATED_RESOURCES=()

echo "Step 1: Creating IAM role for AWS FIS"
# Create trust policy file for AWS FIS
cat > fis-trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "fis.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create IAM role for FIS
echo "Creating IAM role $FIS_ROLE_NAME for AWS FIS..."
FIS_ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$FIS_ROLE_NAME" \
  --assume-role-policy-document file://fis-trust-policy.json)
check_error "$FIS_ROLE_OUTPUT" "aws iam create-role"
aws iam tag-role --role-name "$FIS_ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-fault-injection-service-gs
CREATED_RESOURCES+=("IAM Role: $FIS_ROLE_NAME")

# Create policy document for SSM actions
```

```
cat > fis-ssm-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:SendCommand",
        "ssm:ListCommands",
        "ssm:ListCommandInvocations"
      ],
      "Resource": "*"
    }
  ]
}
EOF

# Attach policy to the role
echo "Attaching policy $FIS_POLICY_NAME to role $FIS_ROLE_NAME..."
FIS_POLICY_OUTPUT=$(aws iam put-role-policy \
  --role-name "$FIS_ROLE_NAME" \
  --policy-name "$FIS_POLICY_NAME" \
  --policy-document file://fis-ssm-policy.json)
check_error "$FIS_POLICY_OUTPUT" "aws iam put-role-policy"
CREATED_RESOURCES+=("IAM Policy: $FIS_POLICY_NAME attached to $FIS_ROLE_NAME")

echo "Step 2: Creating IAM role for EC2 instance with SSM permissions"
# Create trust policy file for EC2
cat > ec2-trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create IAM role for EC2
```

```
echo "Creating IAM role $EC2_ROLE_NAME for EC2 instance..."
EC2_ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$EC2_ROLE_NAME" \
  --assume-role-policy-document file://ec2-trust-policy.json)
check_error "$EC2_ROLE_OUTPUT" "aws iam create-role"
aws iam tag-role --role-name "$EC2_ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-fault-injection-service-gs
CREATED_RESOURCES+=("IAM Role: $EC2_ROLE_NAME")

# Attach SSM policy to the EC2 role
echo "Attaching AmazonSSMManagedInstanceCore policy to role $EC2_ROLE_NAME..."
EC2_POLICY_OUTPUT=$(aws iam attach-role-policy \
  --role-name "$EC2_ROLE_NAME" \
  --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore)
check_error "$EC2_POLICY_OUTPUT" "aws iam attach-role-policy"
CREATED_RESOURCES+=("IAM Policy: AmazonSSMManagedInstanceCore attached to
 $EC2_ROLE_NAME")

# Create instance profile
echo "Creating instance profile $INSTANCE_PROFILE_NAME..."
PROFILE_OUTPUT=$(aws iam create-instance-profile \
  --instance-profile-name "$INSTANCE_PROFILE_NAME")
check_error "$PROFILE_OUTPUT" "aws iam create-instance-profile"
CREATED_RESOURCES+=("IAM Instance Profile: $INSTANCE_PROFILE_NAME")

# Add role to instance profile
echo "Adding role $EC2_ROLE_NAME to instance profile $INSTANCE_PROFILE_NAME..."
ADD_ROLE_OUTPUT=$(aws iam add-role-to-instance-profile \
  --instance-profile-name "$INSTANCE_PROFILE_NAME" \
  --role-name "$EC2_ROLE_NAME")
check_error "$ADD_ROLE_OUTPUT" "aws iam add-role-to-instance-profile"

# Wait for role to propagate
echo "Waiting for IAM role to propagate..."
sleep 10

echo "Step 3: Launching EC2 instance"
# Get the latest Amazon Linux 2 AMI ID
echo "Finding latest Amazon Linux 2 AMI..."
AMI_ID=$(aws ec2 describe-images \
  --owners amazon \
  --filters "Name=name,Values=amzn2-ami-hvm-*-x86_64-gp2"
 "Name=state,Values=available" \
  --query "sort_by(Images, &CreationDate)[-1].ImageId" \
```

```
  --output text)
check_error "$AMI_ID" "aws ec2 describe-images"
echo "Using AMI: $AMI_ID"

# Launch EC2 instance
echo "Launching EC2 instance with AMI $AMI_ID..."
INSTANCE_OUTPUT=$(aws ec2 run-instances \
  --image-id "$AMI_ID" \
  --instance-type t2.micro \
  --iam-instance-profile Name="$INSTANCE_PROFILE_NAME" \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=FIS-Test-
Instance},{Key=project,Value=doc-smith},{Key=tutorial,Value=aws-fault-injection-
service-gs}]')
check_error "$INSTANCE_OUTPUT" "aws ec2 run-instances"

# Get instance ID
INSTANCE_ID=$(echo "$INSTANCE_OUTPUT" | grep -i "InstanceId" | head -1 | awk -
F'"' '{print $4}')
if [ -z "$INSTANCE_ID" ]; then
    echo "Failed to get instance ID"
    cleanup_on_error
    exit 1
fi
echo "Launched instance: $INSTANCE_ID"
CREATED_RESOURCES+=("EC2 Instance: $INSTANCE_ID")

# Enable detailed monitoring
echo "Enabling detailed monitoring for instance $INSTANCE_ID..."
MONITOR_OUTPUT=$(aws ec2 monitor-instances --instance-ids "$INSTANCE_ID")
check_error "$MONITOR_OUTPUT" "aws ec2 monitor-instances"

# Wait for instance to be running and status checks to pass
echo "Waiting for instance to be ready..."
aws ec2 wait instance-running --instance-ids "$INSTANCE_ID"
aws ec2 wait instance-status-ok --instance-ids "$INSTANCE_ID"
echo "Instance is ready"

echo "Step 4: Creating CloudWatch alarm for CPU utilization"
# Create CloudWatch alarm
echo "Creating CloudWatch alarm $ALARM_NAME..."
ALARM_OUTPUT=$(aws cloudwatch put-metric-alarm \
  --alarm-name "$ALARM_NAME" \
  --alarm-description "Alarm when CPU exceeds 50%" \
  --metric-name CPUUtilization \
```

```
  --namespace AWS/EC2 \
  --statistic Maximum \
  --period 60 \
  --threshold 50 \
  --comparison-operator GreaterThanOrEqualToThreshold \
  --dimensions "Name=InstanceId,Value=$INSTANCE_ID" \
  --evaluation-periods 1)
check_error "$ALARM_OUTPUT" "aws cloudwatch put-metric-alarm"
CREATED_RESOURCES+=("CloudWatch Alarm: $ALARM_NAME")

# Get the alarm ARN
echo "Getting CloudWatch alarm ARN..."
ALARM_ARN_OUTPUT=$(aws cloudwatch describe-alarms \
  --alarm-names "$ALARM_NAME")
check_error "$ALARM_ARN_OUTPUT" "aws cloudwatch describe-alarms"
ALARM_ARN=$(echo "$ALARM_ARN_OUTPUT" | grep -i "AlarmArn" | head -1 | awk -F'"'
 '{print $4}')
if [ -z "$ALARM_ARN" ]; then
    echo "Failed to get alarm ARN"
    cleanup_on_error
    exit 1
fi
echo "Alarm ARN: $ALARM_ARN"

# Wait for the alarm to initialize and reach OK state
echo "Waiting for CloudWatch alarm to initialize (60 seconds)..."
sleep 60

# Check alarm state
echo "Checking alarm state..."
ALARM_STATE_OUTPUT=$(aws cloudwatch describe-alarms \
  --alarm-names "$ALARM_NAME")
ALARM_STATE=$(echo "$ALARM_STATE_OUTPUT" | grep -i "StateValue" | head -1 | awk -
F'"' '{print $4}')
echo "Current alarm state: $ALARM_STATE"

# If alarm is not in OK state, wait longer or generate some baseline metrics
if [ "$ALARM_STATE" != "OK" ]; then
    echo "Alarm not in OK state. Waiting for alarm to stabilize (additional 60
 seconds)..."
    sleep 60

    # Check alarm state again
    ALARM_STATE_OUTPUT=$(aws cloudwatch describe-alarms \
```

```
      --alarm-names "$ALARM_NAME")
    ALARM_STATE=$(echo "$ALARM_STATE_OUTPUT" | grep -i "StateValue" | head -1 |
 awk -F'"' '{print $4}')
    echo "Updated alarm state: $ALARM_STATE"

    if [ "$ALARM_STATE" != "OK" ]; then
        echo "Warning: Alarm still not in OK state. Experiment may fail to
 start."
    fi
fi

echo "Step 5: Creating AWS FIS experiment template"
# Get the IAM role ARN
echo "Getting IAM role ARN for $FIS_ROLE_NAME..."
ROLE_ARN_OUTPUT=$(aws iam get-role \
  --role-name "$FIS_ROLE_NAME")
check_error "$ROLE_ARN_OUTPUT" "aws iam get-role"
ROLE_ARN=$(echo "$ROLE_ARN_OUTPUT" | grep -i "Arn" | head -1 | awk -F'"' '{print
 $4}')
if [ -z "$ROLE_ARN" ]; then
    echo "Failed to get role ARN"
    cleanup_on_error
    exit 1
fi
echo "Role ARN: $ROLE_ARN"

# Get account ID and region
ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
REGION=$(aws configure get region)
if [ -z "$REGION" ]; then
    REGION="us-east-1"  # Default to us-east-1 if region not set
fi
INSTANCE_ARN="arn:aws:ec2:${REGION}:${ACCOUNT_ID}:instance/${INSTANCE_ID}"
echo "Instance ARN: $INSTANCE_ARN"

# Create experiment template - Fixed JSON escaping issue
cat > experiment-template.json << EOF
{
  "description": "Test CPU stress predefined SSM document",
  "targets": {
    "testInstance": {
      "resourceType": "aws:ec2:instance",
      "resourceArns": ["$INSTANCE_ARN"],
      "selectionMode": "ALL"
```

```
    }
  },
  "actions": {
    "runCpuStress": {
      "actionId": "aws:ssm:send-command",
      "parameters": {
        "documentArn": "arn:aws:ssm:$REGION::document/AWSFIS-Run-CPU-Stress",
        "documentParameters": "{\"DurationSeconds\":\"120\"}",
        "duration": "PT5M"
      },
      "targets": {
        "Instances": "testInstance"
      }
    }
  },
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": "$ALARM_ARN"
    }
  ],
  "roleArn": "$ROLE_ARN",
  "tags": {
    "Name": "FIS-CPU-Stress-Experiment",
    "project": "doc-smith",
    "tutorial": "aws-fault-injection-service-gs"
  }
}
EOF

# Create experiment template
echo "Creating AWS FIS experiment template..."
TEMPLATE_OUTPUT=$(aws fis create-experiment-template --cli-input-json file://
experiment-template.json)
check_error "$TEMPLATE_OUTPUT" "aws fis create-experiment-template"
TEMPLATE_ID=$(echo "$TEMPLATE_OUTPUT" | grep -i "id" | head -1 | awk -F'"'
 '{print $4}')
if [ -z "$TEMPLATE_ID" ]; then
    echo "Failed to get template ID"
    cleanup_on_error
    exit 1
fi
echo "Experiment template created with ID: $TEMPLATE_ID"
CREATED_RESOURCES+=("FIS Experiment Template: $TEMPLATE_ID")
```

```
echo "Step 6: Starting the experiment"
# Start the experiment
echo "Starting AWS FIS experiment using template $TEMPLATE_ID..."
EXPERIMENT_OUTPUT=$(aws fis start-experiment \
  --experiment-template-id "$TEMPLATE_ID" \
  --tags '{"Name": "FIS-CPU-Stress-Run"}')
check_error "$EXPERIMENT_OUTPUT" "aws fis start-experiment"
EXPERIMENT_ID=$(echo "$EXPERIMENT_OUTPUT" | grep -i "id" | head -1 | awk -F'"'
 '{print $4}')
if [ -z "$EXPERIMENT_ID" ]; then
    echo "Failed to get experiment ID"
    cleanup_on_error
    exit 1
fi
echo "Experiment started with ID: $EXPERIMENT_ID"
CREATED_RESOURCES+=("FIS Experiment: $EXPERIMENT_ID")

echo "Step 7: Tracking experiment progress"
# Track experiment progress
echo "Tracking experiment progress..."
MAX_CHECKS=30
CHECK_COUNT=0
EXPERIMENT_STATE=""

while [ $CHECK_COUNT -lt $MAX_CHECKS ]; do
    EXPERIMENT_INFO=$(aws fis get-experiment --id "$EXPERIMENT_ID")
    # Don't check for errors here, as we expect some experiments to fail

    EXPERIMENT_STATE=$(echo "$EXPERIMENT_INFO" | grep -i "status" | head -1 | awk
 -F'"' '{print $4}')
    echo "Experiment state: $EXPERIMENT_STATE"

    if [ "$EXPERIMENT_STATE" == "completed" ] || [ "$EXPERIMENT_STATE" ==
 "stopped" ] || [ "$EXPERIMENT_STATE" == "failed" ]; then
        # Show the reason for the state
        REASON=$(echo "$EXPERIMENT_INFO" | grep -i "reason" | head -1 | awk -F'"'
 '{print $4}')
        if [ -n "$REASON" ]; then
            echo "Reason: $REASON"
        fi
        break
    fi
```

```
    echo "Waiting 10 seconds before checking again..."
    sleep 10
    CHECK_COUNT=$((CHECK_COUNT + 1))
done

if [ $CHECK_COUNT -eq $MAX_CHECKS ]; then
    echo "Experiment is taking longer than expected. You can check its status
 later using:"
    echo "aws fis get-experiment --id $EXPERIMENT_ID"
fi

echo "Step 8: Verifying experiment results"
# Check CloudWatch alarm state
echo "Checking CloudWatch alarm state..."
ALARM_STATE_OUTPUT=$(aws cloudwatch describe-alarms --alarm-names "$ALARM_NAME")
check_error "$ALARM_STATE_OUTPUT" "aws cloudwatch describe-alarms"
echo "$ALARM_STATE_OUTPUT"

# Get CPU utilization metrics
echo "Getting CPU utilization metrics..."
END_TIME=$(date -u +"%Y-%m-%dT%H:%M:%SZ")

# FIXED: Cross-platform compatible way to calculate time 10 minutes ago
# This approach uses epoch seconds and basic arithmetic which works on all Linux
 distributions
CURRENT_EPOCH=$(date +%s)
TEN_MINUTES_AGO_EPOCH=$((CURRENT_EPOCH - 600))
START_TIME=$(date -u -d "@$TEN_MINUTES_AGO_EPOCH" +"%Y-%m-%dT%H:%M:%SZ" 2>/dev/
null || date -u -r "$TEN_MINUTES_AGO_EPOCH" +"%Y-%m-%dT%H:%M:%SZ")

# Create metric query file
cat > metric-query.json << EOF
[
  {
    "Id": "cpu",
    "MetricStat": {
      "Metric": {
        "Namespace": "AWS/EC2",
        "MetricName": "CPUUtilization",
        "Dimensions": [
          {
            "Name": "InstanceId",
            "Value": "$INSTANCE_ID"
          }
```

```
        ]
      },
      "Period": 60,
      "Stat": "Maximum"
    }
  }
]
EOF

METRICS_OUTPUT=$(aws cloudwatch get-metric-data \
  --start-time "$START_TIME" \
  --end-time "$END_TIME" \
  --metric-data-queries file://metric-query.json)
check_error "$METRICS_OUTPUT" "aws cloudwatch get-metric-data"
echo "CPU Utilization Metrics:"
echo "$METRICS_OUTPUT"

# Display summary of created resources
echo ""
echo "==========================================="
echo "RESOURCES CREATED"
echo "==========================================="
for resource in "${CREATED_RESOURCES[@]}"; do
    echo "- $resource"
done
echo "==========================================="

# Prompt for cleanup
echo ""
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    echo "Starting cleanup process..."

    # Stop experiment if still running
    if [ "$EXPERIMENT_STATE" != "completed" ] && [ "$EXPERIMENT_STATE" !=
 "stopped" ] && [ "$EXPERIMENT_STATE" != "failed" ]; then
        echo "Stopping experiment $EXPERIMENT_ID..."
        STOP_OUTPUT=$(aws fis stop-experiment --id "$EXPERIMENT_ID")
        check_error "$STOP_OUTPUT" "aws fis stop-experiment"
```

```
        echo "Waiting for experiment to stop..."
        sleep 10
    fi

    # Delete experiment template
    echo "Deleting experiment template $TEMPLATE_ID..."
    DELETE_TEMPLATE_OUTPUT=$(aws fis delete-experiment-template --id
"$TEMPLATE_ID")
    check_error "$DELETE_TEMPLATE_OUTPUT" "aws fis delete-experiment-template"

    # Delete CloudWatch alarm
    echo "Deleting CloudWatch alarm $ALARM_NAME..."
    DELETE_ALARM_OUTPUT=$(aws cloudwatch delete-alarms --alarm-names
"$ALARM_NAME")
    check_error "$DELETE_ALARM_OUTPUT" "aws cloudwatch delete-alarms"

    # Terminate EC2 instance
    echo "Terminating EC2 instance $INSTANCE_ID..."
    TERMINATE_OUTPUT=$(aws ec2 terminate-instances --instance-ids "$INSTANCE_ID")
    check_error "$TERMINATE_OUTPUT" "aws ec2 terminate-instances"
    echo "Waiting for instance to terminate..."
    aws ec2 wait instance-terminated --instance-ids "$INSTANCE_ID"

    # Clean up IAM resources
    echo "Removing role from instance profile..."
    REMOVE_ROLE_OUTPUT=$(aws iam remove-role-from-instance-profile \
      --instance-profile-name "$INSTANCE_PROFILE_NAME" \
      --role-name "$EC2_ROLE_NAME")
    check_error "$REMOVE_ROLE_OUTPUT" "aws iam remove-role-from-instance-profile"

    echo "Deleting instance profile..."
    DELETE_PROFILE_OUTPUT=$(aws iam delete-instance-profile \
      --instance-profile-name "$INSTANCE_PROFILE_NAME")
    check_error "$DELETE_PROFILE_OUTPUT" "aws iam delete-instance-profile"

    echo "Deleting FIS role policy..."
    DELETE_POLICY_OUTPUT=$(aws iam delete-role-policy \
      --role-name "$FIS_ROLE_NAME" \
      --policy-name "$FIS_POLICY_NAME")
    check_error "$DELETE_POLICY_OUTPUT" "aws iam delete-role-policy"

    echo "Detaching policy from EC2 role..."
    DETACH_POLICY_OUTPUT=$(aws iam detach-role-policy \
      --role-name "$EC2_ROLE_NAME" \
```

```
      --policy-arn "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore")
    check_error "$DETACH_POLICY_OUTPUT" "aws iam detach-role-policy"

    echo "Deleting FIS role..."
    DELETE_FIS_ROLE_OUTPUT=$(aws iam delete-role \
      --role-name "$FIS_ROLE_NAME")
    check_error "$DELETE_FIS_ROLE_OUTPUT" "aws iam delete-role"

    echo "Deleting EC2 role..."
    DELETE_EC2_ROLE_OUTPUT=$(aws iam delete-role \
      --role-name "$EC2_ROLE_NAME")
    check_error "$DELETE_EC2_ROLE_OUTPUT" "aws iam delete-role"

    # Clean up temporary files
    echo "Cleaning up temporary files..."
    rm -f fis-trust-policy.json ec2-trust-policy.json fis-ssm-policy.json
 experiment-template.json metric-query.json

    echo "Cleanup completed successfully."
else
    echo "Cleanup skipped. Resources will remain in your AWS account."
    echo "You can manually clean up the resources listed above."
fi

echo ""
echo "Script execution completed."
echo "Log file: $LOG_FILE"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AddRoleToInstanceProfile
  - AttachRolePolicy
  - CreateExperimentTemplate
  - CreateInstanceProfile
  - CreateRole
  - DeleteAlarms
  - DeleteExperimentTemplate
  - DeleteInstanceProfile

- DeleteRole
- DeleteRolePolicy
- DescribeAlarms
- DescribeImages
- DetachRolePolicy
- GetCallerIdentity
- GetExperiment
- GetMetricData
- GetRole
- MonitorInstances
- PutMetricAlarm
- PutRolePolicy
- RemoveRoleFromInstanceProfile
- RunInstances
- StartExperiment
- StopExperiment
- TerminateInstances
- Attendi

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Configurare il controllo Attribute-Based degli accessi per DynamoDB utilizzando AWS Command Line Interface v2

Il seguente esempio di codice mostra come implementare Attribute-Based Access Control (ABAC) per DynamoDB.

- Creare una policy IAM per il controllo degli accessi basato su attributi (ABAC).
- Creare tabelle con tag per reparti diversi.
- Elencare e filtrare le tabelle in base ai tag.

Bash

AWS CLI con lo script Bash

Creare una policy IAM per il controllo degli accessi basato su attributi (ABAC).

```
# Step 1: Create a policy document for ABAC
cat > abac-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:DeleteItem",
        "dynamodb:BatchWriteItem"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}",
          "aws:ResourceTag/Environment": "Development"
        }
      }
    }
  ]
}
```

```
EOF

# Step 2: Create the IAM policy
aws iam create-policy \
    --policy-name DynamoDBDepartmentBasedAccess \
    --policy-document file://abac-policy.json
```

Creare tabelle con tag per reparti diversi.

```
# Create a DynamoDB table with tags for ABAC
aws dynamodb create-table \
    --table-name FinanceData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Finance \
        Key=Environment,Value=Development

# Create another table with different tags
aws dynamodb create-table \
    --table-name MarketingData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Marketing \
        Key=Environment,Value=Production
```

Elencare e filtrare le tabelle in base ai tag.

```
# List all DynamoDB tables
echo "Listing all tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nGetting ARNs for all tables:"
```

```
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text |
 xargs -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn"
 --output text)

# For each table ARN, list its tags
echo -e "\nListing tags for each table:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    echo -e "\nTags for table: $TABLE_NAME"
    aws dynamodb list-tags-of-resource --resource-arn $ARN
done

# Example: Find tables with a specific tag
echo -e "\nFinding tables with Environment=Production tag:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - CreatePolicy
  - CreateTable
  - ListTables

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Configurazione della gestione dei sistemi

L'esempio di codice seguente mostra come:

- Creare autorizzazioni IAM per Systems Manager
- Creare un ruolo IAM per Systems Manager

- Configurazione di Systems Manager
- Verifica la configurazione
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS Systems Manager Setup Script
# This script sets up AWS Systems Manager for a single account and region
#
# Version 17 fixes:
# 1. Added cloudformation.amazonaws.com to the IAM role trust policy
# 2. Systems Manager Quick Setup uses CloudFormation for deployments, so the role
 must trust CloudFormation service

# Initialize log file
LOG_FILE="ssm_setup_$(date +%Y%m%d_%H%M%S).log"
UNIQUE_ID=$(openssl rand -hex 4)
echo "Starting AWS Systems Manager setup at $(date)" > "$LOG_FILE"

# Function to log commands and their outputs with immediate terminal display
log_cmd() {
    echo "$(date): Running command: $1" | tee -a "$LOG_FILE"
    local output
    output=$(eval "$1" 2>&1)
    local status=$?
    echo "$output" | tee -a "$LOG_FILE"
    return $status
}

# Function to check for errors in command output
```

```
check_error() {
    local cmd_output="$1"
    local cmd_status="$2"
    local error_msg="$3"

    if [[ $cmd_status -ne 0 || "$cmd_output" =~ [Ee][Rr][Rr][Oo][Rr] ]]; then
        echo "ERROR: $error_msg" | tee -a "$LOG_FILE"
        echo "Command output: $cmd_output" | tee -a "$LOG_FILE"
        cleanup_on_error
        exit 1
    fi
}

# Array to track created resources for cleanup
declare -a CREATED_RESOURCES

# Function to add a resource to the tracking array
track_resource() {
    local resource_type="$1"
    local resource_id="$2"
    CREATED_RESOURCES+=("$resource_type:$resource_id")
    echo "Tracked resource: $resource_type:$resource_id" | tee -a "$LOG_FILE"
}

# Function to clean up resources on error
cleanup_on_error() {
    echo "" | tee -a "$LOG_FILE"
    echo "========================================" | tee -a "$LOG_FILE"
    echo "ERROR OCCURRED - CLEANING UP RESOURCES" | tee -a "$LOG_FILE"
    echo "========================================" | tee -a "$LOG_FILE"
    echo "The following resources were created:" | tee -a "$LOG_FILE"

    # Display resources in reverse order
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        echo "${CREATED_RESOURCES[$i]}" | tee -a "$LOG_FILE"
    done

    echo "" | tee -a "$LOG_FILE"
    echo "Attempting to clean up resources..." | tee -a "$LOG_FILE"

    # Clean up resources in reverse order
    cleanup_resources
}
```

```
# Function to clean up all created resources
cleanup_resources() {
    # Process resources in reverse order (last created, first deleted)
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        IFS=':' read -r resource_type resource_id <<< "${CREATED_RESOURCES[$i]}"

        echo "Deleting $resource_type: $resource_id" | tee -a "$LOG_FILE"

        case "$resource_type" in
            "IAM_POLICY")
                # Delete the policy (detachment should have been handled when the
 role was deleted)
                log_cmd "aws iam delete-policy --policy-arn $resource_id" || true
                ;;
            "IAM_ROLE")
                # Detach all policies from the role first
                if [[ -n "$POLICY_ARN" ]]; then
                    log_cmd "aws iam detach-role-policy --role-name $resource_id
 --policy-arn $POLICY_ARN" || true
                fi

                # Delete the role
                log_cmd "aws iam delete-role --role-name $resource_id" || true
                ;;
            "SSM_CONFIG_MANAGER")
                log_cmd "aws ssm-quicksetup delete-configuration-manager --
manager-arn $resource_id" || true
                ;;
            *)
                echo "Unknown resource type: $resource_type, cannot delete
 automatically" | tee -a "$LOG_FILE"
                ;;
        esac
    done

    echo "Cleanup completed" | tee -a "$LOG_FILE"

    # Clean up temporary files
    rm -f ssm-onboarding-policy.json trust-policy.json ssm-config.json 2>/dev/
null || true
}

# Main script execution
echo "AWS Systems Manager Setup Script"
```

```
echo "================================="
echo "This script will set up AWS Systems Manager for a single account and
 region."
echo "It will create IAM policies and roles, then enable Systems Manager
 features."
echo ""

# Get the current AWS region
CURRENT_REGION=$(aws configure get region)
if [[ -z "$CURRENT_REGION" ]]; then
    echo "No AWS region configured. Please specify a region:"
    CURRENT_REGION="${AWS_DEFAULT_REGION:-us-west-2}"
    if [[ -z "$CURRENT_REGION" ]]; then
        echo "ERROR: A region must be specified" | tee -a "$LOG_FILE"
        exit 1
    fi
fi

echo "Using AWS region: $CURRENT_REGION" | tee -a "$LOG_FILE"

# Step 1: Create IAM policy for Systems Manager onboarding
echo "Step 1: Creating IAM policy for Systems Manager onboarding..."

# Create policy document
cat > ssm-onboarding-policy.json << 'EOF'
{
   "Version":"2012-10-17",
   "Statement": [
     {
       "Sid": "QuickSetupActions",
       "Effect": "Allow",
       "Action": [
         "ssm-quicksetup:*"
       ],
       "Resource": "*"
     },
     {
       "Sid": "SsmReadOnly",
       "Effect": "Allow",
       "Action": [
         "ssm:DescribeAutomationExecutions",
         "ssm:GetAutomationExecution",
         "ssm:ListAssociations",
         "ssm:DescribeAssociation",
```

```
          "ssm:ListDocuments",
          "ssm:ListResourceDataSync",
          "ssm:DescribePatchBaselines",
          "ssm:GetPatchBaseline",
          "ssm:DescribeMaintenanceWindows",
          "ssm:DescribeMaintenanceWindowTasks"
        ],
        "Resource": "*"
      },
      {
        "Sid": "SsmDocument",
        "Effect": "Allow",
        "Action": [
          "ssm:GetDocument",
          "ssm:DescribeDocument"
        ],
        "Resource": [
          "arn:aws:ssm:*:*:document/AWSQuickSetupType-*",
          "arn:aws:ssm:*:*:document/AWS-EnableExplorer"
        ]
      },
      {
        "Sid": "SsmEnableExplorer",
        "Effect": "Allow",
        "Action": "ssm:StartAutomationExecution",
        "Resource": "arn:aws:ssm:*:*:automation-definition/AWS-EnableExplorer:*"
      },
      {
        "Sid": "SsmExplorerRds",
        "Effect": "Allow",
        "Action": [
          "ssm:GetOpsSummary",
          "ssm:CreateResourceDataSync",
          "ssm:UpdateResourceDataSync"
        ],
        "Resource": "arn:aws:ssm:*:*:resource-data-sync/AWS-QuickSetup-*"
      },
      {
        "Sid": "OrgsReadOnly",
        "Effect": "Allow",
        "Action": [
          "organizations:DescribeAccount",
          "organizations:DescribeOrganization",
          "organizations:ListDelegatedAdministrators",
```

```
        "organizations:ListRoots",
        "organizations:ListParents",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:DescribeOrganizationalUnit",
        "organizations:ListAWSServiceAccessForOrganization"
      ],
      "Resource": "*"
    },
    {
      "Sid": "OrgsAdministration",
      "Effect": "Allow",
      "Action": [
        "organizations:EnableAWSServiceAccess",
        "organizations:RegisterDelegatedAdministrator",
        "organizations:DeregisterDelegatedAdministrator"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "organizations:ServicePrincipal": [
            "ssm.amazonaws.com",
            "ssm-quicksetup.amazonaws.com",
            "member.org.stacksets.cloudformation.amazonaws.com",
            "resource-explorer-2.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "CfnReadOnly",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ListStacks",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStackSets",
        "cloudformation:DescribeOrganizationsAccess"
      ],
      "Resource": "*"
    },
    {
      "Sid": "OrgCfnAccess",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ActivateOrganizationsAccess"
```

```
        ],
        "Resource": "*"
      },
      {
        "Sid": "CfnStackActions",
        "Effect": "Allow",
        "Action": [
          "cloudformation:CreateStack",
          "cloudformation:DeleteStack",
          "cloudformation:DescribeStackResources",
          "cloudformation:DescribeStackEvents",
          "cloudformation:GetTemplate",
          "cloudformation:RollbackStack",
          "cloudformation:TagResource",
          "cloudformation:UntagResource",
          "cloudformation:UpdateStack"
        ],
        "Resource": [
          "arn:aws:cloudformation:*:*:stack/StackSet-AWS-QuickSetup-*",
          "arn:aws:cloudformation:*:*:stack/AWS-QuickSetup-*",
          "arn:aws:cloudformation:*:*:type/resource/*"
        ]
      },
      {
        "Sid": "CfnStackSetActions",
        "Effect": "Allow",
        "Action": [
          "cloudformation:CreateStackInstances",
          "cloudformation:CreateStackSet",
          "cloudformation:DeleteStackInstances",
          "cloudformation:DeleteStackSet",
          "cloudformation:DescribeStackInstance",
          "cloudformation:DetectStackSetDrift",
          "cloudformation:ListStackInstanceResourceDrifts",
          "cloudformation:DescribeStackSet",
          "cloudformation:DescribeStackSetOperation",
          "cloudformation:ListStackInstances",
          "cloudformation:ListStackSetOperations",
          "cloudformation:ListStackSetOperationResults",
          "cloudformation:TagResource",
          "cloudformation:UntagResource",
          "cloudformation:UpdateStackSet"
        ],
        "Resource": [
```

```
        "arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-*",
        "arn:aws:cloudformation:*:*:type/resource/*",
        "arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-*:*"
      ]
    },
    {
      "Sid": "ValidationReadonlyActions",
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles",
        "iam:GetRole"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IamRolesMgmt",
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:GetRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:GetRolePolicy",
        "iam:ListRolePolicies"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ]
    },
    {
      "Sid": "IamPassRole",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
```

```
            "ssm.amazonaws.com",
            "ssm-quicksetup.amazonaws.com",
            "cloudformation.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "IamRolesPoliciesMgmt",
      "Effect": "Allow",
      "Action": [
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ],
      "Condition": {
        "ArnEquals": {
          "iam:PolicyARN": [
            "arn:aws:iam::aws:policy/
AWSSystemsManagerEnableExplorerExecutionPolicy",
            "arn:aws:iam::aws:policy/AWSQuickSetupSSMDeploymentRolePolicy"
          ]
        }
      }
    },
    {
      "Sid": "CfnStackSetsSLR",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/
stacksets.cloudformation.amazonaws.com/
AWSServiceRoleForCloudFormationStackSetsOrgAdmin",
        "arn:aws:iam::*:role/aws-service-role/ssm.amazonaws.com/
AWSServiceRoleForAmazonSSM",
        "arn:aws:iam::*:role/aws-service-role/
accountdiscovery.ssm.amazonaws.com/AWSServiceRoleForAmazonSSM_AccountDiscovery",
        "arn:aws:iam::*:role/aws-service-role/ssm-quicksetup.amazonaws.com/
AWSServiceRoleForSSMQuickSetup",
```

```
        "arn:aws:iam::*:role/aws-service-role/resource-explorer-2.amazonaws.com/
AWSServiceRoleForResourceExplorer"
      ]
    }
  ]
}
EOF

# Create the IAM policy
POLICY_OUTPUT=$(log_cmd "aws iam create-policy --policy-name SSMOnboardingPolicy-
$UNIQUE_ID --policy-document file://ssm-onboarding-policy.json --output json")
POLICY_STATUS=$?
check_error "$POLICY_OUTPUT" $POLICY_STATUS "Failed to create IAM policy"

# Extract the policy ARN
POLICY_ARN=$(echo "$POLICY_OUTPUT" | grep -o 'arn:aws:iam::[0-9]*:policy/
SSMOnboardingPolicy-[a-f0-9]*')
if [[ -z "$POLICY_ARN" ]]; then
    echo "ERROR: Failed to extract policy ARN" | tee -a "$LOG_FILE"
    exit 1
fi

# Track the created policy
track_resource "IAM_POLICY" "$POLICY_ARN"
aws iam tag-policy --policy-arn "$POLICY_ARN" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=aws-systems-manager-gs

echo "Created policy: $POLICY_ARN" | tee -a "$LOG_FILE"

# Step 2: Create and configure IAM role for Systems Manager
echo ""
echo "Step 2: Creating IAM role for Systems Manager..."

# Get current user name
USER_OUTPUT=$(log_cmd "aws sts get-caller-identity --output json")
USER_STATUS=$?
check_error "$USER_OUTPUT" $USER_STATUS "Failed to get caller identity"

# Extract account ID
ACCOUNT_ID=$(echo "$USER_OUTPUT" | grep -o '"Account": "[0-9]*"' | cut -d'"' -f4)
if [[ -z "$ACCOUNT_ID" ]]; then
    echo "ERROR: Failed to extract account ID" | tee -a "$LOG_FILE"
    exit 1
fi
```

```
# Generate a unique role name
ROLE_NAME="SSMTutorialRole-$(openssl rand -hex 4)"

# Create trust policy for the role - FIXED: Added cloudformation.amazonaws.com
cat > trust-policy.json << 'EOF'
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ssm.amazonaws.com",
                    "ssm-quicksetup.amazonaws.com",
                    "cloudformation.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNT_ID:root"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF

# Replace ACCOUNT_ID placeholder in trust policy
sed -i "s/ACCOUNT_ID/$ACCOUNT_ID/g" trust-policy.json

# Create the IAM role
ROLE_OUTPUT=$(log_cmd "aws iam create-role --role-name $ROLE_NAME --assume-role-
policy-document file://trust-policy.json --description 'Role for Systems Manager
 tutorial' --output json")
ROLE_STATUS=$?
check_error "$ROLE_OUTPUT" $ROLE_STATUS "Failed to create IAM role"

# Extract the role ARN
ROLE_ARN=$(echo "$ROLE_OUTPUT" | grep -o 'arn:aws:iam::[0-9]*:role/[^"]*')
if [[ -z "$ROLE_ARN" ]]; then
```

```
    echo "ERROR: Failed to extract role ARN" | tee -a "$LOG_FILE"
    cleanup_on_error
    exit 1
fi

# Track the created role
track_resource "IAM_ROLE" "$ROLE_NAME"
aws iam tag-role --role-name "$ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=aws-systems-manager-gs

echo "Created IAM role: $ROLE_NAME" | tee -a "$LOG_FILE"
echo "Role ARN: $ROLE_ARN" | tee -a "$LOG_FILE"

# Set identity variables for cleanup
IDENTITY_TYPE="role"
IDENTITY_NAME="$ROLE_NAME"

# Attach the policy to the role
ATTACH_OUTPUT=$(log_cmd "aws iam attach-role-policy --role-name $ROLE_NAME --
policy-arn $POLICY_ARN")
ATTACH_STATUS=$?
check_error "$ATTACH_OUTPUT" $ATTACH_STATUS "Failed to attach policy to role
 $ROLE_NAME"

echo "Policy attached to role: $ROLE_NAME" | tee -a "$LOG_FILE"

# Step 3: Create Systems Manager configuration using Host Management
echo ""
echo "Step 3: Creating Systems Manager configuration..."

# Generate a random identifier for the configuration name
CONFIG_NAME="SSMSetup-$(openssl rand -hex 4)"

# Create configuration file for Systems Manager setup using Host Management
# Added both required parameters for single account deployment based on
 CloudFormation documentation
cat > ssm-config.json << EOF
[
  {
    "Type": "AWSQuickSetupType-SSMHostMgmt",
    "LocalDeploymentAdministrationRoleArn": "$ROLE_ARN",
    "LocalDeploymentExecutionRoleName": "$ROLE_NAME",
    "Parameters": {
      "TargetAccounts": "$ACCOUNT_ID",
```

```
      "TargetRegions": "$CURRENT_REGION"
    }
  }
]
EOF

echo "Configuration file created:" | tee -a "$LOG_FILE"
cat ssm-config.json | tee -a "$LOG_FILE"

# Create the configuration manager
CONFIG_OUTPUT=$(log_cmd "aws ssm-quicksetup create-configuration-manager --name
 \"$CONFIG_NAME\" --configuration-definitions file://ssm-config.json --region
 $CURRENT_REGION")
CONFIG_STATUS=$?
check_error "$CONFIG_OUTPUT" $CONFIG_STATUS "Failed to create Systems Manager
 configuration"

# Extract the manager ARN
MANAGER_ARN=$(echo "$CONFIG_OUTPUT" | grep -o 'arn:aws:ssm-quicksetup:[^"]*')
if [[ -z "$MANAGER_ARN" ]]; then
    echo "ERROR: Failed to extract manager ARN" | tee -a "$LOG_FILE"
    exit 1
fi

# Track the created configuration manager
track_resource "SSM_CONFIG_MANAGER" "$MANAGER_ARN"

echo "Created Systems Manager configuration: $MANAGER_ARN" | tee -a "$LOG_FILE"

# Step 4: Verify the setup
echo ""
echo "Step 4: Verifying the setup..."

# Wait for the configuration to be fully deployed
echo "Waiting for the configuration to be deployed (this may take a few
 minutes)..."
sleep 30

# Check the configuration manager status
VERIFY_OUTPUT=$(log_cmd "aws ssm-quicksetup get-configuration-manager --manager-
arn $MANAGER_ARN --region $CURRENT_REGION")
VERIFY_STATUS=$?
check_error "$VERIFY_OUTPUT" $VERIFY_STATUS "Failed to verify configuration
 manager"
```

```
echo "Systems Manager setup completed successfully!" | tee -a "$LOG_FILE"

# List the created resources
echo ""
echo "==========================================="
echo "CREATED RESOURCES"
echo "==========================================="
for resource in "${CREATED_RESOURCES[@]+"${CREATED_RESOURCES[@]}"}"; do
    echo "$resource"
done

# Prompt for cleanup
echo ""
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Do you want to clean up all created resources? (y/n): "
CLEANUP_CHOICE="y"

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    echo "Cleaning up resources..." | tee -a "$LOG_FILE"
    cleanup_resources
    echo "Cleanup completed." | tee -a "$LOG_FILE"
else
    echo "Resources will not be cleaned up. You can manually clean them up
 later." | tee -a "$LOG_FILE"
fi

echo ""
echo "Script execution completed. See $LOG_FILE for details."

# Clean up temporary files
rm -f ssm-onboarding-policy.json trust-policy.json ssm-config.json 2>/dev/null ||
 true
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateConfigurationManager
  - CreatePolicy

- CreateRole
- DeleteConfigurationManager
- DeletePolicy
- DeleteRole
- DetachRolePolicy
- GetCallerIdentity
- GetConfigurationManager

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo delle variabili di proprietà nei dashboard di monitoraggio per monitorare più funzioni serverless

L'esempio di codice seguente mostra come:

- Crea funzioni Lambda per il monitoraggio
- Crea un pannello di controllo CloudWatch
- Aggiungi una variabile di proprietà alla dashboard
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# CloudWatch Dashboard with Lambda Function Variable Script
```

```
# This script creates a CloudWatch dashboard with a property variable for Lambda
 function names

set -euo pipefail

# Security: Set restrictive umask
umask 0077

# Set up logging with secure permissions
LOG_FILE="cloudwatch-dashboard-script-v4.log"
touch "$LOG_FILE"
chmod 600 "$LOG_FILE"
echo "Starting script execution at $(date)" >> "$LOG_FILE"

# Function to log commands and their output (with sensitive data sanitization)
log_cmd() {
    local cmd="$1"
    local sanitized_cmd="${cmd//--password*/--password [REDACTED]}"
    sanitized_cmd="${sanitized_cmd//--secret*/--secret [REDACTED]}"
    echo "$(date): Running command: $sanitized_cmd" >> "$LOG_FILE"
    eval "$cmd" 2>&1 | tee -a "$LOG_FILE"
    return ${PIPESTATUS[0]}
}

# Function to check for errors in command output
check_error() {
    local cmd_output="$1"
    local cmd_status="$2"
    local error_msg="$3"

    if [ $cmd_status -ne 0 ] || echo "$cmd_output" | grep -qi "error"; then
        echo "ERROR: $error_msg" | tee -a "$LOG_FILE"
        # Sanitize output before logging
        local sanitized_output="${cmd_output//arn:aws:iam::[0-9]*/
arn:aws:iam::ACCOUNT_ID}"
        echo "Command output: $sanitized_output" | tee -a "$LOG_FILE"
        cleanup_resources
        exit 1
    fi
}

# Trap errors and cleanup
trap 'cleanup_resources' EXIT ERR INT TERM
```

```
# Function to clean up resources
cleanup_resources() {
    local exit_code=$?

    echo "" | tee -a "$LOG_FILE"
    echo "==========================================" | tee -a "$LOG_FILE"
    echo "CLEANUP PROCESS" | tee -a "$LOG_FILE"
    echo "==========================================" | tee -a "$LOG_FILE"

    if [ -n "${DASHBOARD_NAME:-}" ]; then
        echo "Deleting CloudWatch dashboard: $DASHBOARD_NAME" | tee -a
 "$LOG_FILE"
        aws cloudwatch delete-dashboards --dashboard-names "$DASHBOARD_NAME" 2>&1
 >> "$LOG_FILE" || true
    fi

    if [ -n "${LAMBDA_FUNCTION1:-}" ]; then
        echo "Deleting Lambda function: $LAMBDA_FUNCTION1" | tee -a "$LOG_FILE"
        aws lambda delete-function --function-name "$LAMBDA_FUNCTION1" 2>&1 >>
 "$LOG_FILE" || true
    fi

    if [ -n "${LAMBDA_FUNCTION2:-}" ]; then
        echo "Deleting Lambda function: $LAMBDA_FUNCTION2" | tee -a "$LOG_FILE"
        aws lambda delete-function --function-name "$LAMBDA_FUNCTION2" 2>&1 >>
 "$LOG_FILE" || true
    fi

    if [ -n "${ROLE_NAME:-}" ]; then
        echo "Detaching policy from role: $ROLE_NAME" | tee -a "$LOG_FILE"
        aws iam detach-role-policy --role-name "$ROLE_NAME" --policy-arn
 arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole 2>&1 >>
 "$LOG_FILE" || true

        echo "Deleting IAM role: $ROLE_NAME" | tee -a "$LOG_FILE"
        aws iam delete-role --role-name "$ROLE_NAME" 2>&1 >> "$LOG_FILE" || true
    fi

    # Clean up temporary files securely
    shred -vfz -n 3 trust-policy.json lambda_function.py lambda_function.zip 2>/
dev/null || rm -f trust-policy.json lambda_function.py lambda_function.zip

    echo "Cleanup completed." | tee -a "$LOG_FILE"
```

```
    return $exit_code
}

# Validate AWS CLI is installed and authenticated
if ! command -v aws &> /dev/null; then
    echo "ERROR: AWS CLI is not installed" | tee -a "$LOG_FILE"
    exit 1
fi

if ! aws sts get-caller-identity &> /dev/null; then
    echo "ERROR: AWS CLI is not properly authenticated" | tee -a "$LOG_FILE"
    exit 1
fi

# Get AWS region with validation
AWS_REGION=$(aws configure get region 2>/dev/null || echo "")
if [ -z "$AWS_REGION" ]; then
    AWS_REGION="us-east-1"
    echo "No region found in AWS config, defaulting to $AWS_REGION" | tee -a
 "$LOG_FILE"
else
    echo "Using AWS region: $AWS_REGION" | tee -a "$LOG_FILE"
fi

# Validate region format
if ! [[ "$AWS_REGION" =~ ^[a-z]{2}-[a-z]+-[0-9]$ ]]; then
    echo "ERROR: Invalid AWS region format: $AWS_REGION" | tee -a "$LOG_FILE"
    exit 1
fi

# Generate unique identifiers using secure random with validation
RANDOM_ID=$(openssl rand -hex 6)
if [ -z "$RANDOM_ID" ] || [ ${#RANDOM_ID} -ne 12 ]; then
    echo "ERROR: Failed to generate valid random identifier" | tee -a "$LOG_FILE"
    exit 1
fi

DASHBOARD_NAME="LambdaMetricsDashboard-${RANDOM_ID}"
LAMBDA_FUNCTION1="TestFunction1-${RANDOM_ID}"
LAMBDA_FUNCTION2="TestFunction2-${RANDOM_ID}"
ROLE_NAME="LambdaExecutionRole-${RANDOM_ID}"

# Validate resource names don't exceed AWS limits
```

```
if [ ${#DASHBOARD_NAME} -gt 128 ] || [ ${#LAMBDA_FUNCTION1} -gt 64 ] ||
 [ ${#ROLE_NAME} -gt 64 ]; then
    echo "ERROR: Generated resource names exceed AWS limits" | tee -a "$LOG_FILE"
    exit 1
fi

echo "Using random identifier: $RANDOM_ID" | tee -a "$LOG_FILE"
echo "Dashboard name: $DASHBOARD_NAME" | tee -a "$LOG_FILE"
echo "Lambda function names: $LAMBDA_FUNCTION1, $LAMBDA_FUNCTION2" | tee -a
 "$LOG_FILE"
echo "IAM role name: $ROLE_NAME" | tee -a "$LOG_FILE"

# Create IAM role for Lambda functions
echo "Creating IAM role for Lambda..." | tee -a "$LOG_FILE"
TRUST_POLICY='{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'

echo "$TRUST_POLICY" > trust-policy.json
chmod 600 trust-policy.json

# Validate JSON before use
if ! python3 -m json.tool trust-policy.json > /dev/null 2>&1; then
    echo "ERROR: Invalid trust policy JSON" | tee -a "$LOG_FILE"
    exit 1
fi

ROLE_OUTPUT=$(log_cmd "aws iam create-role --role-name '$ROLE_NAME' --assume-
role-policy-document file://trust-policy.json --tags Key=project,Value=doc-smith
 Key=tutorial,Value=cloudwatch-streams --output json")
check_error "$ROLE_OUTPUT" $? "Failed to create IAM role"

ROLE_ARN=$(echo "$ROLE_OUTPUT" | python3 -c "import sys, json;
 print(json.load(sys.stdin)['Role']['Arn'])" 2>/dev/null)
if [ -z "$ROLE_ARN" ]; then
```

```
    echo "ERROR: Failed to extract Role ARN" | tee -a "$LOG_FILE"
    exit 1
fi
echo "Role ARN: $ROLE_ARN" | tee -a "$LOG_FILE"

# Attach Lambda basic execution policy to the role
echo "Attaching Lambda execution policy to role..." | tee -a "$LOG_FILE"
POLICY_OUTPUT=$(log_cmd "aws iam attach-role-policy --role-name '$ROLE_NAME' --
policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole")
check_error "$POLICY_OUTPUT" $? "Failed to attach policy to role"

# Wait for role to propagate
echo "Waiting for IAM role to propagate..." | tee -a "$LOG_FILE"
sleep 10

# Create simple Python Lambda function code with security validation
echo "Creating Lambda function code..." | tee -a "$LOG_FILE"
cat > lambda_function.py << 'LAMBDA_EOF'
def handler(event, context):
    print("Lambda function executed successfully")
    return {
        'statusCode': 200,
        'body': 'Success'
    }
LAMBDA_EOF

chmod 600 lambda_function.py

# Validate Python syntax
if ! python3 -m py_compile lambda_function.py 2>/dev/null; then
    echo "ERROR: Invalid Python syntax in Lambda function" | tee -a "$LOG_FILE"
    exit 1
fi

# Zip the Lambda function code
zip -j -q lambda_function.zip lambda_function.py
if [ ! -f lambda_function.zip ]; then
    echo "ERROR: Failed to create lambda_function.zip" | tee -a "$LOG_FILE"
    exit 1
fi
chmod 600 lambda_function.zip

# Validate zip file integrity
if ! unzip -t lambda_function.zip > /dev/null 2>&1; then
```

```
    echo "ERROR: Created zip file is corrupted" | tee -a "$LOG_FILE"
    exit 1
fi

# Create first Lambda function
echo "Creating first Lambda function: $LAMBDA_FUNCTION1..." | tee -a "$LOG_FILE"
LAMBDA1_OUTPUT=$(log_cmd "aws lambda create-function --function-name
 '$LAMBDA_FUNCTION1' --runtime python3.11 --role '$ROLE_ARN' --handler
 lambda_function.handler --zip-file fileb://lambda_function.zip --timeout 30 --
memory-size 128 --tags project=doc-smith,tutorial=cloudwatch-streams")
check_error "$LAMBDA1_OUTPUT" $? "Failed to create first Lambda function"

# Create second Lambda function
echo "Creating second Lambda function: $LAMBDA_FUNCTION2..." | tee -a "$LOG_FILE"
LAMBDA2_OUTPUT=$(log_cmd "aws lambda create-function --function-name
 '$LAMBDA_FUNCTION2' --runtime python3.11 --role '$ROLE_ARN' --handler
 lambda_function.handler --zip-file fileb://lambda_function.zip --timeout 30 --
memory-size 128 --tags project=doc-smith,tutorial=cloudwatch-streams")
check_error "$LAMBDA2_OUTPUT" $? "Failed to create second Lambda function"

# Invoke Lambda functions to generate some metrics
echo "Invoking Lambda functions to generate metrics..." | tee -a "$LOG_FILE"
log_cmd "aws lambda invoke --function-name '$LAMBDA_FUNCTION1' --payload '{}' /
dev/null" || true
log_cmd "aws lambda invoke --function-name '$LAMBDA_FUNCTION2' --payload '{}' /
dev/null" || true

# Create CloudWatch dashboard with property variable
echo "Creating CloudWatch dashboard with property variable..." | tee -a
 "$LOG_FILE"

# Create dashboard body with proper escaping and validation
DASHBOARD_BODY=$(cat <<EOF
{
  "widgets": [
    {
      "type": "metric",
      "x": 0,
      "y": 0,
      "width": 12,
      "height": 6,
      "properties": {
        "metrics": [
          [ "AWS/Lambda", "Invocations", "FunctionName", "$LAMBDA_FUNCTION1" ]
```

```
        ],
        "view": "timeSeries",
        "stacked": false,
        "region": "$AWS_REGION",
        "title": "Lambda Invocations",
        "period": 300,
        "stat": "Sum"
      }
    }
  ]
}
EOF
)

# Validate JSON before sending
if ! echo "$DASHBOARD_BODY" | python3 -m json.tool > /dev/null 2>&1; then
    echo "ERROR: Dashboard body is not valid JSON" | tee -a "$LOG_FILE"
    exit 1
fi

# First create a basic dashboard without variables
echo "Creating initial dashboard without variables..." | tee -a "$LOG_FILE"
DASHBOARD_OUTPUT=$(aws cloudwatch put-dashboard --dashboard-name
 "$DASHBOARD_NAME" --dashboard-body "$DASHBOARD_BODY" --output json 2>&1)
check_error "$DASHBOARD_OUTPUT" $? "Failed to create initial CloudWatch
 dashboard"

# Now let's try to add a property variable using the console instructions
echo "To complete the tutorial, please follow these steps in the CloudWatch
 console:" | tee -a "$LOG_FILE"
echo "1. Open the CloudWatch console at https://console.aws.amazon.com/
cloudwatch/" | tee -a "$LOG_FILE"
echo "2. Navigate to Dashboards and select your dashboard: $DASHBOARD_NAME" | tee
 -a "$LOG_FILE"
echo "3. Choose Actions > Variables > Create a variable" | tee -a "$LOG_FILE"
echo "4. Choose Property variable" | tee -a "$LOG_FILE"
echo "5. For Property that the variable changes, choose FunctionName" | tee -a
 "$LOG_FILE"
echo "6. For Input type, choose Select menu (dropdown)" | tee -a "$LOG_FILE"
echo "7. Choose Use the results of a metric search" | tee -a "$LOG_FILE"
echo "8. Choose Pre-built queries > Lambda > Errors" | tee -a "$LOG_FILE"
echo "9. Choose By Function Name and then choose Search" | tee -a "$LOG_FILE"
echo "10. (Optional) Configure any secondary settings as desired" | tee -a
 "$LOG_FILE"
```

```
echo "11. Choose Add variable" | tee -a "$LOG_FILE"
echo "" | tee -a "$LOG_FILE"
echo "The dashboard has been created and can be accessed at:" | tee -a
 "$LOG_FILE"
echo "https://console.aws.amazon.com/cloudwatch/home#dashboards:name=
$DASHBOARD_NAME" | tee -a "$LOG_FILE"

# Verify dashboard creation
echo "Verifying dashboard creation..." | tee -a "$LOG_FILE"
VERIFY_OUTPUT=$(aws cloudwatch get-dashboard --dashboard-name "$DASHBOARD_NAME"
 --output json 2>&1)
check_error "$VERIFY_OUTPUT" $? "Failed to verify dashboard creation"

echo "" | tee -a "$LOG_FILE"
echo "==========================================" | tee -a "$LOG_FILE"
echo "DASHBOARD CREATED SUCCESSFULLY" | tee -a "$LOG_FILE"
echo "==========================================" | tee -a "$LOG_FILE"
echo "Dashboard Name: $DASHBOARD_NAME" | tee -a "$LOG_FILE"
echo "Lambda Functions: $LAMBDA_FUNCTION1, $LAMBDA_FUNCTION2" | tee -a
 "$LOG_FILE"
echo "" | tee -a "$LOG_FILE"
echo "You can view your dashboard in the CloudWatch console:" | tee -a
 "$LOG_FILE"
echo "https://console.aws.amazon.com/cloudwatch/home#dashboards:name=
$DASHBOARD_NAME" | tee -a "$LOG_FILE"
echo "" | tee -a "$LOG_FILE"

# Auto-confirm cleanup
echo "" | tee -a "$LOG_FILE"
echo "==========================================" | tee -a "$LOG_FILE"
echo "CLEANUP CONFIRMATION" | tee -a "$LOG_FILE"
echo "==========================================" | tee -a "$LOG_FILE"
echo "The following resources were created:" | tee -a "$LOG_FILE"
echo "- CloudWatch Dashboard: $DASHBOARD_NAME" | tee -a "$LOG_FILE"
echo "- Lambda Function: $LAMBDA_FUNCTION1" | tee -a "$LOG_FILE"
echo "- Lambda Function: $LAMBDA_FUNCTION2" | tee -a "$LOG_FILE"
echo "- IAM Role: $ROLE_NAME" | tee -a "$LOG_FILE"
echo "" | tee -a "$LOG_FILE"
echo "Auto-confirming cleanup of all created resources..." | tee -a "$LOG_FILE"

echo "Script completed successfully." | tee -a "$LOG_FILE"
exit 0
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateFunction
  - CreateRole
  - DeleteDashboards
  - DeleteFunction
  - DeleteRole
  - DetachRolePolicy
  - GetDashboard
  - Invoke
  - PutDashboard

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Lavora con DynamoDB Streams e utilizza Time-to-Live AWS Command Line Interface v2

Il seguente esempio di codice mostra come gestire i flussi e le funzionalità di DynamoDB. Time-to-Live

- Creare una tabella con i flussi abilitati.
- Descrivere i flussi.
- Creare una funzione Lambda per l'elaborazione dei flussi.
- Abilitare TTL in una tabella.
- Aggiungere elementi con attributi TTL.
- Descrivere le impostazioni TTL.

Bash

AWS CLI con lo script Bash

Creare una tabella con i flussi abilitati.

```
# Create a table with DynamoDB Streams enabled
aws dynamodb create-table \
    --table-name StreamsDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES
```

Descrivere i flussi.

```
# Get information about the stream
aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.StreamSpecification"

# Get the stream ARN
STREAM_ARN=$(aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.LatestStreamArn" \
    --output text)

echo "Stream ARN: $STREAM_ARN"

# Describe the stream
aws dynamodbstreams describe-stream \
    --stream-arn $STREAM_ARN
```

Crea una funzione Lambda per i flussi.

```
# Step 1: Create an IAM role for the Lambda function
cat > trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
```

```
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

aws iam create-role \
    --role-name DynamoDBStreamsLambdaRole \
    --assume-role-policy-document file://trust-policy.json

# Step 2: Attach permissions to the role
aws iam attach-role-policy \
    --role-name DynamoDBStreamsLambdaRole \
    --policy-arn arn:aws:iam::aws:policy/service-role/
AWSLambdaDynamoDBExecutionRole

# Step 3: Create a Lambda function (code would be in a separate file)
echo "Lambda function creation would be done separately with appropriate code"

# Step 4: Create an event source mapping
echo "Example command to create event source mapping:"
echo "aws lambda create-event-source-mapping \\"
echo "    --function-name ProcessDynamoDBRecords \\"
echo "    --event-source $STREAM_ARN \\"
echo "    --batch-size 100 \\"
echo "    --starting-position LATEST"
```

Abilitare TTL in una tabella.

```
# Create a table for TTL demonstration
aws dynamodb create-table \
    --table-name TTLDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST

# Wait for table to become active
aws dynamodb wait table-exists --table-name TTLDemo
```

```
# Enable TTL on the table
aws dynamodb update-time-to-live \
    --table-name TTLDemo \
    --time-to-live-specification "Enabled=true, AttributeName=ExpirationTime"
```

Aggiungere elementi con attributi TTL.

```
# Calculate expiration time (current time + 1 day in seconds)
EXPIRATION_TIME=$(date -d "+1 day" +%s)

# Add an item with TTL attribute
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item1"},
        "Data": {"S": "This item will expire in 1 day"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME'"}
    }'

# Add an item that expires in 1 hour
EXPIRATION_TIME_HOUR=$(date -d "+1 hour" +%s)
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item2"},
        "Data": {"S": "This item will expire in 1 hour"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME_HOUR'"}
    }'
```

Descrivere le impostazioni TTL.

```
# Describe TTL settings for a table
aws dynamodb describe-time-to-live \
    --table-name TTLDemo
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateRole

- CreateTable
- DescribeTable
- DescribeTimeToLive
- PutItem
- UpdateTimeToLive

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Lavora con l'API IAM Policy Builder utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Crea policy IAM utilizzando l'API orientata agli oggetti.
- Usa l'API IAM Policy Builder con il servizio IAM.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Gli esempi utilizzano le seguenti importazioni.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.policybuilder.iam.IamConditionOperator;
import software.amazon.awssdk.policybuilder.iam.IamEffect;
import software.amazon.awssdk.policybuilder.iam.IamPolicy;
import software.amazon.awssdk.policybuilder.iam.IamPolicyWriter;
import software.amazon.awssdk.policybuilder.iam.IamPrincipal;
import software.amazon.awssdk.policybuilder.iam.IamPrincipalType;
```

```
import software.amazon.awssdk.policybuilder.iam.IamResource;
import software.amazon.awssdk.policybuilder.iam.IamStatement;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.GetPolicyResponse;
import software.amazon.awssdk.services.iam.model.GetPolicyVersionResponse;
import software.amazon.awssdk.services.sts.StsClient;

import java.net.URLDecoder;
import java.nio.charset.StandardCharsets;
import java.util.Arrays;
import java.util.List;
```

Crea una policy basata sul tempo.

```
        public String timeBasedPolicyExample() {
                IamPolicy policy = IamPolicy.builder()
                                .addStatement(b -> b
                                                .effect(IamEffect.ALLOW)
                                                .addAction("dynamodb:GetItem")
                                                .addResource(IamResource.ALL)
                                                .addCondition(b1 -> b1

 .operator(IamConditionOperator.DATE_GREATER_THAN)

 .key("aws:CurrentTime")

 .value("2020-04-01T00:00:00Z"))
                                                .addCondition(b1 -> b1

 .operator(IamConditionOperator.DATE_LESS_THAN)

 .key("aws:CurrentTime")

 .value("2020-06-30T23:59:59Z")))
                                .build();

                // Use an IamPolicyWriter to write out the JSON string to a more
 readable
                // format.
                return policy.toJson(IamPolicyWriter.builder()
                                .prettyPrint(true)
```

```
                                .build());
        }
```

Crea una policy con più condizioni.

```
        public String multipleConditionsExample() {
                IamPolicy policy = IamPolicy.builder()
                                .addStatement(b -> b
                                                .effect(IamEffect.ALLOW)
                                                .addAction("dynamodb:GetItem")

 .addAction("dynamodb:BatchGetItem")
                                                .addAction("dynamodb:Query")
                                                .addAction("dynamodb:PutItem")
                                                .addAction("dynamodb:UpdateItem")
                                                .addAction("dynamodb:DeleteItem")

 .addAction("dynamodb:BatchWriteItem")

 .addResource("arn:aws:dynamodb:*:*:table/table-name")

 .addConditions(IamConditionOperator.STRING_EQUALS

 .addPrefix("ForAllValues:"),

 "dynamodb:Attributes",
                                                                List.of("column-
name1", "column-name2", "column-name3"))
                                                .addCondition(b1 -> b1

 .operator(IamConditionOperator.STRING_EQUALS

 .addSuffix("IfExists"))

 .key("dynamodb:Select")

 .value("SPECIFIC_ATTRIBUTES")))
                                .build();

                return policy.toJson(IamPolicyWriter.builder()
                                .prettyPrint(true).build());
        }
```

Usa i principi in una policy.

```
        public String specifyPrincipalsExample() {
                IamPolicy policy = IamPolicy.builder()
                                .addStatement(b -> b
                                                .effect(IamEffect.DENY)
                                                .addAction("s3:*")
                                                .addPrincipal(IamPrincipal.ALL)
                                                .addResource("arn:aws:s3:::amzn-
s3-demo-bucket/*")
                                                .addResource("arn:aws:s3:::amzn-
s3-demo-bucket")
                                                .addCondition(b1 -> b1

 .operator(IamConditionOperator.ARN_NOT_EQUALS)

 .key("aws:PrincipalArn")

 .value("arn:aws:iam::444455556666:user/user-name")))
                                .build();
                return policy.toJson(IamPolicyWriter.builder()
                                .prettyPrint(true).build());
        }
```

Consentire l'accesso multi-account.

```
        public String allowCrossAccountAccessExample() {
                IamPolicy policy = IamPolicy.builder()
                                .addStatement(b -> b
                                                .effect(IamEffect.ALLOW)

 .addPrincipal(IamPrincipalType.AWS, "111122223333")
                                                .addAction("s3:PutObject")
                                                .addResource("arn:aws:s3:::amzn-
s3-demo-bucket/*")
                                                .addCondition(b1 -> b1

 .operator(IamConditionOperator.STRING_EQUALS)
                                                                .key("s3:x-amz-
acl")
```

```
                                                                    .value("bucket-
owner-full-control")))
                                .build();
                return policy.toJson(IamPolicyWriter.builder()
                                .prettyPrint(true).build());
        }
```

Crea e carica un `IamPolicy`.

```
        public String createAndUploadPolicyExample(IamClient iam, String
 accountID, String policyName) {
                // Build the policy.
                IamPolicy policy = IamPolicy.builder() // 'version' defaults to
 "2012-10-17".
                                .addStatement(IamStatement.builder()
                                                .effect(IamEffect.ALLOW)
                                                .addAction("dynamodb:PutItem")

 .addResource("arn:aws:dynamodb:us-east-1:" + accountID
                                                                    + ":table/
exampleTableName")
                                                .build())
                                .build();
                // Upload the policy.
                iam.createPolicy(r ->
 r.policyName(policyName).policyDocument(policy.toJson()));
                return
 policy.toJson(IamPolicyWriter.builder().prettyPrint(true).build());
        }
```

Scarica e lavora con un `IamPolicy`.

```
        public String createNewBasedOnExistingPolicyExample(IamClient iam, String
 accountID, String policyName,
                        String newPolicyName) {

                String policyArn = "arn:aws:iam::" + accountID + ":policy/" +
 policyName;
                GetPolicyResponse getPolicyResponse = iam.getPolicy(r ->
 r.policyArn(policyArn));
```

```
                String policyVersion =
getPolicyResponse.policy().defaultVersionId();
                GetPolicyVersionResponse getPolicyVersionResponse = iam
                                .getPolicyVersion(r ->
r.policyArn(policyArn).versionId(policyVersion));

                // Create an IamPolicy instance from the JSON string returned
from IAM.
                String decodedPolicy =
URLDecoder.decode(getPolicyVersionResponse.policyVersion().document(),
                                StandardCharsets.UTF_8);
                IamPolicy policy = IamPolicy.fromJson(decodedPolicy);

                /*
                 * All IamPolicy components are immutable, so use the copy method
that creates a
                 * new instance that
                 * can be altered in the same method call.
                 *
                 * Add the ability to get an item from DynamoDB as an additional
action.
                 */
                IamStatement newStatement = policy.statements().get(0).copy(s ->
s.addAction("dynamodb:GetItem"));

                // Create a new statement that replaces the original statement.
                IamPolicy newPolicy = policy.copy(p ->
p.statements(Arrays.asList(newStatement)));

                // Upload the new policy. IAM now has both policies.
                iam.createPolicy(r -> r.policyName(newPolicyName)
                                .policyDocument(newPolicy.toJson()));

                return
newPolicy.toJson(IamPolicyWriter.builder().prettyPrint(true).build());
        }
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for Java 2.x.
- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK for Java 2.x .
  - CreatePolicy

- GetPolicy
- GetPolicyVersion

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice per AWS STS utilizzo di AWS SDK

I seguenti esempi di codice mostrano come utilizzarlo AWS STS con un kit di sviluppo AWS software (SDK).

Le azioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica chiamando più funzioni all'interno dello stesso servizio o combinate con altri Servizi AWS.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

AWS STS

- Esempi di base per AWS STS utilizzo di AWS SDK
  - Azioni per AWS STS utilizzo di AWS SDK
    - Usare AssumeRolecon un AWS SDK o CLI
    - Utilizzo di AssumeRoleWithWebIdentity con una CLI
    - Utilizzo di DecodeAuthorizationMessage con una CLI
    - Utilizzo di GetFederationToken con una CLI
    - Usare GetSessionTokencon un AWS SDK o CLI
- Scenari per AWS STS utilizzo di AWS SDK
  - Assumi un ruolo IAM che richiede un token MFA con AWS STS utilizzando un AWS SDK
  - Configurare la connettività dei servizi container
  - Costruisci un URL con AWS STS per utenti federati che utilizzano un AWS SDK

- Crea un'attività contenitore per il tipo di avvio serverless
- Crea un'API REST con integrazione proxy di funzioni
- Creazione di un servizio container per le istanze di macchine virtuali
- Creazione di un'area di lavoro di monitoraggio gestita
- Creazione di un dashboard di monitoraggio con il nome della funzione come variabile
- Ottieni un token di sessione che richiede un token MFA con AWS STS utilizzando un AWS SDK
- Guida introduttiva alla gestione della configurazione
- Guida introduttiva ai registri dei container
- Guida introduttiva alla messaggistica sull'Internet of Things
- Guida introduttiva ai feature store di machine learning
- Guida introduttiva ai cluster Kubernetes gestiti
- Guida introduttiva allo streaming gestito
- Guida introduttiva alle notifiche push
- Guida introduttiva ai motori di ricerca e analisi
- Guida introduttiva ai firewall per applicazioni Web
- Spostamento di segreti codificati per proteggere l'archiviazione segreta
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori
- Configurazione della gestione dei sistemi

Esempi di base per AWS STS utilizzo di AWS SDK

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di Servizio di token di sicurezza AWS with SDK. AWS

Esempi

- Azioni per AWS STS utilizzo di AWS SDK
  - Usare AssumeRolecon un AWS SDK o CLI
  - Utilizzo di AssumeRoleWithWebIdentity con una CLI
  - Utilizzo di DecodeAuthorizationMessage con una CLI
  - Utilizzo di GetFederationToken con una CLI
  - Usare GetSessionTokencon un AWS SDK o CLI

Azioni per AWS STS utilizzo di AWS SDK

I seguenti esempi di codice mostrano come eseguire AWS STS azioni individuali con gli AWS SDK. Ogni esempio include un collegamento a GitHub, dove è possibile trovare le istruzioni per la configurazione e l'esecuzione del codice.

Questi estratti richiamano l' AWS STS API e sono estratti di codice di programmi più grandi che devono essere eseguiti nel contesto. È possibile visualizzare le azioni nel contesto in Scenari per AWS STS utilizzo di AWS SDK.

Gli esempi seguenti includono solo le azioni più comunemente utilizzate. Per un elenco completo, consulta la documentazione di riferimento dell'API Servizio di token di sicurezza AWS.

Esempi

- Usare AssumeRolecon un AWS SDK o CLI
- Utilizzo di AssumeRoleWithWebIdentity con una CLI
- Utilizzo di DecodeAuthorizationMessage con una CLI
- Utilizzo di GetFederationToken con una CLI
- Usare GetSessionTokencon un AWS SDK o CLI

Usare **`AssumeRole`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`AssumeRole`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Assunzione un ruolo IAM che richiede un token MFA
- Formulazione di un URL per gli utenti federati

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon;
using Amazon.SecurityToken;
using Amazon.SecurityToken.Model;

namespace AssumeRoleExample
{
    class AssumeRole
    {
        /// <summary>
        /// This example shows how to use the AWS Security Token
        /// Service (AWS STS) to assume an IAM role.
        ///
        /// NOTE: It is important that the role that will be assumed has a
        /// trust relationship with the account that will assume the role.
        ///
        /// Before you run the example, you need to create the role you want to
        /// assume and have it trust the IAM account that will assume that role.
        ///
        /// See https://docs.aws.amazon.com/IAM/latest/UserGuide/
id_roles_create.html
        /// for help in working with roles.
        /// </summary>

        // A region property may be used if the profile or credentials loaded do
 not specify a region,
        // or to use a specific region.
        private static readonly RegionEndpoint REGION = RegionEndpoint.USWest2;

        static async Task Main()
        {
            // Create the SecurityToken client and then display the identity of
 the
            // default user.
            var roleArnToAssume = "arn:aws:iam::123456789012:role/
testAssumeRole";

            var client = new
 Amazon.SecurityToken.AmazonSecurityTokenServiceClient(REGION);

            // Get and display the information about the identity of the default
 user.
```

```
            var callerIdRequest = new GetCallerIdentityRequest();
            var caller = await client.GetCallerIdentityAsync(callerIdRequest);
            Console.WriteLine($"Original Caller: {caller.Arn}");

            // Create the request to use with the AssumeRoleAsync call.
            var assumeRoleReq = new AssumeRoleRequest()
            {
                DurationSeconds = 1600,
                RoleSessionName = "Session1",
                RoleArn = roleArnToAssume
            };

            var assumeRoleRes = await client.AssumeRoleAsync(assumeRoleReq);

            // Now create a new client based on the credentials of the caller
 assuming the role.
            var client2 = new AmazonSecurityTokenServiceClient(credentials:
 assumeRoleRes.Credentials, REGION);

            // Get and display information about the caller that has assumed the
 defined role.
            var caller2 = await client2.GetCallerIdentityAsync(callerIdRequest);
            Console.WriteLine($"AssumedRole Caller: {caller2.Arn}");
        }
    }
}
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
##############################################################################
# function iecho
```

```
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function sts_assume_role
#
# This function assumes a role in the AWS account and returns the temporary
#  credentials.
#
# Parameters:
#       -n role_session_name -- The name of the session.
#       -r role_arn -- The ARN of the role to assume.
#
# Returns:
#       [access_key_id, secret_access_key, session_token]
#     And:
#       0 - If successful.
#       1 - If an error occurred.
##############################################################################
function sts_assume_role() {
  local role_session_name role_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function sts_assume_role"
    echo "Assumes a role in the AWS account and returns the temporary
 credentials:"
```

```
    echo "  -n role_session_name -- The name of the session."
    echo "  -r role_arn -- The ARN of the role to assume."
    echo ""
  }

  while getopts n:r:h option; do
    case "${option}" in
      n) role_session_name=${OPTARG} ;;
      r) role_arn=${OPTARG} ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  response=$(aws sts assume-role \
    --role-session-name "$role_session_name" \
    --role-arn "$role_arn" \
    --output text \
    --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta AssumeRole AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
bool AwsDoc::STS::assumeRole(const Aws::String &roleArn,
                             const Aws::String &roleSessionName,
                             const Aws::String &externalId,
                             Aws::Auth::AWSCredentials &credentials,
                             const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::STS::STSClient sts(clientConfig);
    Aws::STS::Model::AssumeRoleRequest sts_req;

    sts_req.SetRoleArn(roleArn);
    sts_req.SetRoleSessionName(roleSessionName);
    sts_req.SetExternalId(externalId);

    const Aws::STS::Model::AssumeRoleOutcome outcome = sts.AssumeRole(sts_req);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error assuming IAM role. " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Credentials successfully retrieved." << std::endl;
        const Aws::STS::Model::AssumeRoleResult result = outcome.GetResult();
        const Aws::STS::Model::Credentials &temp_credentials =
 result.GetCredentials();

        // Store temporary credentials in return argument.
        // Note: The credentials object returned by assumeRole differs
        // from the AWSCredentials object used in most situations.
        credentials.SetAWSAccessKeyId(temp_credentials.GetAccessKeyId());
        credentials.SetAWSSecretKey(temp_credentials.GetSecretAccessKey());
        credentials.SetSessionToken(temp_credentials.GetSessionToken());
    }
```

```
    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come assumere un ruolo

Il comando `assume-role` seguente recupera un set di credenziali a breve termine per il ruolo IAM `s3-access-example`.

```
aws sts assume-role \
    --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
    --role-session-name s3-access-example
```

Output:

```
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
        "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-
access-example"
    },
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/
qwjzP2iEXAMPLEbw/m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8B
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2016-03-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    }
}
```

L'output del comando contiene una chiave di accesso, una chiave segreta e un token di sessione che puoi utilizzare per l'autenticazione in AWS.

Per l'utilizzo della AWS CLI, è possibile impostare un profilo denominato associato a un ruolo. Quando utilizzi il profilo, la AWS CLI chiamerà assume-role e gestirà le credenziali per te. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM nella CLI nella AWS CLI User Guide AWS .

- Per i dettagli sull'API, consulta AWS CLI Command AssumeRoleReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.sts.StsClient;
import software.amazon.awssdk.services.sts.model.AssumeRoleRequest;
import software.amazon.awssdk.services.sts.model.StsException;
import software.amazon.awssdk.services.sts.model.AssumeRoleResponse;
import software.amazon.awssdk.services.sts.model.Credentials;
import java.time.Instant;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;
import java.util.Locale;

/**
 * To make this code example work, create a Role that you want to assume.
 * Then define a Trust Relationship in the AWS Console. You can use this as an
 * example:
 *
 * {
 * "Version":"2012-10-17",
 * "Statement": [
 * {
 * "Effect": "Allow",
```

```
 * "Principal": {
 * "AWS": "<Specify the ARN of your IAM user you are using in this code example>"
 * },
 * "Action": "sts:AssumeRole"
 * }
 * ]
 * }
 *
 * For more information, see "Editing the Trust Relationship for an Existing
 * Role" in the AWS Directory Service guide.
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class AssumeRole {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <roleArn> <roleSessionName>\s

                Where:
                    roleArn - The Amazon Resource Name (ARN) of the role to
 assume (for example, arn:aws:iam::000008047983:role/s3role).\s
                    roleSessionName - An identifier for the assumed role session
 (for example, mysession).\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String roleArn = args[0];
        String roleSessionName = args[1];
        Region region = Region.US_EAST_1;
        StsClient stsClient = StsClient.builder()
                .region(region)
                .build();
```

```
        assumeGivenRole(stsClient, roleArn, roleSessionName);
        stsClient.close();
    }

    public static void assumeGivenRole(StsClient stsClient, String roleArn,
 String roleSessionName) {
        try {
            AssumeRoleRequest roleRequest = AssumeRoleRequest.builder()
                    .roleArn(roleArn)
                    .roleSessionName(roleSessionName)
                    .build();

            AssumeRoleResponse roleResponse = stsClient.assumeRole(roleRequest);
            Credentials myCreds = roleResponse.credentials();

            // Display the time when the temp creds expire.
            Instant exTime = myCreds.expiration();
            String tokenInfo = myCreds.sessionToken();

            // Convert the Instant to readable date.
            DateTimeFormatter formatter =
 DateTimeFormatter.ofLocalizedDateTime(FormatStyle.SHORT)
                    .withLocale(Locale.US)
                    .withZone(ZoneId.systemDefault());

            formatter.format(exTime);
            System.out.println("The token " + tokenInfo + "  expires on " +
 exTime);

        } catch (StsException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea il client.

```
import { STSClient } from "@aws-sdk/client-sts";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an AWS STS service client object.
export const client = new STSClient({ region: REGION });
```

Assumi il ruolo IAM.

```
import { AssumeRoleCommand } from "@aws-sdk/client-sts";

import { client } from "../libs/client.js";

export const main = async () => {
  try {
    // Returns a set of temporary security credentials that you can use to
    // access Amazon Web Services resources that you might not normally
    // have access to.
    const command = new AssumeRoleCommand({
      // The Amazon Resource Name (ARN) of the role to assume.
      RoleArn: "ROLE_ARN",
      // An identifier for the assumed role session.
      RoleSessionName: "session1",
      // The duration, in seconds, of the role session. The value specified
      // can range from 900 seconds (15 minutes) up to the maximum session
      // duration set for the role.
      DurationSeconds: 900,
    });
    const response = await client.send(command);
    console.log(response);
```

```
  } catch (err) {
    console.error(err);
  }
};
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
const AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

var roleToAssume = {
  RoleArn: "arn:aws:iam::123456789012:role/RoleName",
  RoleSessionName: "session1",
  DurationSeconds: 900,
};
var roleCreds;

// Create the STS service object
var sts = new AWS.STS({ apiVersion: "2011-06-15" });

//Assume Role
sts.assumeRole(roleToAssume, function (err, data) {
  if (err) console.log(err, err.stack);
  else {
    roleCreds = {
      accessKeyId: data.Credentials.AccessKeyId,
      secretAccessKey: data.Credentials.SecretAccessKey,
      sessionToken: data.Credentials.SessionToken,
    };
    stsGetCallerIdentity(roleCreds);
  }
```

```
});

//Get Arn of current identity
function stsGetCallerIdentity(creds) {
  var stsParams = { credentials: creds };
  // Create STS service object
  var sts = new AWS.STS(stsParams);

  sts.getCallerIdentity({}, function (err, data) {
    if (err) {
      console.log(err, err.stack);
    } else {
      console.log(data.Arn);
    }
  });
}
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per JavaScript API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: restituisce un set di credenziali temporanee (chiave di accesso, chiave segreta e token di sessione) che possono essere utilizzate per un'ora per accedere a AWS risorse a cui l'utente richiedente potrebbe normalmente non avere accesso. Le credenziali restituite hanno le autorizzazioni consentite dalla policy di accesso del ruolo assunto e dalla policy fornita (non è possibile utilizzare la policy fornita per concedere autorizzazioni superiori a quelle definite dalla policy di accesso del ruolo assunto).

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -Policy "...JSON policy..." -DurationInSeconds 3600
```

Esempio 2: restituisce un set di credenziali temporanee, valide per un'ora, con le stesse autorizzazioni definite nella policy di accesso del ruolo assunto.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600
```

Esempio 3: restituisce un set di credenziali temporanee che forniscono il numero di serie e il token generato da un MFA associato alle credenziali utente utilizzate per eseguire il cmdlet.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -SerialNumber "GAHT12345678" -TokenCode "123456"
```

Esempio 4: restituisce un set di credenziali temporanee che hanno assunto un ruolo definito in un account cliente. Per ogni ruolo che la terza parte può assumere, l'account cliente deve creare un ruolo utilizzando un identificatore che deve essere passato nel ExternalId parametro - ogni volta che viene assunto il ruolo.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -ExternalId "ABC123"
```

- Per i dettagli sull'API, vedere AssumeRolein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: restituisce un set di credenziali temporanee (chiave di accesso, chiave segreta e token di sessione) che possono essere utilizzate per un'ora per accedere a AWS risorse a cui l'utente richiedente potrebbe normalmente non avere accesso. Le credenziali restituite hanno le autorizzazioni consentite dalla policy di accesso del ruolo assunto e dalla policy fornita (non è possibile utilizzare la policy fornita per concedere autorizzazioni superiori a quelle definite dalla policy di accesso del ruolo assunto).

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -Policy "...JSON policy..." -DurationInSeconds 3600
```

Esempio 2: restituisce un set di credenziali temporanee, valide per un'ora, con le stesse autorizzazioni definite nella policy di accesso del ruolo assunto.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600
```

Esempio 3: restituisce un set di credenziali temporanee che forniscono il numero di serie e il token generato da un MFA associato alle credenziali utente utilizzate per eseguire il cmdlet.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -SerialNumber "GAHT12345678" -TokenCode "123456"
```

Esempio 4: restituisce un set di credenziali temporanee che hanno assunto un ruolo definito in un account cliente. Per ogni ruolo che la terza parte può assumere, l'account cliente deve creare un ruolo utilizzando un identificatore che deve essere passato nel ExternalId parametro - ogni volta che viene assunto il ruolo.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -ExternalId "ABC123"
```

- Per i dettagli sull'API, vedere AssumeRolein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Assumi un ruolo IAM che richiede un token MFA e utilizza le credenziali temporanee per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_from_assumed_role_with_mfa(
    assume_role_arn, session_name, mfa_serial_number, mfa_totp, sts_client
):
    """
    Assumes a role from another account and uses the temporary credentials from
    that role to list the Amazon S3 buckets that are owned by the other account.
    Requires an MFA device serial number and token.

    The assumed role must grant permission to list the buckets in the other
 account.

    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an ARN.
```

```
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
role.
    """
    response = sts_client.assume_role(
        RoleArn=assume_role_arn,
        RoleSessionName=session_name,
        SerialNumber=mfa_serial_number,
        TokenCode=mfa_totp,
    )
    temp_credentials = response["Credentials"]
    print(f"Assumed role {assume_role_arn} and got temporary credentials.")

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )

    print(f"Listing buckets for the assumed role's account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

- Per i dettagli sull'API, consulta AssumeRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
 # Creates an AWS Security Token Service (AWS STS) client with specified
credentials.
```

```
 # This is separated into a factory function so that it can be mocked for unit
testing.
 #
 # @param key_id [String] The ID of the access key used by the STS client.
 # @param key_secret [String] The secret part of the access key used by the STS
client.
 def create_sts_client(key_id, key_secret)
   Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
 end

 # Gets temporary credentials that can be used to assume a role.
 #
 # @param role_arn [String] The ARN of the role that is assumed when these
credentials
 #                          are used.
 # @param sts_client [AWS::STS::Client] An AWS STS client.
 # @return [Aws::AssumeRoleCredentials] The credentials that can be used to
assume the role.
 def assume_role(role_arn, sts_client)
   credentials = Aws::AssumeRoleCredentials.new(
     client: sts_client,
     role_arn: role_arn,
     role_session_name: 'create-use-assume-role-scenario'
   )
   @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
   credentials
 end
```

- Per i dettagli sull'API, consulta la AssumeRolesezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
async fn assume_role(config: &SdkConfig, role_name: String, session_name:
 Option<String>) {
    let provider = aws_config::sts::AssumeRoleProvider::builder(role_name)
        .session_name(session_name.unwrap_or("rust_sdk_example_session".into()))
        .configure(config)
        .build()
        .await;

    let local_config = aws_config::from_env()
        .credentials_provider(provider)
        .load()
        .await;
    let client = Client::new(&local_config);
    let req = client.get_caller_identity();
    let resp = req.send().await;
    match resp {
        Ok(e) => {
            println!("UserID :               {}",
 e.user_id().unwrap_or_default());
            println!("Account:               {}",
 e.account().unwrap_or_default());
            println!("Arn    :               {}", e.arn().unwrap_or_default());
        }
        Err(e) => println!("{:?}", e),
    }
}
```

- Per i dettagli sulle API, consulta la AssumeRoleguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import AWSSTS

    public func assumeRole(role: IAMClientTypes.Role, sessionName: String)
        async throws -> STSClientTypes.Credentials
    {
        let input = AssumeRoleInput(
            roleArn: role.arn,
            roleSessionName: sessionName
        )
        do {
            let output = try await stsClient.assumeRole(input: input)

            guard let credentials = output.credentials else {
                throw ServiceHandlerError.authError
            }

            return credentials
        } catch {
            print("Error assuming role: ", dump(error))
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la AssumeRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`AssumeRoleWithWebIdentity`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`AssumeRoleWithWebIdentity`.

CLI

AWS CLI

Per ottenere credenziali a breve termine per un ruolo autenticato con l'identità Web (OAuth 2."0)

Il comando `assume-role-with-web-identity` seguente recupera un set di credenziali a breve termine per il ruolo IAM `app1`. La richiesta viene autenticata utilizzando il token di identità Web fornito dal provider di identità Web specificato. Alla sessione vengono applicate due policy aggiuntive per limitare ulteriormente le azioni concesse all'utente. Le credenziali scadono un'ora dopo la loro generazione.

```
aws sts assume-role-with-web-identity \
    --duration-seconds 3600 \
    --role-session-name "app1" \
    --provider-id "www.amazon.com" \
    --policy-arns "arn:aws:iam::123456789012:policy/
q=webidentitydemopolicy1","arn:aws:iam::123456789012:policy/
webidentitydemopolicy2" \
    --role-arn arn:aws:iam::123456789012:role/FederatedWebIdentityRole \
    --web-identity-token "Atza
%7CIQEBLjAsAhRFiXuWpUXuRvQ9PZL3GMFcYevydwIUFAHZwXZXXXXXXXXJnrulxKDHwy87oGKPznh0D6bEQZTSCz
CrKqjG7nPBjNIL016GGvuS5gSvPRUxWES3VYfm1wl7WTI7jn-Pcb6M-
buCgHhFOzTQxod27L9CqnOLio7N3gZAGpsp6n1-
AJBOCJckcyXe2c6uD0srOJeZlKUm2eTDVMf8IehDVI0r1QOnTV6KzzAI3OY87Vd_cVMQ"
```

Output:

```
{
    "SubjectFromWebIdentityToken": "amzn1.account.AF6RHO7KZU5XRVQJGXK6HB56KR2A",
    "Audience": "client.5498841531868486423.1548@apps.example.com",
    "AssumedRoleUser": {
        "Arn": "arn:aws:sts::123456789012:assumed-role/FederatedWebIdentityRole/
app1",
        "AssumedRoleId": "AROACLKWSDQRAOEXAMPLE:app1"
    },
    "Credentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    },
    "Provider": "www.amazon.com"
}
```

Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza temporanee nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta AssumeRoleWithWebIdentity AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: restituisce un set temporaneo di credenziali, valido per un'ora, per un utente che è stato autenticato con il provider di identità Login with Amazon. Le credenziali presuppongono la policy di accesso associata al ruolo identificato dall'ARN del ruolo. Facoltativamente, è possibile passare una policy JSON al parametro -Policy che perfeziona ulteriormente le autorizzazioni di accesso (non è possibile concedere più autorizzazioni di quelle disponibili nelle autorizzazioni associate al ruolo). Il valore fornito a - WebIdentityToken è l'identificatore utente univoco restituito dal provider di identità.

```
Use-STSWebIdentityRole -DurationInSeconds 3600 -ProviderId "www.amazon.com"
 -RoleSessionName "app1" -RoleArn "arn:aws:iam::123456789012:role/
FederatedWebIdentityRole" -WebIdentityToken "Atza...DVI0r1"
```

- Per i dettagli sull'API, vedere AssumeRoleWithWebIdentityin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: restituisce un set temporaneo di credenziali, valido per un'ora, per un utente che è stato autenticato con il provider di identità Login with Amazon. Le credenziali presuppongono la policy di accesso associata al ruolo identificato dall'ARN del ruolo. Facoltativamente, è possibile passare una policy JSON al parametro -Policy che perfeziona ulteriormente le autorizzazioni di accesso (non è possibile concedere più autorizzazioni di quelle disponibili nelle autorizzazioni associate al ruolo). Il valore fornito a - WebIdentityToken è l'identificatore utente univoco restituito dal provider di identità.

```
Use-STSWebIdentityRole -DurationInSeconds 3600 -ProviderId "www.amazon.com"
 -RoleSessionName "app1" -RoleArn "arn:aws:iam::123456789012:role/
FederatedWebIdentityRole" -WebIdentityToken "Atza...DVI0r1"
```

- Per i dettagli sull'API, vedere AssumeRoleWithWebIdentityin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`DecodeAuthorizationMessage`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`DecodeAuthorizationMessage`.

CLI

AWS CLI

Per decodificare un messaggio di autorizzazione codificato restituito in risposta a una richiesta

Il seguente esempio `decode-authorization-message` decodifica informazioni aggiuntive sullo stato di autorizzazione di una richiesta da un messaggio codificato restituito in risposta a una richiesta Amazon Web Services.

```
aws sts decode-authorization-message \
    --encoded-message EXAMPLEWodyRNrtlQARDip-
eTA6i6DrlUhHhPQrLWB_lAbl5pAKxl9mPDLexYcGBreyIKQC1BGBIpBKr3dFDkwqeO7e2NMk5j_hmzAiChJN-8oy3
Ojau7BMjOTWwOtHPhV_Zaz87yENdipr745EjQwRd5LaoL3vN8_5ZfA9UiBMKDgVh1gjqZJFUiQoubv78V1RbHNYnK
pOu3FZjwYStfvTb3GHs3-6rLribGO9jZOktkfE6vqxlFzLyeDr4P2ihC1wty9tArCvvGzIAUNmARQJ2VVWPxioqgo
JWP5pwe_mAyqh0NLw-r1S56YC_90onj9A80sNrHlI-
tIiNd7tgNTYzDuPQYD2FMDBnp82V9eVmYGtPp5NIeSpuf3fOHanFuBZgENxZQZ2dlH3xJGMTtYayzZrRXjiq_SfX9
FaoPIb8LmmKVBLpIB0iFhU9sEHPqKHVPi6jdxXqKaZaFGvYVmVOiuQdNQKuyk0p067POFrZECLjjOtNPBOZCcuEKE
```

Output:

```
{
    "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":true,
\"matchedStatements\":{\"items\":[{\"statementId\":\"VisualEditor0\",\"effect
\":\"DENY\",\"principals\":{\"items\":[{\"value\":\"AROA123456789EXAMPLE
\"}]},\"principalGroups\":{\"items\":[]},\"actions\":{\"items\":[{\"value
\":\"ec2:RunInstances\"}]},\"resources\":{\"items\":[{\"value\":\"*
\"}]},\"conditions\":{\"items\":[]}}]},\"failures\":{\"items\":[]},
\"context\":{\"principal\":{\"id\":\"AROA123456789EXAMPLE:Ana\",\"arn
\":\"arn:aws:sts::111122223333:assumed-role/Developer/Ana\"},\"action\":
\"RunInstances\",\"resource\":\"arn:aws:ec2:us-east-1:111122223333:instance/*
\",\"conditions\":{\"items\":[{\"key\":\"ec2:MetadataHttpPutResponseHopLimit\",
\"values\":{\"items\":[{\"value\":\"2\"}]}},{\"key\":\"ec2:InstanceMarketType
\",\"values\":{\"items\":[{\"value\":\"on-demand\"}]}},{\"key\":\"aws:Resource
\",\"values\":{\"items\":[{\"value\":\"instance/*\"}]}},{\"key\":\"aws:Account
```

```
\",\"values\":{\"items\":[{\"value\":\"111122223333\"}]}},{\"key\":
\"ec2:AvailabilityZone\",\"values\":{\"items\":[{\"value\":\"us-east-1f\"}]}},
{\"key\":\"ec2:ebsOptimized\",\"values\":{\"items\":[{\"value\":\"false\"}]}},
{\"key\":\"ec2:IsLaunchTemplateResource\",\"values\":{\"items\":[{\"value\":
\"false\"}]}},{\"key\":\"ec2:InstanceType\",\"values\":{\"items\":[{\"value
\":\"t2.micro\"}]}},{\"key\":\"ec2:RootDeviceType\",\"values\":{\"items\":
[{\"value\":\"ebs\"}]}},{\"key\":\"aws:Region\",\"values\":{\"items\":[{\"value
\":\"us-east-1\"}]}},{\"key\":\"ec2:MetadataHttpEndpoint\",\"values\":{\"items
\":[{\"value\":\"enabled\"}]}},{\"key\":\"aws:Service\",\"values\":{\"items
\":[{\"value\":\"ec2\"}]}},{\"key\":\"ec2:InstanceID\",\"values\":{\"items\":
[{\"value\":\"*\"}]}},{\"key\":\"ec2:MetadataHttpTokens\",\"values\":{\"items
\":[{\"value\":\"required\"}]}},{\"key\":\"aws:Type\",\"values\":{\"items
\":[{\"value\":\"instance\"}]}},{\"key\":\"ec2:Tenancy\",\"values\":{\"items
\":[{\"value\":\"default\"}]}},{\"key\":\"ec2:Region\",\"values\":{\"items
\":[{\"value\":\"us-east-1\"}]}},{\"key\":\"aws:ARN\",\"values\":{\"items\":
[{\"value\":\"arn:aws:ec2:us-east-1:111122223333:instance/*\"}]}}]}}"
}
```

Per ulteriori informazioni, consulta Logica di valutazione delle policy nella Guida per l'utente di AWS IAM.

- Per i dettagli sull'API, consulta DecodeAuthorizationMessage AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: decodifica le informazioni aggiuntive contenute nel contenuto del messaggio codificato fornito e restituito in risposta a una richiesta. Le informazioni aggiuntive sono codificate perché i dettagli dello stato di autorizzazione possono costituire informazioni con privilegi che l'utente che ha richiesto l'azione non dovrebbe vedere.

```
Convert-STSAuthorizationMessage -EncodedMessage "...encoded message..."
```

- Per i dettagli sull'API, vedere DecodeAuthorizationMessagein AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: decodifica le informazioni aggiuntive contenute nel contenuto del messaggio codificato fornito e restituito in risposta a una richiesta. Le informazioni aggiuntive sono

codificate perché i dettagli dello stato di autorizzazione possono costituire informazioni con privilegi che l'utente che ha richiesto l'azione non dovrebbe vedere.

```
Convert-STSAuthorizationMessage -EncodedMessage "...encoded message..."
```

- Per i dettagli sull'API, vedere DecodeAuthorizationMessagein AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo di **`GetFederationToken`** con una CLI

I seguenti esempi di codice mostrano come utilizzare`GetFederationToken`.

CLI

AWS CLI

Per restituire un set di credenziali di sicurezza temporanee utilizzando le credenziali della chiave di accesso utente IAM

Il seguente esempio `get-federation-token` restituisce un set di credenziali di sicurezza temporanee (ovvero l'ID chiave di accesso, una chiave di accesso segreta e un token di sicurezza) per un utente. Devi chiamare l'operazione `GetFederationToken` tramite le credenziali di sicurezza a lungo termine di un utente IAM.

```
aws sts get-federation-token \
    --name Bob \
    --policy file://myfile.json \
    --policy-arns arn=arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \
    --duration-seconds 900
```

Contenuto di `myfile.json`:

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "elasticloadbalancing:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:Describe*",
            "Resource": "*"
        }
    ]
}
```

Output:

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "EXAMPLEpZ2luX2VjEGoaCXVzLXdlc3QtMiJIMEYCIQC/
W9pL5ArQyDD5JwFL3/h5+WGopQ24GEXweNctwhi9sgIhAMkg
+MZE35iWM8s4r5Lr25f9rSTVPFH98G42QQunWMTfKq0DCOP/////////
wEQAxoMNDUyOTI1MTcwNTA3Igxuy3AOpuuoLsk3MJwqgQPg8QOd9HuoClUxq26wnc/nm
+eZLjHDyGf2KUAHK2DuaS/nrGSEXAMPLE",
        "Expiration": "2023-12-20T02:06:07+00:00"
    },
    "FederatedUser": {
        "FederatedUserId": "111122223333:Bob",
        "Arn": "arn:aws:sts::111122223333:federated-user/Bob"
    },
```

```
    "PackedPolicySize": 36
}
```

Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza temporanee nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta GetFederationToken AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: richiede un token federato valido per un'ora utilizzando "Bob" come nome dell'utente federato. Questo nome può essere usato per fare riferimento al nome utente federato in una policy basata sulle risorse (ad esempio una policy di bucket di Amazon S3). La policy IAM fornita, in formato JSON, viene utilizzata per definire le autorizzazioni disponibili per l'utente IAM. La policy fornita non può concedere più autorizzazioni di quelle concesse all'utente richiedente, e le autorizzazioni finali per l'utente federato sono il set più restrittivo in base all'intersezione tra la policy passata e la policy utente IAM.

```
Get-STSFederationToken -Name "Bob" -Policy "...JSON policy..." -DurationInSeconds
 3600
```

- Per i dettagli sull'API, vedere GetFederationTokenin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: richiede un token federato valido per un'ora utilizzando "Bob" come nome dell'utente federato. Questo nome può essere usato per fare riferimento al nome utente federato in una policy basata sulle risorse (ad esempio una policy di bucket di Amazon S3). La policy IAM fornita, in formato JSON, viene utilizzata per definire le autorizzazioni disponibili per l'utente IAM. La policy fornita non può concedere più autorizzazioni di quelle concesse all'utente richiedente, e le autorizzazioni finali per l'utente federato sono il set più restrittivo in base all'intersezione tra la policy passata e la policy utente IAM.

```
Get-STSFederationToken -Name "Bob" -Policy "...JSON policy..." -DurationInSeconds
 3600
```

- Per i dettagli sull'API, vedere GetFederationTokenin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, vedere. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usare **`GetSessionToken`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetSessionToken`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Recupero di un token di sessione che richiede un token MFA

CLI

AWS CLI

Come ottenere un set di credenziali a breve termine per un'identità IAM

il comando `get-session-token` seguente recupera un set di credenziali a breve termine per l'identità IAM che esegue la chiamata. Le credenziali risultanti possono essere utilizzate per richieste in cui l'autenticazione a più fattori (MFA) è richiesta dalla policy. Le credenziali scadono 15 minuti dopo la loro generazione.

```
aws sts get-session-token \
    --duration-seconds 900 \
    --serial-number "YourMFADeviceSerialNumber" \
    --token-code 123456
```

Output:

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
```

```
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    }
}
```

Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza temporanee nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta GetSessionToken AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: restituisce un'istanza **Amazon.RuntimeAWSCredentials** contenente credenziali temporanee valide per un determinato periodo di tempo. Le credenziali utilizzate per richiedere credenziali temporanee vengono dedotte dalle impostazioni predefinite correnti della shell. Per specificare altre credenziali, utilizzare i parametri - ProfileName o - AccessKey SecretKey /.

```
Get-STSSessionToken
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 2: restituisce un'istanza **Amazon.RuntimeAWSCredentials** contenente credenziali temporanee valide per un'ora. Le credenziali utilizzate per effettuare la richiesta vengono ottenute dal profilo specificato.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
```

```
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 3: restituisce un'istanza **`Amazon.RuntimeAWSCredentials`** contenente credenziali temporanee valide per un'ora utilizzando il numero di identificazione del dispositivo MFA associato all'account le cui credenziali sono specificate nel profilo 'myprofilename' e il valore fornito dal dispositivo.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile -SerialNumber
 YourMFADeviceSerialNumber -TokenCode 123456
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

- Per i dettagli sull'API, vedere GetSessionTokenin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: restituisce un'istanza **`Amazon.RuntimeAWSCredentials`** contenente credenziali temporanee valide per un determinato periodo di tempo. Le credenziali utilizzate per richiedere credenziali temporanee vengono dedotte dalle impostazioni predefinite correnti della shell. Per specificare altre credenziali, utilizzare i parametri - ProfileName o - AccessKey SecretKey /.

```
Get-STSSessionToken
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
```

```
EXAMPLEACCESSKEYID                      2/16/2015 9:12:28 PM
 examplesecretaccesskey...              SamPleTokeN.....
```

Esempio 2: restituisce un'istanza **`Amazon.RuntimeAWSCredentials`** contenente credenziali temporanee valide per un'ora. Le credenziali utilizzate per effettuare la richiesta vengono ottenute dal profilo specificato.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile
```

Output:

```
AccessKeyId                             Expiration
 SecretAccessKey                        SessionToken
-----------                             ----------
 ---------------                        ------------
EXAMPLEACCESSKEYID                      2/16/2015 9:12:28 PM
 examplesecretaccesskey...              SamPleTokeN.....
```

Esempio 3: restituisce un'istanza **`Amazon.RuntimeAWSCredentials`** contenente credenziali temporanee valide per un'ora utilizzando il numero di identificazione del dispositivo MFA associato all'account le cui credenziali sono specificate nel profilo 'myprofilename' e il valore fornito dal dispositivo.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile -SerialNumber
 YourMFADeviceSerialNumber -TokenCode 123456
```

Output:

```
AccessKeyId                             Expiration
 SecretAccessKey                        SessionToken
-----------                             ----------
 ---------------                        ------------
EXAMPLEACCESSKEYID                      2/16/2015 9:12:28 PM
 examplesecretaccesskey...              SamPleTokeN.....
```

- Per i dettagli sull'API, vedere GetSessionTokenin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Recupera un token di sessione passando un token MFA e utilizzalo per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_with_session_token_with_mfa(mfa_serial_number, mfa_totp,
 sts_client):
    """
    Gets a session token with MFA credentials and uses the temporary session
    credentials to list Amazon S3 buckets.

    Requires an MFA device serial number and token.

    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an Amazon Resource Name (ARN).
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
 role.
    """
    if mfa_serial_number is not None:
        response = sts_client.get_session_token(
            SerialNumber=mfa_serial_number, TokenCode=mfa_totp
        )
    else:
        response = sts_client.get_session_token()
    temp_credentials = response["Credentials"]

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )
```

```
    print(f"Buckets for the account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

- Per i dettagli sull'API, consulta GetSessionToken AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per AWS STS utilizzo di AWS SDK

I seguenti esempi di codice mostrano come implementare scenari comuni AWS STS con gli AWS SDK. Questi scenari mostrano come eseguire attività specifiche richiamando più funzioni all'interno AWS STS o combinandole con altre. Servizi AWS Ogni scenario include un collegamento al codice sorgente completo, dove è possibile trovare le istruzioni su come configurare ed eseguire il codice.

Gli scenari sono relativi a un livello intermedio di esperienza per aiutarti a comprendere le azioni di servizio nel contesto.

Esempi

- Assumi un ruolo IAM che richiede un token MFA con AWS STS utilizzando un AWS SDK
- Configurare la connettività dei servizi container
- Costruisci un URL con AWS STS per utenti federati che utilizzano un AWS SDK
- Crea un'attività contenitore per il tipo di avvio serverless
- Crea un'API REST con integrazione proxy di funzioni
- Creazione di un servizio container per le istanze di macchine virtuali
- Creazione di un'area di lavoro di monitoraggio gestita
- Creazione di un dashboard di monitoraggio con il nome della funzione come variabile
- Ottieni un token di sessione che richiede un token MFA con AWS STS utilizzando un AWS SDK
- Guida introduttiva alla gestione della configurazione

- Guida introduttiva ai registri dei container
- Guida introduttiva alla messaggistica sull'Internet of Things
- Guida introduttiva ai feature store di machine learning
- Guida introduttiva ai cluster Kubernetes gestiti
- Guida introduttiva allo streaming gestito
- Guida introduttiva alle notifiche push
- Guida introduttiva ai motori di ricerca e analisi
- Guida introduttiva ai firewall per applicazioni Web
- Spostamento di segreti codificati per proteggere l'archiviazione segreta
- Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori
- Configurazione della gestione dei sistemi

Assumi un ruolo IAM che richiede un token MFA con AWS STS utilizzando un AWS SDK

L'esempio di codice seguente mostra come assumere un ruolo che richiede un token MFA.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come Centro identità AWS IAM.

- Creare un ruolo IAM che conceda l'autorizzazione per elencare i bucket Amazon S3.
- Creare un utente IAM che abbia il permesso di assumere il ruolo solo quando vengono fornite le credenziali MFA.
- Registrare un dispositivo MFA per l'utente.
- Assumere il ruolo ed elencare i bucket Amazon S3 utilizzando le credenziali temporanee.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Creare un utente IAM, registrare un dispositivo MFA e creare un ruolo che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo.

```
def setup(iam_resource):
    """
    Creates a new user with no permissions.
    Creates a new virtual MFA device.
    Displays the QR code to seed the device.
    Asks for two codes from the MFA device.
    Registers the MFA device for the user.
    Creates an access key pair for the user.
    Creates a role with a policy that lets the user assume the role and requires
 MFA.
    Creates a policy that allows listing Amazon S3 buckets.
    Attaches the policy to the role.
    Creates an inline policy for the user that lets the user assume the role.

    For demonstration purposes, the user is created in the same account as the
 role,
    but in practice the user would likely be from another account.

    Any MFA device that can scan a QR code will work with this demonstration.
    Common choices are mobile apps like LastPass Authenticator,
    Microsoft Authenticator, or Google Authenticator.

    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
 resource
                         that has permissions to create users, roles, and
 policies
                         in the account.
    :return: The newly created user, user key, virtual MFA device, and role.
```

```
    """
    user = iam_resource.create_user(UserName=unique_name("user"))
    print(f"Created user {user.name}.")

    virtual_mfa_device = iam_resource.create_virtual_mfa_device(
        VirtualMFADeviceName=unique_name("mfa")
    )
    print(f"Created virtual MFA device {virtual_mfa_device.serial_number}")

    print(
        f"Showing the QR code for the device. Scan this in the MFA app of your "
        f"choice."
    )
    with open("qr.png", "wb") as qr_file:
        qr_file.write(virtual_mfa_device.qr_code_png)
    webbrowser.open(qr_file.name)

    print(f"Enter two consecutive code from your MFA device.")
    mfa_code_1 = input("Enter the first code: ")
    mfa_code_2 = input("Enter the second code: ")
    user.enable_mfa(
        SerialNumber=virtual_mfa_device.serial_number,
        AuthenticationCode1=mfa_code_1,
        AuthenticationCode2=mfa_code_2,
    )
    os.remove(qr_file.name)
    print(f"MFA device is registered with the user.")

    user_key = user.create_access_key_pair()
    print(f"Created access key pair for user.")

    print(f"Wait for user to be ready.", end="")
    progress_bar(10)

    role = iam_resource.create_role(
        RoleName=unique_name("role"),
        AssumeRolePolicyDocument=json.dumps(
            {
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {"AWS": user.arn},
                        "Action": "sts:AssumeRole",
```

```
                        "Condition": {"Bool": {"aws:MultiFactorAuthPresent":
True}},
                    }
                ],
            }
        ),
    )
    print(f"Created role {role.name} that requires MFA.")

    policy = iam_resource.create_policy(
        PolicyName=unique_name("policy"),
        PolicyDocument=json.dumps(
            {
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": "s3:ListAllMyBuckets",
                        "Resource": "arn:aws:s3:::*",
                    }
                ],
            }
        ),
    )
    role.attach_policy(PolicyArn=policy.arn)
    print(f"Created policy {policy.policy_name} and attached it to the role.")

    user.create_policy(
        PolicyName=unique_name("user-policy"),
        PolicyDocument=json.dumps(
            {
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": "sts:AssumeRole",
                        "Resource": role.arn,
                    }
                ],
            }
        ),
    )
    print(
        f"Created an inline policy for {user.name} that lets the user assume "
```

```
        f"the role."
    )

    print("Give AWS time to propagate these new resources and connections.",
 end="")
    progress_bar(10)

    return user, user_key, virtual_mfa_device, role
```

Dimostra che non è consentito assumere il ruolo senza un token MFA.

```
def try_to_assume_role_without_mfa(assume_role_arn, session_name, sts_client):
    """
    Shows that attempting to assume the role without sending MFA credentials
 results
    in an AccessDenied error.

    :param assume_role_arn: The Amazon Resource Name (ARN) of the role to assume.
    :param session_name: The name of the STS session.
    :param sts_client: A Boto3 STS instance that has permission to assume the
 role.
    """
    print(f"Trying to assume the role without sending MFA credentials...")
    try:
        sts_client.assume_role(RoleArn=assume_role_arn,
 RoleSessionName=session_name)
        raise RuntimeError("Expected AccessDenied error.")
    except ClientError as error:
        if error.response["Error"]["Code"] == "AccessDenied":
            print("Got AccessDenied.")
        else:
            raise
```

Assumere il ruolo che concede l'autorizzazione per elencare i bucket Amazon S3, passando il token MFA richiesto e mostrare che i bucket possono essere elencati.

```
def list_buckets_from_assumed_role_with_mfa(
```

```
    assume_role_arn, session_name, mfa_serial_number, mfa_totp, sts_client
):
    """
    Assumes a role from another account and uses the temporary credentials from
    that role to list the Amazon S3 buckets that are owned by the other account.
    Requires an MFA device serial number and token.

    The assumed role must grant permission to list the buckets in the other
 account.

    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an ARN.
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
 role.
    """
    response = sts_client.assume_role(
        RoleArn=assume_role_arn,
        RoleSessionName=session_name,
        SerialNumber=mfa_serial_number,
        TokenCode=mfa_totp,
    )
    temp_credentials = response["Credentials"]
    print(f"Assumed role {assume_role_arn} and got temporary credentials.")

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )

    print(f"Listing buckets for the assumed role's account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

Elimina le risorse create per la demo.

```
def teardown(user, virtual_mfa_device, role):
    """
    Removes all resources created during setup.

    :param user: The demo user.
    :param role: The demo role.
    """
    for attached in role.attached_policies.all():
        policy_name = attached.policy_name
        role.detach_policy(PolicyArn=attached.arn)
        attached.delete()
        print(f"Detached and deleted {policy_name}.")
    role.delete()
    print(f"Deleted {role.name}.")
    for user_pol in user.policies.all():
        user_pol.delete()
        print("Deleted inline user policy.")
    for key in user.access_keys.all():
        key.delete()
        print("Deleted user's access key.")
    for mfa in user.mfa_devices.all():
        mfa.disassociate()
    virtual_mfa_device.delete()
    user.delete()
    print(f"Deleted {user.name}.")
```

Esegui questo scenario utilizzando le funzioni definite in precedenza.

```
def usage_demo():
    """Drives the demonstration."""
    print("-" * 88)
    print(
        f"Welcome to the AWS Security Token Service assume role demo, "
        f"starring multi-factor authentication (MFA)!"
    )
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    user, user_key, virtual_mfa_device, role = setup(iam_resource)
```

```
    print(f"Created {user.name} and {role.name}.")
    try:
        sts_client = boto3.client(
            "sts", aws_access_key_id=user_key.id,
aws_secret_access_key=user_key.secret
        )
        try_to_assume_role_without_mfa(role.arn, "demo-sts-session", sts_client)
        mfa_totp = input("Enter the code from your registered MFA device: ")
        list_buckets_from_assumed_role_with_mfa(
            role.arn,
            "demo-sts-session",
            virtual_mfa_device.serial_number,
            mfa_totp,
            sts_client,
        )
    finally:
        teardown(user, virtual_mfa_device, role)
        print("Thanks for watching!")
```

- Per i dettagli sull'API, consulta AssumeRole AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Configurare la connettività dei servizi container

L'esempio di codice seguente mostra come:

- Crea l'infrastruttura VPC
- Configurare la registrazione
- Crea il cluster ECS
- Configura i ruoli IAM
- Crea il servizio con Service Connect
- Verifica della distribuzione
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# ECS Service Connect Tutorial Script v4 - Modified to use Default VPC
# This script creates an ECS cluster with Service Connect and deploys an nginx
 service
# Uses the default VPC to avoid VPC limits

set -e  # Exit on any error

# Configuration
SCRIPT_NAME="ECS Service Connect Tutorial"
LOG_FILE="ecs-service-connect-tutorial-v4-default-vpc.log"
REGION=${AWS_DEFAULT_REGION:-${AWS_REGION:-$(aws configure get region 2>/dev/
null)}}
if [ -z "$REGION" ]; then
    echo "ERROR: No AWS region configured."
    echo "Set one with: aws configure set region us-east-1"
    exit 1
fi
ENV_PREFIX="tutorial"
CLUSTER_NAME="${ENV_PREFIX}-cluster"
NAMESPACE_NAME="service-connect"

# Generate random suffix for unique resource names
RANDOM_SUFFIX=$(openssl rand -hex 6)

# Arrays to track created resources for cleanup
declare -a CREATED_RESOURCES=()

# Logging function
log() {
```

```
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | tee -a "$LOG_FILE"
}

# Error handling function
handle_error() {
    log "ERROR: Script failed at line $1"
    log "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Set up error handling
trap 'handle_error $LINENO' ERR

# Function to add resource to tracking array
track_resource() {
    CREATED_RESOURCES+=("$1")
    log "Tracking resource: $1"
}

# Function to check if command output contains actual errors
check_for_errors() {
    local output="$1"
    local command_name="$2"

    # Check for specific AWS CLI error patterns, not just any occurrence of
 "error"
    if echo "$output" | grep -qi "An error occurred\|InvalidParameterException\|
AccessDenied\|ValidationException\|ResourceNotFoundException"; then
        log "ERROR in $command_name: $output"
        return 1
    fi
    return 0
}

# Function to get AWS account ID
get_account_id() {
    ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
    log "Using AWS Account ID: $ACCOUNT_ID"
}

# Function to wait for resources to be ready
wait_for_resource() {
    local resource_type="$1"
```

```
    local resource_id="$2"

    case "$resource_type" in
        "cluster")
            log "Waiting for cluster $resource_id to be active..."
            local attempt=1
            local max_attempts=30
            while [ $attempt -le $max_attempts ]; do
                local status=$(aws ecs describe-clusters --clusters
 "$resource_id" --query 'clusters[0].status' --output text)
                if [ "$status" = "ACTIVE" ]; then
                    log "Cluster is now active"
                    return 0
                fi
                log "Cluster status: $status (attempt $attempt/$max_attempts)"
                sleep 10
                ((attempt++))
            done
            log "ERROR: Cluster did not become active within expected time"
            return 1
            ;;
        "service")
            log "Waiting for service $resource_id to be stable..."
            aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$resource_id"
            ;;
        "nat-gateway")
            log "Waiting for NAT Gateway $resource_id to be available..."
            aws ec2 wait nat-gateway-available --nat-gateway-ids "$resource_id"
            ;;
    esac
}

# Function to use default VPC infrastructure
setup_default_vpc_infrastructure() {
    log "Using default VPC infrastructure..."

    # Get default VPC
    VPC_ID=$(aws ec2 describe-vpcs --filters "Name=isDefault,Values=true" --query
 'Vpcs[0].VpcId' --output text)
    if [[ "$VPC_ID" == "None" || -z "$VPC_ID" ]]; then
        log "ERROR: No default VPC found. Please create a default VPC first."
        exit 1
    fi
```

```
    log "Using default VPC: $VPC_ID"

    # Get default subnets
    SUBNETS=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC_ID"
 "Name=default-for-az,Values=true" --query 'Subnets[].SubnetId' --output text)
    SUBNET_ARRAY=($SUBNETS)

    if [ ${#SUBNET_ARRAY[@]} -lt 2 ]; then
        log "ERROR: Need at least 2 subnets for ECS Service Connect. Found:
 ${#SUBNET_ARRAY[@]}"
        exit 1
    fi

    PUBLIC_SUBNET1=${SUBNET_ARRAY[0]}
    PUBLIC_SUBNET2=${SUBNET_ARRAY[1]}

    log "Using subnets: $PUBLIC_SUBNET1, $PUBLIC_SUBNET2"

    # Create security group for ECS tasks
    SG_OUTPUT=$(aws ec2 create-security-group \
        --group-name "${ENV_PREFIX}-ecs-sg-${RANDOM_SUFFIX}" \
        --description "Security group for ECS Service Connect tutorial" \
        --vpc-id "$VPC_ID" \
        --tag-specifications 'ResourceType=security-
group,Tags=[{Key=project,Value=doc-smith},{Key=tutorial,Value=amazon-ecs-service-
connect}]' 2>&1)
    check_for_errors "$SG_OUTPUT" "create-security-group"
    SECURITY_GROUP_ID=$(echo "$SG_OUTPUT" | grep -o '"GroupId": "[^"]*"' | cut -
d'"' -f4)
    track_resource "SG:$SECURITY_GROUP_ID"
    log "Created security group: $SECURITY_GROUP_ID"

    # Add inbound rules to security group
    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 80 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true

    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 443 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true
```

```
    log "Default VPC infrastructure setup completed"
}

# Function to create CloudWatch log groups
create_log_groups() {
    log "Creating CloudWatch log groups..."

    # Create log group for nginx container
    aws logs create-log-group --log-group-name "/ecs/service-connect-nginx"
 --tags project=doc-smith,tutorial=amazon-ecs-service-connect 2>&1 | grep -v
 "ResourceAlreadyExistsException" || {
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-nginx created"
            track_resource "LOG_GROUP:/ecs/service-connect-nginx"
        else
            log "Log group /ecs/service-connect-nginx already exists"
        fi
    }

    # Create log group for service connect proxy
    aws logs create-log-group --log-group-name "/ecs/service-connect-proxy"
 --tags project=doc-smith,tutorial=amazon-ecs-service-connect 2>&1 | grep -v
 "ResourceAlreadyExistsException" || {
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-proxy created"
            track_resource "LOG_GROUP:/ecs/service-connect-proxy"
        else
            log "Log group /ecs/service-connect-proxy already exists"
        fi
    }
}

# Function to create ECS cluster with Service Connect
create_ecs_cluster() {
    log "Creating ECS cluster with Service Connect..."

    CLUSTER_OUTPUT=$(aws ecs create-cluster \
        --cluster-name "$CLUSTER_NAME" \
        --service-connect-defaults namespace="$NAMESPACE_NAME" \
        --tags key=Environment,value=tutorial key=project,value=doc-smith
 key=tutorial,value=amazon-ecs-service-connect 2>&1)
    check_for_errors "$CLUSTER_OUTPUT" "create-cluster"
```

```
    track_resource "CLUSTER:$CLUSTER_NAME"
    log "Created ECS cluster: $CLUSTER_NAME"

    wait_for_resource "cluster" "$CLUSTER_NAME"

    # Track the Service Connect namespace that gets created
    # Wait a moment for the namespace to be created
    sleep 5
    NAMESPACE_ID=$(aws servicediscovery list-namespaces \
        --filters Name=TYPE,Values=HTTP \
        --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")

    if [[ -n "$NAMESPACE_ID" && "$NAMESPACE_ID" != "None" ]]; then
        track_resource "NAMESPACE:$NAMESPACE_ID"
        log "Service Connect namespace created: $NAMESPACE_ID"
    fi
}

# Function to create IAM roles
create_iam_roles() {
    log "Creating IAM roles..."

    # Check if ecsTaskExecutionRole exists
    if aws iam get-role --role-name ecsTaskExecutionRole >/dev/null 2>&1; then
        log "IAM role ecsTaskExecutionRole exists"
    else
        log "Creating ecsTaskExecutionRole..."
        aws iam create-role \
            --role-name ecsTaskExecutionRole \
            --assume-role-policy-document '{
                "Version":"2012-10-17",
                "Statement": [{
                    "Effect": "Allow",
                    "Principal": {"Service": "ecs-tasks.amazonaws.com"},
                    "Action": "sts:AssumeRole"
                }]
            }' >/dev/null 2>&1
        aws iam attach-role-policy \
            --role-name ecsTaskExecutionRole \
            --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy >/dev/null 2>&1
        track_resource "ROLE:ecsTaskExecutionRole"
```

```
        aws iam tag-role --role-name ecsTaskExecutionRole --tags
 Key=project,Value=doc-smith Key=tutorial,Value=amazon-ecs-service-connect
        log "Created ecsTaskExecutionRole"
        sleep 10
    fi

    # Check if ecsTaskRole exists, create if not
    if aws iam get-role --role-name ecsTaskRole >/dev/null 2>&1; then
        log "IAM role ecsTaskRole exists"
    else
        log "IAM role ecsTaskRole does not exist, will create it"

        # Create trust policy for ECS tasks
        cat > /tmp/ecs-task-trust-policy.json << EOF
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF

        aws iam create-role \
            --role-name ecsTaskRole \
            --assume-role-policy-document file:///tmp/ecs-task-trust-policy.json
 >/dev/null

        track_resource "IAM_ROLE:ecsTaskRole"
        aws iam tag-role --role-name ecsTaskRole --tags Key=project,Value=doc-
smith Key=tutorial,Value=amazon-ecs-service-connect
        log "Created ecsTaskRole"

        # Wait for role to be available
        sleep 10
    fi
}

# Function to create task definition
```

```
create_task_definition() {
    log "Creating task definition..."

    # Create task definition JSON
    cat > /tmp/task-definition.json << EOF
{
    "family": "service-connect-nginx",
    "networkMode": "awsvpc",
    "requiresCompatibilities": ["FARGATE"],
    "cpu": "256",
    "memory": "512",
    "executionRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskExecutionRole",
    "taskRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskRole",
    "containerDefinitions": [
        {
            "name": "nginx",
            "image": "public.ecr.aws/docker/library/nginx:latest",
            "portMappings": [
                {
                    "containerPort": 80,
                    "protocol": "tcp",
                    "name": "nginx-port"
                }
            ],
            "essential": true,
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/service-connect-nginx",
                    "awslogs-region": "${REGION}",
                    "awslogs-stream-prefix": "ecs"
                }
            }
        }
    ]
}
EOF

    TASK_DEF_OUTPUT=$(aws ecs register-task-definition --cli-input-json file:///
tmp/task-definition.json 2>&1)
    check_for_errors "$TASK_DEF_OUTPUT" "register-task-definition"

    TASK_DEF_ARN=$(echo "$TASK_DEF_OUTPUT" | grep -o '"taskDefinitionArn":
 "[^"]*"' | cut -d'"' -f4)
```

```
    track_resource "TASK_DEF:service-connect-nginx"
    log "Created task definition: $TASK_DEF_ARN"

    # Clean up temporary file
    rm -f /tmp/task-definition.json
}

# Function to create ECS service with Service Connect
create_ecs_service() {
    log "Creating ECS service with Service Connect..."

    # Create service definition JSON
    cat > /tmp/service-definition.json << EOF
{
    "serviceName": "service-connect-nginx-service",
    "cluster": "${CLUSTER_NAME}",
    "taskDefinition": "service-connect-nginx",
    "desiredCount": 1,
    "launchType": "FARGATE",
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "subnets": ["${PUBLIC_SUBNET1}", "${PUBLIC_SUBNET2}"],
            "securityGroups": ["${SECURITY_GROUP_ID}"],
            "assignPublicIp": "ENABLED"
        }
    },
    "serviceConnectConfiguration": {
        "enabled": true,
        "namespace": "${NAMESPACE_NAME}",
        "services": [
            {
                "portName": "nginx-port",
                "discoveryName": "nginx",
                "clientAliases": [
                    {
                        "port": 80,
                        "dnsName": "nginx"
                    }
                ]
            }
        ],
        "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
```

```
                "awslogs-group": "/ecs/service-connect-proxy",
                "awslogs-region": "${REGION}",
                "awslogs-stream-prefix": "ecs-service-connect"
            }
        }
    },
    "tags": [
        {
            "key": "Environment",
            "value": "tutorial"
        },
        {
            "key": "project",
            "value": "doc-smith"
        },
        {
            "key": "tutorial",
            "value": "amazon-ecs-service-connect"
        }
    ]
}
EOF

    SERVICE_OUTPUT=$(aws ecs create-service --cli-input-json file:///tmp/service-
definition.json 2>&1)
    check_for_errors "$SERVICE_OUTPUT" "create-service"

    track_resource "SERVICE:service-connect-nginx-service"
    log "Created ECS service: service-connect-nginx-service"

    wait_for_resource "service" "service-connect-nginx-service"

    # Clean up temporary file
    rm -f /tmp/service-definition.json
}

# Function to verify deployment
verify_deployment() {
    log "Verifying deployment..."

    # Check service status
    SERVICE_STATUS=$(aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
```

```
        --query 'services[0].status' --output text)
    log "Service status: $SERVICE_STATUS"

    # Check running tasks
    RUNNING_COUNT=$(aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].runningCount' --output text)
    log "Running tasks: $RUNNING_COUNT"

    # Get task ARN
    TASK_ARN=$(aws ecs list-tasks \
        --cluster "$CLUSTER_NAME" \
        --service-name "service-connect-nginx-service" \
        --query 'taskArns[0]' --output text)

    if [[ "$TASK_ARN" != "None" && -n "$TASK_ARN" ]]; then
        log "Task ARN: $TASK_ARN"

        # Try to get task IP address
        TASK_IP=$(aws ecs describe-tasks \
            --cluster "$CLUSTER_NAME" \
            --tasks "$TASK_ARN" \
            --query 'tasks[0].attachments[0].details[?
name==`privateIPv4Address`].value' \
            --output text 2>/dev/null || echo "")

        if [[ -n "$TASK_IP" && "$TASK_IP" != "None" ]]; then
            log "Task IP address: $TASK_IP"
        else
            log "Could not retrieve task IP address"
        fi
    fi

    # Check Service Connect namespace
    NAMESPACE_STATUS=$(aws servicediscovery list-namespaces \
        --filters Name=TYPE,Values=HTTP \
        --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")

    if [[ -n "$NAMESPACE_STATUS" && "$NAMESPACE_STATUS" != "None" ]]; then
        log "Service Connect namespace '$NAMESPACE_NAME' is active"
    else
        log "Service Connect namespace '$NAMESPACE_NAME' not found or not active"
```

```
    fi

    # Display Service Connect configuration
    log "Service Connect configuration:"
    aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].serviceConnectConfiguration' 2>/dev/null || true
}

# Function to display created resources
display_resources() {
    echo ""
    echo "============================================="
    echo "CREATED RESOURCES"
    echo "============================================="
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "- $resource"
    done
    echo "============================================="
    echo ""
}

# Function to cleanup resources
cleanup_resources() {
    log "Starting cleanup process..."

    # Delete resources in reverse order of creation
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        resource="${CREATED_RESOURCES[i]}"
        resource_type=$(echo "$resource" | cut -d':' -f1)
        resource_id=$(echo "$resource" | cut -d':' -f2)

        log "Cleaning up $resource_type: $resource_id"

        case "$resource_type" in
            "SERVICE")
                aws ecs update-service --cluster "$CLUSTER_NAME" --service
 "$resource_id" --desired-count 0 2>&1 | grep -qi "error" && log "Warning: Failed
 to scale down service $resource_id"
                aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$resource_id" 2>/dev/null || true
```

```
                aws ecs delete-service --cluster "$CLUSTER_NAME" --service
 "$resource_id" --force 2>&1 | grep -qi "error" && log "Warning: Failed to delete
 service $resource_id"
                ;;
            "TASK_DEF")
                TASK_DEF_ARNS=$(aws ecs list-task-definitions --family-prefix
 "$resource_id" --query 'taskDefinitionArns' --output text 2>/dev/null)
                for arn in $TASK_DEF_ARNS; do
                    aws ecs deregister-task-definition --task-definition "$arn"
 >/dev/null 2>&1 || true
                done
                ;;
            "ROLE")
                aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
                aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete role $resource_id"
                ;;
            "IAM_ROLE")
                aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
                aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete role $resource_id"
                ;;
            "CLUSTER")
                aws ecs delete-cluster --cluster "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete cluster $resource_id"
                ;;
            "SG")
                for attempt in 1 2 3 4 5; do
                    if aws ec2 delete-security-group --group-id "$resource_id"
 2>/dev/null; then
                        break
                    fi
                    log "Security group $resource_id still has dependencies,
 retrying in 30s ($attempt/5)..."
                    sleep 30
                done
                ;;
            "LOG_GROUP")
                aws logs delete-log-group --log-group-name "$resource_id" 2>&1 |
 grep -qi "error" && log "Warning: Failed to delete log group $resource_id"
```

```
                ;;
            "NAMESPACE")
                # First, delete any services in the namespace
                NAMESPACE_SERVICES=$(aws servicediscovery list-services \
                    --filters Name=NAMESPACE_ID,Values="$resource_id" \
                    --query 'Services[].Id' --output text 2>/dev/null || echo "")

                if [[ -n "$NAMESPACE_SERVICES" && "$NAMESPACE_SERVICES" !=
 "None" ]]; then
                    for service_id in $NAMESPACE_SERVICES; do
                        aws servicediscovery delete-service --id "$service_id" >/
dev/null 2>&1 || true
                        sleep 2
                    done
                fi

                # Then delete the namespace
                aws servicediscovery delete-namespace --id "$resource_id" >/dev/
null 2>&1 || true
                ;;
        esac

        sleep 2  # Brief pause between deletions
    done

    # Clean up temporary files
    rm -f /tmp/ecs-task-trust-policy.json
    rm -f /tmp/task-definition.json
    rm -f /tmp/service-definition.json

    log "Cleanup completed"
}

# Main execution
main() {
    log "Starting $SCRIPT_NAME v4 (Default VPC)"
    log "Region: $REGION"
    log "Log file: $LOG_FILE"

    # Get AWS account ID
    get_account_id

    # Setup infrastructure using default VPC
    setup_default_vpc_infrastructure
```

```
    # Create CloudWatch log groups
    create_log_groups

    # Create ECS cluster
    create_ecs_cluster

    # Create IAM roles
    create_iam_roles

    # Create task definition
    create_task_definition

    # Create ECS service
    create_ecs_service

    # Verify deployment
    verify_deployment

    log "Tutorial completed successfully!"

    # Display created resources
    display_resources

    # Ask user if they want to clean up
    echo ""
    echo "============================================"
    echo "CLEANUP CONFIRMATION"
    echo "============================================"
    echo "Do you want to clean up all created resources? (y/n): "
    CLEANUP_CHOICE="y"

    if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
        cleanup_resources
        log "All resources have been cleaned up"
    else
        log "Resources left intact. You can clean them up later by running the
cleanup function."
        echo ""
        echo "To clean up resources later, you can use the AWS CLI commands or
the AWS Management Console."
        echo "Remember to delete resources in the correct order to avoid
dependency issues."
    fi
```

```
}

# Make script executable and run
chmod +x "$0"
main "$@"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress
  - CreateCluster
  - CreateLogGroup
  - CreateRole
  - CreateSecurityGroup
  - CreateService
  - DeleteCluster
  - DeleteLogGroup
  - DeleteNamespace
  - DeleteRole
  - DeleteSecurityGroup
  - DeleteService (AWS Cloud Map)
  - DeleteService (Amazon ECS)
  - DeregisterTaskDefinition
  - DescribeClusters
  - DescribeServices
  - DescribeSubnets
  - DescribeTasks
  - DescribeVpcs
  - DetachRolePolicy
  - GetCallerIdentity
  - GetRole
  - ListNamespaces

- ListServices
- ListTaskDefinitions
- ListTasks
- RegisterTaskDefinition
- UpdateService
- Attendi (Amazon EC2)
- Attendi (Amazon ECS)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Costruisci un URL con AWS STS per utenti federati che utilizzano un AWS SDK

L'esempio di codice seguente mostra come:

- Creare un ruolo IAM che conceda l'accesso in sola lettura alle risorse Amazon S3 dell'account corrente.
- Ottieni un token di sicurezza dall'endpoint della AWS federazione.
- Creare un URL che possa essere utilizzato per accedere alla console con credenziali federate.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Crea un ruolo che conceda l'accesso in sola lettura alle risorse Amazon S3 dell'account corrente.

```
def setup(iam_resource):
    """
```

```
    Creates a role that can be assumed by the current user.
    Attaches a policy that allows only Amazon S3 read-only access.

    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
 instance
                         that has the permission to create a role.
    :return: The newly created role.
    """
    role = iam_resource.create_role(
        RoleName=unique_name("role"),
        AssumeRolePolicyDocument=json.dumps(
            {
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {"AWS": iam_resource.CurrentUser().arn},
                        "Action": "sts:AssumeRole",
                    }
                ],
            }
        ),
    )
    role.attach_policy(PolicyArn="arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess")
    print(f"Created role {role.name}.")

    print("Give AWS time to propagate these new resources and connections.",
 end="")
    progress_bar(10)

    return role
```

Ottieni un token di sicurezza dall'endpoint AWS federativo e crea un URL che può essere utilizzato per accedere alla console con credenziali federate.

```
def construct_federated_url(assume_role_arn, session_name, issuer, sts_client):
    """
    Constructs a URL that gives federated users direct access to the AWS
 Management
```

```
    Console.

    1. Acquires temporary credentials from AWS Security Token Service (AWS STS)
that
       can be used to assume a role with limited permissions.
    2. Uses the temporary credentials to request a sign-in token from the
       AWS federation endpoint.
    3. Builds a URL that can be used in a browser to navigate to the AWS
federation
       endpoint, includes the sign-in token for authentication, and redirects to
       the AWS Management Console with permissions defined by the role that was
       specified in step 1.

    :param assume_role_arn: The role that specifies the permissions that are
granted.
                            The current user must have permission to assume the
role.
    :param session_name: The name for the STS session.
    :param issuer: The organization that issues the URL.
    :param sts_client: A Boto3 STS instance that can assume the role.
    :return: The federated URL.
    """
    response = sts_client.assume_role(
        RoleArn=assume_role_arn, RoleSessionName=session_name
    )
    temp_credentials = response["Credentials"]
    print(f"Assumed role {assume_role_arn} and got temporary credentials.")

    session_data = {
        "sessionId": temp_credentials["AccessKeyId"],
        "sessionKey": temp_credentials["SecretAccessKey"],
        "sessionToken": temp_credentials["SessionToken"],
    }
    aws_federated_signin_endpoint = "https://signin.aws.amazon.com/federation"

    # Make a request to the AWS federation endpoint to get a sign-in token.
    # The requests.get function URL-encodes the parameters and builds the query
string
    # before making the request.
    response = requests.get(
        aws_federated_signin_endpoint,
        params={
            "Action": "getSigninToken",
            "SessionDuration": str(datetime.timedelta(hours=12).seconds),
```

```
            "Session": json.dumps(session_data),
        },
    )
    signin_token = json.loads(response.text)
    print(f"Got a sign-in token from the AWS sign-in federation endpoint.")

    # Make a federated URL that can be used to sign into the AWS Management
 Console.
    query_string = urllib.parse.urlencode(
        {
            "Action": "login",
            "Issuer": issuer,
            "Destination": "https://console.aws.amazon.com/",
            "SigninToken": signin_token["SigninToken"],
        }
    )
    federated_url = f"{aws_federated_signin_endpoint}?{query_string}"
    return federated_url
```

Elimina le risorse create per la demo.

```
def teardown(role):
    """
    Removes all resources created during setup.

    :param role: The demo role.
    """
    for attached in role.attached_policies.all():
        role.detach_policy(PolicyArn=attached.arn)
        print(f"Detached {attached.policy_name}.")
    role.delete()
    print(f"Deleted {role.name}.")
```

Esegui questo scenario utilizzando le funzioni definite in precedenza.

```
def usage_demo():
    """Drives the demonstration."""
```

```
    print("-" * 88)
    print(f"Welcome to the AWS Security Token Service federated URL demo.")
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    role = setup(iam_resource)
    sts_client = boto3.client("sts")
    try:
        federated_url = construct_federated_url(
            role.arn, "AssumeRoleDemoSession", "example.org", sts_client
        )
        print(
            "Constructed a federated URL that can be used to connect to the "
            "AWS Management Console with role-defined permissions:"
        )
        print("-" * 88)
        print(federated_url)
        print("-" * 88)
        _ = input(
            "Copy and paste the above URL into a browser to open the AWS "
            "Management Console with limited permissions. When done, press "
            "Enter to clean up and complete this demo."
        )
    finally:
        teardown(role)
        print("Thanks for watching!")
```

- Per i dettagli sull'API, consulta AssumeRole AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea un'attività contenitore per il tipo di avvio serverless

L'esempio di codice seguente mostra come:

- Creazione del cluster
- Creazione di una definizione di attività
- Crea il servizio

- Eliminazione

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon ECS Fargate Tutorial Script - Version 5
# This script creates an ECS cluster, task definition, and service using Fargate
 launch type
# Fixed version with proper resource dependency handling during cleanup
# Security improvements applied

set -e  # Exit on any error

# Initialize logging
LOG_FILE="ecs-fargate-tutorial-v5.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon ECS Fargate tutorial at $(date)"
echo "Log file: $LOG_FILE"

# Generate random identifier for unique resource names
RANDOM_ID=$(openssl rand -hex 6)
CLUSTER_NAME="fargate-cluster-${RANDOM_ID}"
SERVICE_NAME="fargate-service-${RANDOM_ID}"
TASK_FAMILY="sample-fargate-${RANDOM_ID}"
SECURITY_GROUP_NAME="ecs-fargate-sg-${RANDOM_ID}"

# Array to track created resources for cleanup
CREATED_RESOURCES=()

# Function to log and execute commands with input validation
```

```
execute_command() {
    local cmd="$1"
    local description="$2"

    # Validate that cmd is not empty
    if [[ -z "$cmd" ]]; then
        echo "ERROR: Command is empty"
        return 1
    fi

    echo ""
    echo "========================================="
    echo "EXECUTING: $description"
    echo "COMMAND: $cmd"
    echo "========================================="

    local output
    local exit_code
    set +e  # Temporarily disable exit on error
    output=$(eval "$cmd" 2>&1)
    exit_code=$?
    set -e  # Re-enable exit on error

    if [[ $exit_code -eq 0 ]]; then
        echo "SUCCESS: $description"
        echo "OUTPUT: $output"
        return 0
    else
        echo "FAILED: $description"
        echo "EXIT CODE: $exit_code"
        echo "OUTPUT: $output"
        return 1
    fi
}

# Function to check for actual AWS API errors in command output
check_for_aws_errors() {
    local output="$1"
    local description="$2"

    # Look for specific AWS error patterns, not just the word "error"
    if echo "$output" | grep -qi "An error occurred\|InvalidParameter\|
AccessDenied\|ResourceNotFound\|ValidationException"; then
        echo "AWS API ERROR detected in output for: $description"
```

```
        echo "Output: $output"
        return 1
    fi
    return 0
}

# Function to safely extract JSON values
safe_json_extract() {
    local json="$1"
    local key="$2"
    local value

    value=$(echo "$json" | grep -o "\"$key\": \"[^\"]*\"" | cut -d'"' -f4 2>/dev/
null || echo "")
    echo "$value"
}

# Function to wait for network interfaces to be cleaned up
wait_for_network_interfaces_cleanup() {
    local security_group_id="$1"
    local max_attempts=30
    local attempt=1

    # Validate security group ID format
    if [[ ! "$security_group_id" =~ ^sg-[a-z0-9]{8,17}$ ]]; then
        echo "ERROR: Invalid security group ID format: $security_group_id"
        return 1
    fi

    echo "Waiting for network interfaces to be cleaned up..."

    while [[ $attempt -le $max_attempts ]]; do
        echo "Attempt $attempt/$max_attempts: Checking for dependent network
 interfaces..."

        # Check if there are any network interfaces still using this security
 group
        local eni_count
        eni_count=$(aws ec2 describe-network-interfaces \
            --filters "Name=group-id,Values=$security_group_id" \
            --query "length(NetworkInterfaces)" \
            --output text 2>/dev/null || echo "0")

        if [[ "$eni_count" == "0" ]]; then
```

```
            echo "No network interfaces found using security group
 $security_group_id"
            return 0
        else
            echo "Found $eni_count network interface(s) still using security
 group $security_group_id"
            echo "Waiting 10 seconds before next check..."
            sleep 10
            ((attempt++))
        fi
    done

    echo "WARNING: Network interfaces may still be attached after $max_attempts
 attempts"
    echo "This is normal and the security group deletion will be retried"
    return 1
}

# Function to retry security group deletion with exponential backoff
retry_security_group_deletion() {
    local security_group_id="$1"
    local max_attempts=10
    local attempt=1
    local wait_time=5

    # Validate security group ID format
    if [[ ! "$security_group_id" =~ ^sg-[a-z0-9]{8,17}$ ]]; then
        echo "ERROR: Invalid security group ID format: $security_group_id"
        return 1
    fi

    while [[ $attempt -le $max_attempts ]]; do
        echo "Attempt $attempt/$max_attempts: Trying to delete security group
 $security_group_id"

        if execute_command "aws ec2 delete-security-group --group-id
 '$security_group_id'" "Delete security group (attempt $attempt)"; then
            echo "Successfully deleted security group $security_group_id"
            return 0
        else
            if [[ $attempt -eq $max_attempts ]]; then
                echo "FAILED: Could not delete security group $security_group_id
 after $max_attempts attempts"
```

```
                echo "This may be due to network interfaces that are still being
 cleaned up by AWS"
                echo "You can manually delete it later using: aws ec2 delete-
security-group --group-id $security_group_id"
                return 1
            else
                echo "Waiting $wait_time seconds before retry..."
                sleep $wait_time
                wait_time=$((wait_time * 2))  # Exponential backoff
                ((attempt++))
            fi
        fi
    done
}

# Function to cleanup resources with proper dependency handling
cleanup_resources() {
    echo ""
    echo "==========================================="
    echo "CLEANUP PROCESS"
    echo "==========================================="
    echo "The following resources were created:"
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "  - $resource"
    done
    echo ""
    echo "Auto-confirming cleanup of all created resources..."

    CLEANUP_CHOICE="y"

    if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
        echo "Starting cleanup process..."

        # Step 1: Scale service to 0 tasks first, then delete service
        if [[ " ${CREATED_RESOURCES[*]} " =~ " ECS Service: $SERVICE_NAME " ]];
 then
            echo ""
            echo "Step 1: Scaling service to 0 tasks..."
            if execute_command "aws ecs update-service --cluster '$CLUSTER_NAME'
 --service '$SERVICE_NAME' --desired-count 0" "Scale service to 0 tasks"; then
                echo "Waiting for service to stabilize after scaling to 0..."
                execute_command "aws ecs wait services-stable --cluster
 '$CLUSTER_NAME' --services '$SERVICE_NAME'" "Wait for service to stabilize"
```

```
                echo "Deleting service..."
                execute_command "aws ecs delete-service --cluster '$CLUSTER_NAME'
--service '$SERVICE_NAME'" "Delete ECS service"
            else
                echo "WARNING: Failed to scale service. Attempting to delete
anyway..."
                execute_command "aws ecs delete-service --cluster '$CLUSTER_NAME'
--service '$SERVICE_NAME' --force" "Force delete ECS service"
            fi
        fi

        # Step 2: Wait a bit for tasks to fully terminate
        echo ""
        echo "Step 2: Waiting for tasks to fully terminate..."
        sleep 15

        # Step 3: Delete cluster
        if [[ " ${CREATED_RESOURCES[*]} " =~ " ECS Cluster: $CLUSTER_NAME " ]];
then
            echo ""
            echo "Step 3: Deleting cluster..."
            execute_command "aws ecs delete-cluster --cluster '$CLUSTER_NAME'"
"Delete ECS cluster"
        fi

        # Step 4: Wait for network interfaces to be cleaned up, then delete
security group
        if [[ -n "$SECURITY_GROUP_ID" && "$SECURITY_GROUP_ID" != "None" ]]; then
            echo ""
            echo "Step 4: Cleaning up security group..."

            # First, wait for network interfaces to be cleaned up
            wait_for_network_interfaces_cleanup "$SECURITY_GROUP_ID"

            # Then retry security group deletion with backoff
            retry_security_group_deletion "$SECURITY_GROUP_ID"
        fi

        # Step 5: Clean up task definition (deregister all revisions)
        if [[ " ${CREATED_RESOURCES[*]} " =~ " Task Definition: $TASK_FAMILY
" ]]; then
            echo ""
            echo "Step 5: Deregistering task definition revisions..."
```

```
            # Get all revisions of the task definition
            local revisions
            revisions=$(aws ecs list-task-definitions --family-prefix
 "$TASK_FAMILY" --query "taskDefinitionArns" --output text 2>/dev/null || echo
 "")

            if [[ -n "$revisions" && "$revisions" != "None" ]]; then
                for revision_arn in $revisions; do
                    echo "Deregistering task definition: $revision_arn"
                    execute_command "aws ecs deregister-task-definition --task-
definition '$revision_arn'" "Deregister task definition $revision_arn" || true
                done
            else
                echo "No task definition revisions found to deregister"
            fi
        fi

        echo ""
        echo "============================================"
        echo "CLEANUP COMPLETED"
        echo "============================================"
        echo "All resources have been cleaned up successfully!"

    else
        echo "Cleanup skipped. Resources remain active."
        echo ""
        echo "To clean up manually later, use the following commands in order:"
        echo "1. Scale service to 0: aws ecs update-service --cluster
 $CLUSTER_NAME --service $SERVICE_NAME --desired-count 0"
        echo "2. Wait for stability: aws ecs wait services-stable --cluster
 $CLUSTER_NAME --services $SERVICE_NAME"
        echo "3. Delete service: aws ecs delete-service --cluster $CLUSTER_NAME
 --service $SERVICE_NAME"
        echo "4. Delete cluster: aws ecs delete-cluster --cluster $CLUSTER_NAME"
        echo "5. Wait 2-3 minutes, then delete security group: aws ec2 delete-
security-group --group-id $SECURITY_GROUP_ID"
        if [[ " ${CREATED_RESOURCES[*]} " =~ " Task Definition: $TASK_FAMILY
 " ]]; then
            echo "6. Deregister task definitions: aws ecs list-task-definitions
 --family-prefix $TASK_FAMILY"
            echo "   Then for each ARN: aws ecs deregister-task-definition --
task-definition <ARN>"
        fi
    fi
```

```
}

# Trap to handle script interruption
trap cleanup_resources EXIT

echo "Using random identifier: $RANDOM_ID"
echo "Cluster name: $CLUSTER_NAME"
echo "Service name: $SERVICE_NAME"
echo "Task family: $TASK_FAMILY"

# Step 1: Ensure ECS task execution role exists
echo ""
echo "============================================="
echo "STEP 1: VERIFY ECS TASK EXECUTION ROLE"
echo "============================================="

ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
if [[ ! "$ACCOUNT_ID" =~ ^[0-9]{12}$ ]]; then
    echo "ERROR: Invalid AWS Account ID retrieved: $ACCOUNT_ID"
    exit 1
fi

EXECUTION_ROLE_ARN="arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskExecutionRole"

# Check if role exists
if aws iam get-role --role-name ecsTaskExecutionRole >/dev/null 2>&1; then
    echo "ECS task execution role already exists"
else
    echo "Creating ECS task execution role..."

    # Create trust policy with strict validation
    cat > trust-policy.json << 'EOF'
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

```
EOF

    # Validate JSON before using
    if ! jq empty trust-policy.json 2>/dev/null; then
        echo "ERROR: Invalid JSON in trust policy"
        rm -f trust-policy.json
        exit 1
    fi

    execute_command "aws iam create-role --role-name ecsTaskExecutionRole --
assume-role-policy-document file://trust-policy.json" "Create ECS task execution
 role"

    aws iam tag-role --role-name ecsTaskExecutionRole --tags
 Key=project,Value=doc-smith Key=tutorial,Value=amazon-ecs-fargate-linux

    execute_command "aws iam attach-role-policy --role-name
 ecsTaskExecutionRole --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy" "Attach ECS task execution policy"

    # Clean up temporary file securely
    shred -vfz -n 3 trust-policy.json 2>/dev/null || rm -f trust-policy.json

    CREATED_RESOURCES+=("IAM Role: ecsTaskExecutionRole")
fi

# Step 2: Create ECS cluster
echo ""
echo "============================================="
echo "STEP 2: CREATE ECS CLUSTER"
echo "============================================="

CLUSTER_OUTPUT=$(execute_command "aws ecs create-cluster --cluster-name
 '$CLUSTER_NAME' --tags key=project,value=doc-smith key=tutorial,value=amazon-
ecs-fargate-linux" "Create ECS cluster")
check_for_aws_errors "$CLUSTER_OUTPUT" "Create ECS cluster"

CREATED_RESOURCES+=("ECS Cluster: $CLUSTER_NAME")

# Step 3: Create task definition
echo ""
echo "============================================="
echo "STEP 3: CREATE TASK DEFINITION"
echo "============================================="
```

```
# Create task definition JSON with validated inputs
cat > task-definition.json << EOF
{
    "family": "$TASK_FAMILY",
    "networkMode": "awsvpc",
    "requiresCompatibilities": ["FARGATE"],
    "cpu": "256",
    "memory": "512",
    "executionRoleArn": "$EXECUTION_ROLE_ARN",
    "containerDefinitions": [
        {
            "name": "fargate-app",
            "image": "public.ecr.aws/docker/library/httpd:2.4-alpine",
            "portMappings": [
                {
                    "containerPort": 80,
                    "hostPort": 80,
                    "protocol": "tcp"
                }
            ],
            "essential": true,
            "entryPoint": ["sh", "-c"],
            "command": [
                "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
 App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </
head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</
h1> <h2>Congratulations!</h2> <p>Your application is now running on a container
 in Amazon ECS.</p> </div></body></html>' >  /usr/local/apache2/htdocs/index.html
 && httpd-foreground\""
            ],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/fargate-sample",
                    "awslogs-region": "us-east-1",
                    "awslogs-stream-prefix": "ecs"
                }
            }
        }
    ]
}
EOF
```

```
# Validate JSON before using
if ! jq empty task-definition.json 2>/dev/null; then
    echo "ERROR: Invalid JSON in task definition"
    rm -f task-definition.json
    exit 1
fi

TASK_DEF_OUTPUT=$(execute_command "aws ecs register-task-definition --cli-input-
json file://task-definition.json" "Register task definition")
check_for_aws_errors "$TASK_DEF_OUTPUT" "Register task definition"

# Clean up temporary file securely
shred -vfz -n 3 task-definition.json 2>/dev/null || rm -f task-definition.json

CREATED_RESOURCES+=("Task Definition: $TASK_FAMILY")

# Step 4: Set up networking
echo ""
echo "==========================================="
echo "STEP 4: SET UP NETWORKING"
echo "==========================================="

# Get default VPC ID
VPC_ID=$(aws ec2 describe-vpcs --filters "Name=is-default,Values=true" --query
 "Vpcs[0].VpcId" --output text)
if [[ "$VPC_ID" == "None" || -z "$VPC_ID" ]]; then
    echo "ERROR: No default VPC found. Please create a default VPC or specify a
 custom VPC."
    exit 1
fi

# Validate VPC ID format
if [[ ! "$VPC_ID" =~ ^vpc-[a-z0-9]{8,17}$ ]]; then
    echo "ERROR: Invalid VPC ID format: $VPC_ID"
    exit 1
fi

echo "Using default VPC: $VPC_ID"

# Create security group with restricted access
# Note: This allows HTTP access from anywhere for demo purposes
# In production, restrict source to specific IP ranges or security groups
SECURITY_GROUP_OUTPUT=$(execute_command "aws ec2 create-security-group --
group-name '$SECURITY_GROUP_NAME' --description 'Security group for ECS
```

```
 Fargate tutorial - HTTP access' --vpc-id '$VPC_ID' --tag-specifications
 'ResourceType=security-group,Tags=[{Key=project,Value=doc-smith},
{Key=tutorial,Value=amazon-ecs-fargate-linux}]'" "Create security group")
check_for_aws_errors "$SECURITY_GROUP_OUTPUT" "Create security group"

SECURITY_GROUP_ID=$(echo "$SECURITY_GROUP_OUTPUT" | grep -o '"GroupId": "[^"]*"'
 | head -1 | cut -d'"' -f4)
if [[ -z "$SECURITY_GROUP_ID" || "$SECURITY_GROUP_ID" == "None" ]]; then
    SECURITY_GROUP_ID=$(aws ec2 describe-security-groups --group-names
 "$SECURITY_GROUP_NAME" --filters "Name=vpc-id,Values=$VPC_ID" --query
 "SecurityGroups[0].GroupId" --output text)
fi

# Validate security group ID format
if [[ ! "$SECURITY_GROUP_ID" =~ ^sg-[a-z0-9]{8,17}$ ]]; then
    echo "ERROR: Invalid security group ID format: $SECURITY_GROUP_ID"
    exit 1
fi

echo "Created security group: $SECURITY_GROUP_ID"
CREATED_RESOURCES+=("Security Group: $SECURITY_GROUP_ID")

# Add HTTP inbound rule
# WARNING: This allows HTTP access from anywhere (0.0.0.0/0)
# In production environments, restrict this to specific IP ranges
execute_command "aws ec2 authorize-security-group-ingress --group-id
 '$SECURITY_GROUP_ID' --protocol tcp --port 80 --cidr 0.0.0.0/0" "Add HTTP
 inbound rule to security group"

# Get subnet IDs from default VPC
echo "Getting subnet IDs from default VPC..."
SUBNET_IDS_RAW=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC_ID"
 --query "Subnets[*].SubnetId" --output text)
if [[ -z "$SUBNET_IDS_RAW" ]]; then
    echo "ERROR: No subnets found in default VPC"
    exit 1
fi

# Convert to proper comma-separated format, handling both spaces and tabs
SUBNET_IDS_COMMA=$(echo "$SUBNET_IDS_RAW" | tr -s '[:space:]' ',' | sed 's/,$//')
echo "Raw subnet IDs: $SUBNET_IDS_RAW"
echo "Formatted subnet IDs: $SUBNET_IDS_COMMA"

# Validate subnet IDs format
```

```
if [[ ! "$SUBNET_IDS_COMMA" =~ ^subnet-[a-z0-9]+(,subnet-[a-z0-9]+)*$ ]]; then
    echo "ERROR: Invalid subnet ID format: $SUBNET_IDS_COMMA"
    exit 1
fi

# Step 5: Create ECS service
echo ""
echo "============================================="
echo "STEP 5: CREATE ECS SERVICE"
echo "============================================="

# Create the service with proper JSON formatting for network configuration
SERVICE_CMD="aws ecs create-service --cluster '$CLUSTER_NAME' --service-name
 '$SERVICE_NAME' --task-definition '$TASK_FAMILY' --desired-count 1 --launch-
type FARGATE --network-configuration '{\"awsvpcConfiguration\":{\"subnets
\":[\"$(echo "$SUBNET_IDS_COMMA" | sed 's/,/","/g')\"],\"securityGroups
\":[\"$SECURITY_GROUP_ID\"],\"assignPublicIp\":\"ENABLED\"}}' --tags
 key=project,value=doc-smith key=tutorial,value=amazon-ecs-fargate-linux"

echo "Service creation command: $SERVICE_CMD"

SERVICE_OUTPUT=$(execute_command "$SERVICE_CMD" "Create ECS service")
check_for_aws_errors "$SERVICE_OUTPUT" "Create ECS service"

CREATED_RESOURCES+=("ECS Service: $SERVICE_NAME")

# Step 6: Wait for service to stabilize and get public IP
echo ""
echo "============================================="
echo "STEP 6: WAIT FOR SERVICE AND GET PUBLIC IP"
echo "============================================="

echo "Waiting for service to stabilize (this may take a few minutes)..."
execute_command "aws ecs wait services-stable --cluster '$CLUSTER_NAME' --
services '$SERVICE_NAME'" "Wait for service to stabilize"

# Get task ARN
TASK_ARN=$(aws ecs list-tasks --cluster "$CLUSTER_NAME" --service-name
 "$SERVICE_NAME" --query "taskArns[0]" --output text)
if [[ "$TASK_ARN" == "None" || -z "$TASK_ARN" ]]; then
    echo "ERROR: No running tasks found for service"
    exit 1
fi
```

```
echo "Task ARN: $TASK_ARN"

# Get network interface ID
ENI_ID=$(aws ecs describe-tasks --cluster "$CLUSTER_NAME" --tasks "$TASK_ARN"
 --query "tasks[0].attachments[0].details[?name=='networkInterfaceId'].value" --
output text)
if [[ "$ENI_ID" == "None" || -z "$ENI_ID" ]]; then
    echo "ERROR: Could not retrieve network interface ID"
    exit 1
fi

# Validate ENI ID format
if [[ ! "$ENI_ID" =~ ^eni-[a-z0-9]{8,17}$ ]]; then
    echo "ERROR: Invalid network interface ID format: $ENI_ID"
    exit 1
fi

echo "Network Interface ID: $ENI_ID"

# Get public IP
PUBLIC_IP=$(aws ec2 describe-network-interfaces --network-interface-ids "$ENI_ID"
 --query "NetworkInterfaces[0].Association.PublicIp" --output text)
if [[ "$PUBLIC_IP" == "None" || -z "$PUBLIC_IP" ]]; then
    echo "WARNING: No public IP assigned to the task"
    echo "The task may be in a private subnet or public IP assignment failed"
else
    # Validate IP format
    if [[ ! "$PUBLIC_IP" =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]];
 then
        echo "ERROR: Invalid IP address format: $PUBLIC_IP"
        PUBLIC_IP=""
    else
        echo ""
        echo "========================================="
        echo "SUCCESS! APPLICATION IS RUNNING"
        echo "========================================="
        echo "Your application is available at: http://$PUBLIC_IP"
        echo "You can test it by opening this URL in your browser"
        echo ""
    fi
fi

# Display service information
echo ""
```

```
echo "============================================="
echo "SERVICE INFORMATION"
echo "============================================="
execute_command "aws ecs describe-services --cluster '$CLUSTER_NAME' --services
 '$SERVICE_NAME'" "Get service details"

echo ""
echo "============================================="
echo "TUTORIAL COMPLETED SUCCESSFULLY"
echo "============================================="
echo "Resources created:"
for resource in "${CREATED_RESOURCES[@]}"; do
    echo "  - $resource"
done

if [[ -n "$PUBLIC_IP" && "$PUBLIC_IP" != "None" ]]; then
    echo ""
    echo "Application URL: http://$PUBLIC_IP"
fi

echo ""
echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress
  - CreateCluster
  - CreateRole
  - CreateSecurityGroup
  - CreateService
  - DeleteCluster
  - DeleteSecurityGroup
  - DeleteService
  - DeregisterTaskDefinition
  - DescribeNetworkInterfaces
  - DescribeSecurityGroups
  - DescribeServices

- DescribeSubnets
- DescribeTasks
- DescribeVpcs
- GetCallerIdentity
- GetRole
- ListTaskDefinitions
- ListTasks
- RegisterTaskDefinition
- UpdateService
- Attendi

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea un'API REST con integrazione proxy di funzioni

L'esempio di codice seguente mostra come:

- Creare un ruolo IAM per l'esecuzione di Lambda
- Creare e distribuire una funzione Lambda
- Crea un'API REST
- Configurare l'integrazione del proxy Lambda
- Distribuzione e test dell'API
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

set -euo pipefail

# Simple API Gateway Lambda Integration Script
# This script creates a REST API with Lambda proxy integration

# Generate random identifiers
FUNCTION_NAME="GetStartedLambdaProxyIntegration-$(openssl rand -hex 4)"
ROLE_NAME="GetStartedLambdaBasicExecutionRole-$(openssl rand -hex 4)"
API_NAME="LambdaProxyAPI-$(openssl rand -hex 4)"

# Get AWS account info
ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
REGION=$(aws configure get region || echo "us-east-1")

# Validate inputs
if [[ -z "$ACCOUNT_ID" ]] || [[ -z "$REGION" ]]; then
    echo "Error: Failed to retrieve AWS account information" >&2
    exit 1
fi

echo "Creating Lambda function code..."

# Create Lambda function code with input validation
cat > lambda_function.py << 'EOF'
import json
import logging

logger = logging.getLogger()
logger.setLevel(logging.INFO)
```

```
def lambda_handler(event, context):
    try:
        logger.info("Received event: %s", json.dumps(event))

        greeter = 'World'

        # Safely retrieve greeter from query string parameters
        query_params = event.get('queryStringParameters') or {}
        if isinstance(query_params, dict) and 'greeter' in query_params:
            greeter_value = query_params.get('greeter')
            if isinstance(greeter_value, str) and greeter_value:
                greeter = greeter_value

        # Safely retrieve greeter from multi-value headers
        multi_headers = event.get('multiValueHeaders') or {}
        if isinstance(multi_headers, dict) and 'greeter' in multi_headers:
            greeter_list = multi_headers.get('greeter', [])
            if isinstance(greeter_list, list) and greeter_list:
                greeter = " and ".join(str(g) for g in greeter_list if g)

        # Safely retrieve greeter from headers
        headers = event.get('headers') or {}
        if isinstance(headers, dict) and 'greeter' in headers:
            greeter_value = headers.get('greeter')
            if isinstance(greeter_value, str) and greeter_value:
                greeter = greeter_value

        # Safely retrieve greeter from body
        body = event.get('body')
        if body and isinstance(body, str):
            try:
                body_dict = json.loads(body)
                if isinstance(body_dict, dict) and 'greeter' in body_dict:
                    greeter_value = body_dict.get('greeter')
                    if isinstance(greeter_value, str) and greeter_value:
                        greeter = greeter_value
            except (json.JSONDecodeError, ValueError) as e:
                logger.warning("Failed to parse body: %s", str(e))

        # Sanitize greeter to prevent injection
        greeter = greeter.replace('"', '\\"').replace("'", "\\'")

        response = {
```

```
            "statusCode": 200,
            "headers": {
                "Content-Type": "application/json"
            },
            "body": json.dumps({"message": f"Hello, {greeter}!"})
        }

        logger.info("Response: %s", json.dumps(response))
        return response

    except Exception as e:
        logger.error("Unexpected error: %s", str(e), exc_info=True)
        return {
            "statusCode": 500,
            "headers": {
                "Content-Type": "application/json"
            },
            "body": json.dumps({"error": "Internal server error"})
        }
EOF

# Create deployment package
zip -q function.zip lambda_function.py || {
    echo "Error: Failed to create function.zip" >&2
    exit 1
}

echo "Creating IAM role..."

# Create IAM trust policy
cat > trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

```
# Create IAM role with error handling
aws iam create-role \
    --role-name "$ROLE_NAME" \
    --assume-role-policy-document file://trust-policy.json \
    --description "Temporary role for Lambda execution" || {
    echo "Error: Failed to create IAM role" >&2
    exit 1
}

aws iam tag-role --role-name "$ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=apigateway-lambda-integration

# Attach execution policy
aws iam attach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" || {
    echo "Error: Failed to attach IAM policy" >&2
    exit 1
}

# Wait for role propagation
sleep 15

echo "Creating Lambda function..."

# Create Lambda function with Python 3.11 (more recent runtime)
aws lambda create-function \
    --function-name "$FUNCTION_NAME" \
    --runtime python3.11 \
    --role "arn:aws:iam::$ACCOUNT_ID:role/$ROLE_NAME" \
    --handler lambda_function.lambda_handler \
    --zip-file fileb://function.zip \
    --timeout 30 \
    --memory-size 128 \
    --environment "Variables={LOG_LEVEL=INFO}" \
    --tags project=doc-smith,tutorial=apigateway-lambda-integration || {
    echo "Error: Failed to create Lambda function" >&2
    exit 1
}

echo "Creating API Gateway..."
```

```
# Create REST API with minimum logging
API_RESPONSE=$(aws apigateway create-rest-api \
    --name "$API_NAME" \
    --endpoint-configuration types=REGIONAL \
    --description "API for Lambda proxy integration tutorial" \
    --tags project=doc-smith,tutorial=apigateway-lambda-integration \
    --output json)

API_ID=$(echo "$API_RESPONSE" | grep -o '"id": "[^"]*"' | head -1 | cut -d'"' -
f4)

if [[ -z "$API_ID" ]]; then
    echo "Error: Failed to create API Gateway" >&2
    exit 1
fi

# Get root resource ID
ROOT_RESOURCE_ID=$(aws apigateway get-resources --rest-api-id "$API_ID" --query
 'items[?path==`/`].id' --output text)

# Create helloworld resource
aws apigateway create-resource \
    --rest-api-id "$API_ID" \
    --parent-id "$ROOT_RESOURCE_ID" \
    --path-part helloworld || {
    echo "Error: Failed to create resource" >&2
    exit 1
}

# Get resource ID
RESOURCE_ID=$(aws apigateway get-resources --rest-api-id "$API_ID" --query
 "items[?pathPart=='helloworld'].id" --output text)

# Create ANY method with no authorization (intentional for tutorial)
aws apigateway put-method \
    --rest-api-id "$API_ID" \
    --resource-id "$RESOURCE_ID" \
    --http-method ANY \
    --authorization-type NONE || {
    echo "Error: Failed to create method" >&2
    exit 1
}

# Set up Lambda proxy integration
```

```
LAMBDA_URI="arn:aws:apigateway:$REGION:lambda:path/2015-03-31/functions/
arn:aws:lambda:$REGION:$ACCOUNT_ID:function:$FUNCTION_NAME/invocations"

aws apigateway put-integration \
    --rest-api-id "$API_ID" \
    --resource-id "$RESOURCE_ID" \
    --http-method ANY \
    --type AWS_PROXY \
    --integration-http-method POST \
    --uri "$LAMBDA_URI" || {
    echo "Error: Failed to create integration" >&2
    exit 1
}

# Grant API Gateway permission to invoke Lambda
STATEMENT_ID="apigateway-invoke-$(openssl rand -hex 4)"
SOURCE_ARN="arn:aws:execute-api:$REGION:$ACCOUNT_ID:$API_ID/*/*"

aws lambda add-permission \
    --function-name "$FUNCTION_NAME" \
    --statement-id "$STATEMENT_ID" \
    --action lambda:InvokeFunction \
    --principal apigateway.amazonaws.com \
    --source-arn "$SOURCE_ARN" || {
    echo "Error: Failed to add Lambda permission" >&2
    exit 1
}

# Deploy API
aws apigateway create-deployment \
    --rest-api-id "$API_ID" \
    --stage-name test \
    --description "Test deployment" || {
    echo "Error: Failed to deploy API" >&2
    exit 1
}

echo "Testing API..."

# Test the API
INVOKE_URL="https://$API_ID.execute-api.$REGION.amazonaws.com/test/helloworld"

echo "API URL: $INVOKE_URL"
```

```
# Test with query parameter (with proper URL encoding)
echo "Testing with query parameter:"
curl -s -X GET "$INVOKE_URL?greeter=John" | jq . 2>/dev/null || curl -s -X GET
 "$INVOKE_URL?greeter=John"
echo ""

# Test with header
echo "Testing with header:"
curl -s -X GET "$INVOKE_URL" \
    -H 'content-type: application/json' \
    -H 'greeter: John' | jq . 2>/dev/null || curl -s -X GET "$INVOKE_URL" \
    -H 'content-type: application/json' \
    -H 'greeter: John'
echo ""

# Test with body
echo "Testing with POST body:"
curl -s -X POST "$INVOKE_URL" \
    -H 'content-type: application/json' \
    -d '{"greeter": "John"}' | jq . 2>/dev/null || curl -s -X POST "$INVOKE_URL"
 \
    -H 'content-type: application/json' \
    -d '{"greeter": "John"}'
echo ""

echo "Tutorial completed! API is available at: $INVOKE_URL"

# Cleanup
echo "Cleaning up resources..."

# Delete API
aws apigateway delete-rest-api --rest-api-id "$API_ID" || echo "Warning: Failed
 to delete API" >&2

# Delete Lambda function
aws lambda delete-function --function-name "$FUNCTION_NAME" || echo "Warning:
 Failed to delete Lambda function" >&2

# Detach policy and delete role
aws iam detach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" || echo "Warning: Failed to detach policy" >&2
```

```
aws iam delete-role --role-name "$ROLE_NAME" || echo "Warning: Failed to delete
 role" >&2

# Clean up local files securely
rm -f lambda_function.py function.zip trust-policy.json

echo "Cleanup completed!"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AddPermission
  - AttachRolePolicy
  - CreateDeployment
  - CreateFunction
  - CreateResource
  - CreateRestApi
  - CreateRole
  - DeleteFunction
  - DeleteRestApi
  - DeleteRole
  - DetachRolePolicy
  - GetCallerIdentity
  - GetResources
  - GetRestApis
  - PutIntegration
  - PutMethod

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione di un servizio container per le istanze di macchine virtuali

L'esempio di codice seguente mostra come:

- Crea un cluster ECS
- Crea e monitora un servizio
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# ECS EC2 Launch Type Tutorial Script - UPDATED VERSION
# This script demonstrates creating an ECS cluster, launching a container
 instance,
# registering a task definition, and creating a service using the EC2 launch
 type.
# Updated to match the tutorial draft with nginx web server and service creation.
#
# - UPDATED: Changed from sleep task to nginx web server with service

set -e  # Exit on any error

# Configuration
SCRIPT_NAME="ecs-ec2-tutorial"
LOG_FILE="${SCRIPT_NAME}-$(date +%Y%m%d-%H%M%S).log"
CLUSTER_NAME="tutorial-cluster-$(openssl rand -hex 4)"
TASK_FAMILY="nginx-task-$(openssl rand -hex 4)"
SERVICE_NAME="nginx-service-$(openssl rand -hex 4)"
KEY_PAIR_NAME="ecs-tutorial-key-$(openssl rand -hex 4)"
SECURITY_GROUP_NAME="ecs-tutorial-sg-$(openssl rand -hex 4)"

# Tags
PROJECT_TAG="doc-smith"
TUTORIAL_TAG="ecs-ec2"
```

```
# Get current AWS region dynamically
AWS_REGION=$(aws configure get region || echo "us-east-1")

# Resource tracking arrays
CREATED_RESOURCES=()
CLEANUP_ORDER=()

# Logging function
log() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | tee -a "$LOG_FILE"
}

# Error handling function
handle_error() {
    local exit_code=$?
    log "ERROR: Script failed with exit code $exit_code"
    log "ERROR: Last command: $BASH_COMMAND"

    echo ""
    echo "========================================="
    echo "ERROR OCCURRED - ATTEMPTING CLEANUP"
    echo "========================================="
    echo "Resources created before error:"
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "  - $resource"
    done

    cleanup_resources
    exit $exit_code
}

# Set error trap
trap handle_error ERR

# FIXED: Enhanced cleanup function with proper error handling and logging
cleanup_resources() {
    log "Starting cleanup process..."
    local cleanup_errors=0

    # Delete service first (this will stop tasks automatically)
    if [[ -n "${SERVICE_ARN:-}" ]]; then
        log "Updating service to desired count 0: $SERVICE_NAME"
        if ! aws ecs update-service --cluster "$CLUSTER_NAME" --service
 "$SERVICE_NAME" --desired-count 0 2>>"$LOG_FILE"; then
```

```
            log "WARNING: Failed to update service desired count to 0"
            ((cleanup_errors++))
        else
            log "Waiting for service tasks to stop..."
            sleep 30  # Give time for tasks to stop
        fi

        log "Deleting service: $SERVICE_NAME"
        if ! aws ecs delete-service --cluster "$CLUSTER_NAME" --service
 "$SERVICE_NAME" 2>>"$LOG_FILE"; then
            log "WARNING: Failed to delete service $SERVICE_NAME"
            ((cleanup_errors++))
        fi
    fi

    # Stop and delete any remaining tasks
    if [[ -n "${TASK_ARN:-}" ]]; then
        log "Stopping task: $TASK_ARN"
        if ! aws ecs stop-task --cluster "$CLUSTER_NAME" --task "$TASK_ARN" --
reason "Tutorial cleanup" 2>>"$LOG_FILE"; then
            log "WARNING: Failed to stop task $TASK_ARN"
            ((cleanup_errors++))
        else
            log "Waiting for task to stop..."
            if ! aws ecs wait tasks-stopped --cluster "$CLUSTER_NAME" --tasks
 "$TASK_ARN" 2>>"$LOG_FILE"; then
                log "WARNING: Task stop wait failed for $TASK_ARN"
                ((cleanup_errors++))
            fi
        fi
    fi

    # Deregister task definition
    if [[ -n "${TASK_DEFINITION_ARN:-}" ]]; then
        log "Deregistering task definition: $TASK_DEFINITION_ARN"
        if ! aws ecs deregister-task-definition --task-definition
 "$TASK_DEFINITION_ARN" 2>>"$LOG_FILE"; then
            log "WARNING: Failed to deregister task definition
 $TASK_DEFINITION_ARN"
            ((cleanup_errors++))
        fi
    fi

    # Terminate EC2 instance
```

```
    if [[ -n "${INSTANCE_ID:-}" ]]; then
        log "Terminating EC2 instance: $INSTANCE_ID"
        if ! aws ec2 terminate-instances --instance-ids "$INSTANCE_ID"
2>>"$LOG_FILE"; then
            log "WARNING: Failed to terminate instance $INSTANCE_ID"
            ((cleanup_errors++))
        else
            log "Waiting for instance to terminate..."
            if ! aws ec2 wait instance-terminated --instance-ids "$INSTANCE_ID"
2>>"$LOG_FILE"; then
                log "WARNING: Instance termination wait failed for $INSTANCE_ID"
                ((cleanup_errors++))
            fi
        fi
    fi

    # Delete security group with retry logic
    if [[ -n "${SECURITY_GROUP_ID:-}" ]]; then
        log "Deleting security group: $SECURITY_GROUP_ID"
        local retry_count=0
        local max_retries=3

        while [[ $retry_count -lt $max_retries ]]; do
            if aws ec2 delete-security-group --group-id "$SECURITY_GROUP_ID"
2>>"$LOG_FILE"; then
                log "Successfully deleted security group"
                break
            else
                ((retry_count++))
                if [[ $retry_count -lt $max_retries ]]; then
                    log "Retry $retry_count/$max_retries: Waiting 10 seconds
before retrying security group deletion..."
                    sleep 10
                else
                    log "ERROR: Failed to delete security group after
$max_retries attempts"
                    ((cleanup_errors++))
                fi
            fi
        done
    fi

    # Delete key pair
    if [[ -n "${KEY_PAIR_NAME:-}" ]]; then
```

```
        log "Deleting key pair: $KEY_PAIR_NAME"
        if ! aws ec2 delete-key-pair --key-name "$KEY_PAIR_NAME" 2>>"$LOG_FILE";
 then
            log "WARNING: Failed to delete key pair $KEY_PAIR_NAME"
            ((cleanup_errors++))
        fi
        rm -f "${KEY_PAIR_NAME}.pem" 2>>"$LOG_FILE" || log "WARNING: Failed to
 remove local key file"
    fi

    # Delete ECS cluster
    if [[ -n "${CLUSTER_NAME:-}" ]]; then
        log "Deleting ECS cluster: $CLUSTER_NAME"
        if ! aws ecs delete-cluster --cluster "$CLUSTER_NAME" 2>>"$LOG_FILE";
 then
            log "WARNING: Failed to delete cluster $CLUSTER_NAME"
            ((cleanup_errors++))
        fi
    fi

    if [[ $cleanup_errors -eq 0 ]]; then
        log "Cleanup completed successfully"
    else
        log "Cleanup completed with $cleanup_errors warnings/errors. Check log
 file for details."
    fi
}

# Function to check prerequisites
check_prerequisites() {
    log "Checking prerequisites..."

    # Check AWS CLI
    if ! command -v aws &> /dev/null; then
        log "ERROR: AWS CLI is not installed"
        exit 1
    fi

    # Check AWS credentials
    if ! aws sts get-caller-identity &> /dev/null; then
        log "ERROR: AWS credentials not configured"
        exit 1
    fi
```

```
    # Get caller identity
    CALLER_IDENTITY=$(aws sts get-caller-identity --output text --query
 'Account')
    log "AWS Account: $CALLER_IDENTITY"
    log "AWS Region: $AWS_REGION"

    # Check for default VPC
    DEFAULT_VPC=$(aws ec2 describe-vpcs --filters "Name=is-default,Values=true"
 --query 'Vpcs[0].VpcId' --output text)
    if [[ "$DEFAULT_VPC" == "None" ]]; then
        log "ERROR: No default VPC found. Please create a VPC first."
        exit 1
    fi
    log "Using default VPC: $DEFAULT_VPC"

    # Get default subnet
    DEFAULT_SUBNET=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=
$DEFAULT_VPC" "Name=default-for-az,Values=true" --query 'Subnets[0].SubnetId' --
output text)
    if [[ "$DEFAULT_SUBNET" == "None" ]]; then
        log "ERROR: No default subnet found"
        exit 1
    fi
    log "Using default subnet: $DEFAULT_SUBNET"

    log "Prerequisites check completed successfully"
}

# Function to create ECS cluster
create_cluster() {
    log "Creating ECS cluster: $CLUSTER_NAME"

    CLUSTER_ARN=$(aws ecs create-cluster --cluster-name "$CLUSTER_NAME" --
tags key=project,value=$PROJECT_TAG key=tutorial,value=$TUTORIAL_TAG --query
 'cluster.clusterArn' --output text)

    if [[ -z "$CLUSTER_ARN" ]]; then
        log "ERROR: Failed to create cluster"
        exit 1
    fi

    log "Created cluster: $CLUSTER_ARN"
    CREATED_RESOURCES+=("ECS Cluster: $CLUSTER_NAME")
}
```

```
# Function to create key pair
create_key_pair() {
    log "Creating EC2 key pair: $KEY_PAIR_NAME"

    # FIXED: Set secure umask before key creation
    umask 077
    aws ec2 create-key-pair --key-name "$KEY_PAIR_NAME" --query 'KeyMaterial' --
output text > "${KEY_PAIR_NAME}.pem"
    chmod 400 "${KEY_PAIR_NAME}.pem"
    umask 022  # Reset umask

    log "Created key pair: $KEY_PAIR_NAME"
    CREATED_RESOURCES+=("EC2 Key Pair: $KEY_PAIR_NAME")

    aws ec2 create-tags --resources "$KEY_PAIR_NAME" --tags Key=project,Value=
$PROJECT_TAG Key=tutorial,Value=$TUTORIAL_TAG 2>>"$LOG_FILE" || log "WARNING:
 Failed to tag key pair"
}

# Function to create security group
create_security_group() {
    log "Creating security group: $SECURITY_GROUP_NAME"

    SECURITY_GROUP_ID=$(aws ec2 create-security-group \
        --group-name "$SECURITY_GROUP_NAME" \
        --description "ECS tutorial security group" \
        --vpc-id "$DEFAULT_VPC" \
        --tag-specifications "ResourceType=security-
group,Tags=[{Key=project,Value=$PROJECT_TAG},{Key=tutorial,Value=$TUTORIAL_TAG}]"
 \
        --query 'GroupId' --output text)

    if [[ -z "$SECURITY_GROUP_ID" ]]; then
        log "ERROR: Failed to create security group"
        exit 1
    fi

    # Add HTTP access rule for nginx web server with restricted CIDR
    # SECURITY FIX: Restrict access to specific CIDR if available, otherwise
 document the risk
    log "WARNING: Security group allows HTTP (port 80) from 0.0.0.0/0 - restrict
 this in production"
    aws ec2 authorize-security-group-ingress \
```

```
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 80 \
        --cidr "0.0.0.0/0"

    log "Created security group: $SECURITY_GROUP_ID"
    log "Added HTTP access on port 80"
    CREATED_RESOURCES+=("Security Group: $SECURITY_GROUP_ID")
}

# Function to get ECS optimized AMI
get_ecs_ami() {
    log "Getting ECS-optimized AMI ID..."

    ECS_AMI_ID=$(aws ssm get-parameters \
        --names /aws/service/ecs/optimized-ami/amazon-linux-2/recommended \
        --query 'Parameters[0].Value' --output text | jq -r '.image_id')

    if [[ -z "$ECS_AMI_ID" ]]; then
        log "ERROR: Failed to get ECS-optimized AMI ID"
        exit 1
    fi

    log "ECS-optimized AMI ID: $ECS_AMI_ID"
}

# Function to create IAM role for ECS instance (if it doesn't exist)
ensure_ecs_instance_role() {
    log "Checking for ecsInstanceRole..."

    if ! aws iam get-role --role-name ecsInstanceRole &> /dev/null; then
        log "Creating ecsInstanceRole..."

        # Create trust policy
        cat > ecs-instance-trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
```

```
    }
  ]
}
EOF

        # SECURITY FIX: Validate JSON before using
        if ! jq empty ecs-instance-trust-policy.json 2>/dev/null; then
            log "ERROR: Invalid JSON in trust policy"
            rm -f ecs-instance-trust-policy.json
            exit 1
        fi

        # Create role
        aws iam create-role \
            --role-name ecsInstanceRole \
            --assume-role-policy-document file://ecs-instance-trust-policy.json

        aws iam tag-role --role-name ecsInstanceRole --tags Key=project,Value=
$PROJECT_TAG Key=tutorial,Value=$TUTORIAL_TAG

        # Attach managed policy
        aws iam attach-role-policy \
            --role-name ecsInstanceRole \
            --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role

        # Create instance profile
        aws iam create-instance-profile --instance-profile-name ecsInstanceRole

        # Add role to instance profile
        aws iam add-role-to-instance-profile \
            --instance-profile-name ecsInstanceRole \
            --role-name ecsInstanceRole

        # FIXED: Enhanced wait for role to be ready
        log "Waiting for IAM role to be ready..."
        aws iam wait role-exists --role-name ecsInstanceRole
        sleep 30  # Additional buffer for eventual consistency

        rm -f ecs-instance-trust-policy.json
        log "Created ecsInstanceRole"
        CREATED_RESOURCES+=("IAM Role: ecsInstanceRole")
    else
        log "ecsInstanceRole already exists"
```

```
    fi
}

# Function to launch container instance
launch_container_instance() {
    log "Launching ECS container instance..."

    # Create user data script
    cat > ecs-user-data.sh << EOF
#!/bin/bash
echo ECS_CLUSTER=$CLUSTER_NAME >> /etc/ecs/ecs.config
EOF

    # SECURITY FIX: Validate user data script before use
    if ! bash -n ecs-user-data.sh 2>/dev/null; then
        log "ERROR: Invalid user data script"
        rm -f ecs-user-data.sh
        exit 1
    fi

    INSTANCE_ID=$(aws ec2 run-instances \
        --image-id "$ECS_AMI_ID" \
        --instance-type t3.micro \
        --key-name "$KEY_PAIR_NAME" \
        --security-group-ids "$SECURITY_GROUP_ID" \
        --subnet-id "$DEFAULT_SUBNET" \
        --iam-instance-profile Name=ecsInstanceRole \
        --user-data file://ecs-user-data.sh \
        --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=ecs-
tutorial-instance},{Key=project,Value=$PROJECT_TAG},{Key=tutorial,Value=
$TUTORIAL_TAG}]" \
        --monitoring Enabled=false \
        --metadata-options HttpTokens=required,HttpPutResponseHopLimit=1 \
        --query 'Instances[0].InstanceId' --output text)

    if [[ -z "$INSTANCE_ID" ]]; then
        log "ERROR: Failed to launch EC2 instance"
        rm -f ecs-user-data.sh
        exit 1
    fi

    log "Launched EC2 instance: $INSTANCE_ID"
    log "Instance metadata options: IMDSv2 enforced with hop limit 1"
    CREATED_RESOURCES+=("EC2 Instance: $INSTANCE_ID")
```

```
    # Wait for instance to be running
    log "Waiting for instance to be running..."
    aws ec2 wait instance-running --instance-ids "$INSTANCE_ID"

    # Wait for ECS agent to register
    log "Waiting for ECS agent to register with cluster..."
    local max_attempts=30
    local attempt=0

    while [[ $attempt -lt $max_attempts ]]; do
        CONTAINER_INSTANCES=$(aws ecs list-container-instances --cluster
 "$CLUSTER_NAME" --query 'containerInstanceArns' --output text)
        if [[ -n "$CONTAINER_INSTANCES" && "$CONTAINER_INSTANCES" != "None" ]];
 then
            log "Container instance registered successfully"
            break
        fi

        attempt=$((attempt + 1))
        log "Waiting for container instance registration... (attempt $attempt/
$max_attempts)"
        sleep 10
    done

    if [[ $attempt -eq $max_attempts ]]; then
        log "ERROR: Container instance failed to register within expected time"
        rm -f ecs-user-data.sh
        exit 1
    fi

    rm -f ecs-user-data.sh
}

# Function to register task definition
register_task_definition() {
    log "Creating task definition..."

    # Create nginx task definition JSON matching the tutorial
    cat > task-definition.json << 'EOF'
{
    "family": "TASK_FAMILY_PLACEHOLDER",
    "containerDefinitions": [
        {
```

```
            "name": "nginx",
            "image": "public.ecr.aws/docker/library/nginx:latest",
            "cpu": 256,
            "memory": 512,
            "essential": true,
            "portMappings": [
                {
                    "containerPort": 80,
                    "hostPort": 80,
                    "protocol": "tcp"
                }
            ],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/nginx-task",
                    "awslogs-region": "REGION_PLACEHOLDER",
                    "awslogs-stream-prefix": "ecs"
                }
            }
        }
    ],
    "requiresCompatibilities": ["EC2"],
    "networkMode": "bridge",
    "tags": [
        {
            "key": "project",
            "value": "PROJECT_TAG_PLACEHOLDER"
        },
        {
            "key": "tutorial",
            "value": "TUTORIAL_TAG_PLACEHOLDER"
        }
    ]
}
EOF

    # Replace placeholders securely
    sed -i "s|TASK_FAMILY_PLACEHOLDER|$TASK_FAMILY|g" task-definition.json
    sed -i "s|REGION_PLACEHOLDER|$AWS_REGION|g" task-definition.json
    sed -i "s|PROJECT_TAG_PLACEHOLDER|$PROJECT_TAG|g" task-definition.json
    sed -i "s|TUTORIAL_TAG_PLACEHOLDER|$TUTORIAL_TAG|g" task-definition.json

    # FIXED: Validate JSON before registration
```

```
    if ! jq empty task-definition.json 2>/dev/null; then
        log "ERROR: Invalid JSON in task definition"
        rm -f task-definition.json
        exit 1
    fi

    TASK_DEFINITION_ARN=$(aws ecs register-task-definition \
        --cli-input-json file://task-definition.json \
        --query 'taskDefinition.taskDefinitionArn' --output text)

    if [[ -z "$TASK_DEFINITION_ARN" ]]; then
        log "ERROR: Failed to register task definition"
        rm -f task-definition.json
        exit 1
    fi

    log "Registered task definition: $TASK_DEFINITION_ARN"
    log "Task definition includes CloudWatch Logs configuration for monitoring"
    CREATED_RESOURCES+=("Task Definition: $TASK_DEFINITION_ARN")

    rm -f task-definition.json
}

# Function to create service
create_service() {
    log "Creating ECS service..."

    SERVICE_ARN=$(aws ecs create-service \
        --cluster "$CLUSTER_NAME" \
        --service-name "$SERVICE_NAME" \
        --task-definition "$TASK_FAMILY" \
        --desired-count 1 \
        --tags key=project,value=$PROJECT_TAG key=tutorial,value=$TUTORIAL_TAG \
        --query 'service.serviceArn' --output text)

    if [[ -z "$SERVICE_ARN" ]]; then
        log "ERROR: Failed to create service"
        exit 1
    fi

    log "Created service: $SERVICE_ARN"
    CREATED_RESOURCES+=("ECS Service: $SERVICE_NAME")

    # Wait for service to be stable
```

```
    log "Waiting for service to be stable..."
    aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$SERVICE_NAME"

    log "Service is now stable and running"

    # Get the task ARN for monitoring
    TASK_ARN=$(aws ecs list-tasks --cluster "$CLUSTER_NAME" --service-name
 "$SERVICE_NAME" --query 'taskArns[0]' --output text)
    if [[ -n "$TASK_ARN" && "$TASK_ARN" != "None" ]]; then
        log "Service task: $TASK_ARN"
        CREATED_RESOURCES+=("ECS Task: $TASK_ARN")
    fi
}

# Function to demonstrate monitoring and testing
demonstrate_monitoring() {
    log "Demonstrating monitoring capabilities..."

    # List services
    log "Listing services in cluster:"
    aws ecs list-services --cluster "$CLUSTER_NAME" --output table

    # Describe service
    log "Service details:"
    aws ecs describe-services --cluster "$CLUSTER_NAME" --
services "$SERVICE_NAME" --output table --query 'services[0].
{ServiceName:serviceName,Status:status,RunningCount:runningCount,DesiredCount:desiredCoun

    # List tasks
    log "Listing tasks in service:"
    aws ecs list-tasks --cluster "$CLUSTER_NAME" --service-name "$SERVICE_NAME"
 --output table

    # Describe task
    if [[ -n "$TASK_ARN" && "$TASK_ARN" != "None" ]]; then
        log "Task details:"
        aws ecs describe-tasks --cluster "$CLUSTER_NAME"
 --tasks "$TASK_ARN" --output table --query 'tasks[0].
{TaskArn:taskArn,LastStatus:lastStatus,DesiredStatus:desiredStatus,CreatedAt:createdAt}'
    fi

    # List container instances
    log "Container instances in cluster:"
```

```
    aws ecs list-container-instances --cluster "$CLUSTER_NAME" --output table

    # Describe container instance
    CONTAINER_INSTANCE_ARN=$(aws ecs list-container-instances --cluster
 "$CLUSTER_NAME" --query 'containerInstanceArns[0]' --output text)
    if [[ -n "$CONTAINER_INSTANCE_ARN" && "$CONTAINER_INSTANCE_ARN" != "None" ]];
 then
        log "Container instance details:"
        aws ecs describe-container-instances --cluster "$CLUSTER_NAME"
 --container-instances "$CONTAINER_INSTANCE_ARN" --output table --query
 'containerInstances[0].
{Arn:containerInstanceArn,Status:status,RunningTasks:runningTasksCount,PendingTasks:pendi
    fi

    # Test the nginx web server
    log "Testing nginx web server..."
    PUBLIC_IP=$(aws ec2 describe-instances --instance-ids "$INSTANCE_ID" --query
 'Reservations[0].Instances[0].PublicIpAddress' --output text)

    if [[ -n "$PUBLIC_IP" && "$PUBLIC_IP" != "None" ]]; then
        log "Container instance public IP: $PUBLIC_IP"
        log "Testing HTTP connection to nginx..."

        # Wait a moment for nginx to be fully ready
        sleep 10

        if curl -s --connect-timeout 10 "http://$PUBLIC_IP" | grep -q "Welcome to
 nginx"; then
            log "SUCCESS: Nginx web server is responding correctly"
            echo ""
            echo "============================================="
            echo "WEB SERVER TEST SUCCESSFUL"
            echo "============================================="
            echo "You can access your nginx web server at: http://$PUBLIC_IP"
            echo "The nginx welcome page should be visible in your browser."
        else
            log "WARNING: Nginx web server may not be fully ready yet. Try
 accessing http://$PUBLIC_IP in a few minutes."
        fi
    else
        log "WARNING: Could not retrieve public IP address"
    fi
}
```

```
# Main execution
main() {
    log "Starting ECS EC2 Launch Type Tutorial (UPDATED VERSION)"
    log "Log file: $LOG_FILE"
    log "Security improvements: IMDSv2 enforced, JSON validation, input
 sanitization, CloudWatch Logs configured"

    check_prerequisites
    create_cluster
    create_key_pair
    create_security_group
    get_ecs_ami
    ensure_ecs_instance_role
    launch_container_instance
    register_task_definition
    create_service
    demonstrate_monitoring

    log "Tutorial completed successfully!"

    echo ""
    echo "==========================================="
    echo "TUTORIAL COMPLETED SUCCESSFULLY"
    echo "==========================================="
    echo "Resources created:"
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "  - $resource"
    done
    echo ""
    echo "The nginx service will continue running and maintain the desired task
 count."
    echo "You can monitor the service status using:"
    echo "  aws ecs describe-services --cluster $CLUSTER_NAME --services
 $SERVICE_NAME"
    echo ""
    if [[ -n "${PUBLIC_IP:-}" ]]; then
        echo "Access your web server at: http://$PUBLIC_IP"
        echo ""
    fi
    echo "==========================================="
    echo "CLEANUP CONFIRMATION"
    echo "==========================================="
    log "Auto-confirming cleanup - proceeding with resource cleanup"
    cleanup_resources
```

```
    log "All resources have been cleaned up"
}

# Run main function
main "$@"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AddRoleToInstanceProfile
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress
  - CreateCluster
  - CreateInstanceProfile
  - CreateKeyPair
  - CreateRole
  - CreateSecurityGroup
  - CreateService
  - DeleteCluster
  - DeleteKeyPair
  - DeleteSecurityGroup
  - DeleteService
  - DeregisterTaskDefinition
  - DescribeContainerInstances
  - DescribeInstances
  - DescribeServices
  - DescribeSubnets
  - DescribeTasks
  - DescribeVpcs
  - GetCallerIdentity
  - GetParameters
  - GetRole
  - ListContainerInstances

- ListServices
- ListTasks
- RegisterTaskDefinition
- RunInstances
- StopTask
- TerminateInstances
- UpdateService
- Attendi (Amazon EC2)
- Attendi (Amazon ECS)
- Aspetta (IAM)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione di un'area di lavoro di monitoraggio gestita

L'esempio di codice seguente mostra come:

- Crea un ruolo IAM per il tuo spazio di lavoro
- Crea uno spazio di lavoro Grafana
- Configura l'autenticazione
- Configura le impostazioni opzionali
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon Managed Grafana Workspace Creation Script
# This script creates an Amazon Managed Grafana workspace and configures it

# Set up logging
LOG_FILE="grafana-workspace-creation.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon Managed Grafana workspace creation script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2

    if echo "$output" | grep -i "error\|exception\|fail" > /dev/null; then
        echo "ERROR: Command '$cmd' failed with output:"
        echo "$output"
        cleanup_on_error
        exit 1
    fi
}

# Function to clean up resources on error
cleanup_on_error() {
    echo "Error encountered. Attempting to clean up resources..."

    if [ -n "$WORKSPACE_ID" ]; then
        echo "Deleting workspace $WORKSPACE_ID..."
        aws grafana delete-workspace --workspace-id "$WORKSPACE_ID"
    fi

    if [ -n "$ROLE_NAME" ]; then
        echo "Detaching policies from role $ROLE_NAME..."
        if [ -n "$POLICY_ARN" ]; then
            aws iam detach-role-policy --role-name "$ROLE_NAME" --policy-arn
 "$POLICY_ARN"
        fi

        echo "Deleting role $ROLE_NAME..."
        aws iam delete-role --role-name "$ROLE_NAME"
```

```
    fi

    if [ -n "$POLICY_ARN" ]; then
        echo "Deleting policy..."
        aws iam delete-policy --policy-arn "$POLICY_ARN"
    fi

    # Clean up JSON files
    rm -f trust-policy.json cloudwatch-policy.json

    echo "Cleanup completed. See $LOG_FILE for details."
}

# Generate a random identifier for resource names
RANDOM_ID=$(openssl rand -hex 4)
WORKSPACE_NAME="GrafanaWorkspace-${RANDOM_ID}"
ROLE_NAME="GrafanaWorkspaceRole-${RANDOM_ID}"

echo "Using workspace name: $WORKSPACE_NAME"
echo "Using role name: $ROLE_NAME"

# Step 1: Get AWS account ID
echo "Getting AWS account ID..."
ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
check_error "$ACCOUNT_ID" "get-caller-identity"
echo "AWS Account ID: $ACCOUNT_ID"

# Step 2: Create IAM role for Grafana workspace
echo "Creating IAM role for Grafana workspace..."

# Create trust policy document
cat > trust-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "grafana.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
EOF

# Create IAM role
ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$ROLE_NAME" \
  --assume-role-policy-document file://trust-policy.json \
  --description "Role for Amazon Managed Grafana workspace")

check_error "$ROLE_OUTPUT" "create-role"
aws iam tag-role --role-name "$ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=amazon-managed-grafana-gs
echo "IAM role created successfully"

# Extract role ARN
ROLE_ARN=$(echo "$ROLE_OUTPUT" | grep -o '"Arn": "[^"]*' | cut -d'"' -f4)
echo "Role ARN: $ROLE_ARN"

# Attach policies to the role
echo "Attaching policies to the role..."

# CloudWatch policy
cat > cloudwatch-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:GetMetricData"
      ],
      "Resource": "*"
    }
  ]
}
EOF

POLICY_OUTPUT=$(aws iam create-policy \
  --policy-name "GrafanaCloudWatchPolicy-${RANDOM_ID}" \
  --policy-document file://cloudwatch-policy.json)
```

```
check_error "$POLICY_OUTPUT" "create-policy"

POLICY_ARN=$(echo "$POLICY_OUTPUT" | grep -o '"Arn": "[^"]*' | cut -d'"' -f4)
echo "CloudWatch policy ARN: $POLICY_ARN"

ATTACH_OUTPUT=$(aws iam attach-role-policy \
  --role-name "$ROLE_NAME" \
  --policy-arn "$POLICY_ARN")

check_error "$ATTACH_OUTPUT" "attach-role-policy"
echo "CloudWatch policy attached to role"

# Step 3: Create the Grafana workspace
echo "Creating Amazon Managed Grafana workspace..."
WORKSPACE_OUTPUT=$(aws grafana create-workspace \
  --workspace-name "$WORKSPACE_NAME" \
  --authentication-providers "SAML" \
  --permission-type "CUSTOMER_MANAGED" \
  --account-access-type "CURRENT_ACCOUNT" \
  --workspace-role-arn "$ROLE_ARN" \
  --workspace-data-sources "CLOUDWATCH" "PROMETHEUS" "XRAY" \
  --grafana-version "10.4" \
  --tags Environment=Development,project=doc-smith,tutorial=amazon-managed-
grafana-gs)

check_error "$WORKSPACE_OUTPUT" "create-workspace"

echo "Workspace creation initiated:"
echo "$WORKSPACE_OUTPUT"

# Extract workspace ID
WORKSPACE_ID=$(echo "$WORKSPACE_OUTPUT" | grep -o '"id": "[^"]*' | cut -d'"' -f4)

if [ -z "$WORKSPACE_ID" ]; then
    echo "ERROR: Failed to extract workspace ID from output"
    exit 1
fi

echo "Workspace ID: $WORKSPACE_ID"

# Step 4: Wait for workspace to become active
echo "Waiting for workspace to become active. This may take several minutes..."
ACTIVE=false
```

```
MAX_ATTEMPTS=30
ATTEMPT=0

while [ $ACTIVE = false ] && [ $ATTEMPT -lt $MAX_ATTEMPTS ]; do
    ATTEMPT=$((ATTEMPT+1))
    echo "Checking workspace status (attempt $ATTEMPT of $MAX_ATTEMPTS)..."

    DESCRIBE_OUTPUT=$(aws grafana describe-workspace --workspace-id
 "$WORKSPACE_ID")
    check_error "$DESCRIBE_OUTPUT" "describe-workspace"

    STATUS=$(echo "$DESCRIBE_OUTPUT" | grep -o '"status": "[^"]*' | cut -d'"' -
f4)
    echo "Current status: $STATUS"

    if [ "$STATUS" = "ACTIVE" ]; then
        ACTIVE=true
        echo "Workspace is now ACTIVE"
    elif [ "$STATUS" = "FAILED" ]; then
        echo "ERROR: Workspace creation failed"
        cleanup_on_error
        exit 1
    else
        echo "Workspace is still being created. Waiting 30 seconds..."
        sleep 30
    fi
done

if [ $ACTIVE = false ]; then
    echo "ERROR: Workspace did not become active within the expected time"
    cleanup_on_error
    exit 1
fi

# Extract workspace endpoint URL
WORKSPACE_URL=$(echo "$DESCRIBE_OUTPUT" | grep -o '"endpoint": "[^"]*' | cut -
d'"' -f4)
echo "Workspace URL: https://$WORKSPACE_URL"

# Step 5: Display workspace information
echo ""
echo "========================================="
echo "WORKSPACE INFORMATION"
echo "========================================="
```

```
echo "Workspace ID: $WORKSPACE_ID"
echo "Workspace URL: https://$WORKSPACE_URL"
echo "Workspace Name: $WORKSPACE_NAME"
echo "IAM Role: $ROLE_NAME"
echo ""
echo "Note: Since SAML authentication is used, you need to configure SAML
 settings"
echo "using the AWS Management Console or the update-workspace-authentication
 command."
echo "========================================="

# Step 6: Prompt for cleanup
echo ""
echo "========================================="
echo "CLEANUP CONFIRMATION"
echo "========================================="
echo "Resources created:"
echo "- Amazon Managed Grafana workspace: $WORKSPACE_ID"
echo "- IAM Role: $ROLE_NAME"
echo "- IAM Policy: GrafanaCloudWatchPolicy-${RANDOM_ID}"
echo ""
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "$CLEANUP_CHOICE" =~ ^[Yy] ]]; then
    echo "Cleaning up resources..."

    echo "Deleting workspace $WORKSPACE_ID..."
    DELETE_OUTPUT=$(aws grafana delete-workspace --workspace-id "$WORKSPACE_ID")
    check_error "$DELETE_OUTPUT" "delete-workspace"

    echo "Waiting for workspace to be deleted..."
    DELETED=false
    ATTEMPT=0

    while [ $DELETED = false ] && [ $ATTEMPT -lt $MAX_ATTEMPTS ]; do
        ATTEMPT=$((ATTEMPT+1))
        echo "Checking deletion status (attempt $ATTEMPT of $MAX_ATTEMPTS)..."

        if aws grafana describe-workspace --workspace-id "$WORKSPACE_ID" 2>&1 |
 grep -i "not found\|does not exist" > /dev/null; then
            DELETED=true
            echo "Workspace has been deleted"
        else
```

```
            echo "Workspace is still being deleted. Waiting 30 seconds..."
            sleep 30
        fi
    done

    if [ $DELETED = false ]; then
        echo "WARNING: Workspace deletion is taking longer than expected. It may
 still be in progress."
    fi

    # Detach policy from role
    echo "Detaching policy from role..."
    aws iam detach-role-policy \
      --role-name "$ROLE_NAME" \
      --policy-arn "$POLICY_ARN"

    # Delete policy
    echo "Deleting IAM policy..."
    aws iam delete-policy \
      --policy-arn "$POLICY_ARN"

    # Delete role
    echo "Deleting IAM role..."
    aws iam delete-role \
      --role-name "$ROLE_NAME"

    # Clean up JSON files
    rm -f trust-policy.json cloudwatch-policy.json

    echo "Cleanup completed"
else
    echo "Skipping cleanup. Resources will remain in your AWS account."
fi

echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreatePolicy
  - CreateRole

- CreateWorkspace
- DeletePolicy
- DeleteRole
- DeleteWorkspace
- DescribeWorkspace
- DetachRolePolicy
- GetCallerIdentity

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione di un dashboard di monitoraggio con il nome della funzione come variabile

L'esempio di codice seguente mostra come:

- Crea un pannello di controllo CloudWatch
- Aggiungi widget metrici Lambda con una variabile con nome di funzione
- Verifica il pannello di controllo
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Script to create a CloudWatch dashboard with Lambda function name as a variable
```

```
# This script creates a CloudWatch dashboard that allows you to switch between
 different Lambda functions

# Set up logging with secure permissions
LOG_FILE="${HOME}/.cloudwatch-dashboard-script.log"
touch "$LOG_FILE" && chmod 600 "$LOG_FILE"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "$(date): Starting CloudWatch dashboard creation script"

# Security: Set strict error handling
set -uo pipefail
trap 'handle_error "Script failed at line $LINENO"' ERR

# Function to handle errors
handle_error() {
    local error_msg="${1:-Unknown error}"
    echo "ERROR: $error_msg" >&2
    echo "Resources created:"
    echo "- CloudWatch Dashboard: LambdaMetricsDashboard"
    echo ""
    echo "=========================================="
    echo "CLEANUP CONFIRMATION"
    echo "=========================================="
    echo "An error occurred. Proceeding with automatic cleanup..."

    echo "Cleaning up resources..."
    aws cloudwatch delete-dashboards --dashboard-names LambdaMetricsDashboard 2>/
dev/null || true

    # Clean up temporary files
    if [ -n "${TEMP_DIR:-}" ] && [ -d "$TEMP_DIR" ]; then
        rm -rf "$TEMP_DIR"
    fi
    rm -f dashboard-body.json

    echo "Cleanup complete."
    exit 1
}

# Security: Validate AWS CLI is installed
if ! command -v aws &> /dev/null; then
    handle_error "AWS CLI is not installed. Please install it and try again."
fi
```

```
# Check if AWS CLI is installed and configured
echo "Checking AWS CLI configuration..."
if ! aws sts get-caller-identity > /dev/null 2>&1; then
    handle_error "AWS CLI is not properly configured. Please configure it with
 'aws configure' and try again."
fi

# Get the current region securely
REGION=$(aws configure get region 2>/dev/null || echo "")
if [ -z "$REGION" ]; then
    REGION="us-east-1"
    echo "No region found in AWS config, defaulting to $REGION"
fi
echo "Using region: $REGION"

# Validate region format
if ! [[ "$REGION" =~ ^[a-z]{2}-[a-z]+-[0-9]{1}$ ]]; then
    handle_error "Invalid AWS region format: $REGION"
fi

# Check if there are any Lambda functions in the account
echo "Checking for Lambda functions..."
LAMBDA_FUNCTIONS=$(aws lambda list-functions --region "$REGION" --query
 "Functions[*].FunctionName" --output text 2>/dev/null || echo "")

if [ -z "$LAMBDA_FUNCTIONS" ]; then
    echo "No Lambda functions found in your account. Creating a simple test
 function..."

    # Create a temporary directory for Lambda function code with secure
 permissions
    TEMP_DIR=$(mktemp -d)
    chmod 700 "$TEMP_DIR"
    trap 'rm -rf "$TEMP_DIR"' EXIT

    # Create a simple Lambda function
    cat > "$TEMP_DIR/index.js" << 'EOF'
exports.handler = async (event) => {
    console.log('Event:', JSON.stringify(event, null, 2));
    return {
        statusCode: 200,
        body: JSON.stringify('Hello from Lambda!'),
    };
```

```
};
EOF

    # Zip the function code
    if ! cd "$TEMP_DIR"; then
        handle_error "Failed to change to temporary directory"
    fi

    if ! zip -q function.zip index.js; then
        handle_error "Failed to create zip file"
    fi

    # Create a role for the Lambda function with restricted trust policy
    ROLE_NAME="LambdaDashboardTestRole-$(date +%s)"
    TRUST_POLICY='{"Version":"2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":
{"Service":"lambda.amazonaws.com"},"Action":"sts:AssumeRole"}]}'

    if ! ROLE_ARN=$(aws iam create-role \
        --role-name "$ROLE_NAME" \
        --assume-role-policy-document "$TRUST_POLICY" \
        --tags Key=project,Value=doc-smith Key=tutorial,Value=cloudwatch-
dynamicdash \
        --query "Role.Arn" \
        --output text 2>/dev/null); then
        handle_error "Failed to create IAM role for Lambda function"
    fi

    echo "Waiting for role to be available..."
    sleep 10

    # Attach basic Lambda execution policy
    if ! aws iam attach-role-policy \
        --role-name "$ROLE_NAME" \
        --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"; then
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
        handle_error "Failed to attach policy to IAM role"
    fi

    # Create the Lambda function
    FUNCTION_NAME="DashboardTestFunction-$(date +%s)"
    if ! aws lambda create-function \
        --function-name "$FUNCTION_NAME" \
```

```
        --runtime nodejs18.x \
        --role "$ROLE_ARN" \
        --handler index.handler \
        --zip-file fileb://function.zip \
        --tags project=doc-smith,tutorial=cloudwatch-dynamicdash \
        --region "$REGION" > /dev/null 2>&1; then
        aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null || true
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
        handle_error "Failed to create Lambda function"
    fi

    # Invoke the function to generate some metrics
    echo "Invoking Lambda function to generate metrics..."
    for i in {1..5}; do
        aws lambda invoke --function-name "$FUNCTION_NAME" --payload '{}' /dev/
null --region "$REGION" > /dev/null 2>&1 || true
        sleep 1
    done

    # Go back to original directory
    cd - > /dev/null

    # Set the function name for the dashboard
    DEFAULT_FUNCTION="$FUNCTION_NAME"
else
    # Use the first Lambda function as default
    DEFAULT_FUNCTION=$(echo "$LAMBDA_FUNCTIONS" | awk '{print $1}')
    echo "Found Lambda functions. Using $DEFAULT_FUNCTION as default."
    FUNCTION_NAME=""
    ROLE_NAME=""
fi

# Create a dashboard with Lambda metrics and a function name variable
echo "Creating CloudWatch dashboard with Lambda function name variable..."

# Create a JSON file for the dashboard body with secure permissions
DASHBOARD_JSON="dashboard-body-$$.json"
touch "$DASHBOARD_JSON" && chmod 600 "$DASHBOARD_JSON"

# Escape special characters in region and function name for JSON
REGION_ESCAPED=$(printf '%s\n' "$REGION" | sed 's:[\/&]:\\&:g')
```

```
FUNCTION_ESCAPED=$(printf '%s\n' "$DEFAULT_FUNCTION" | sed 's:[\/&]:\\&:g')

cat > "$DASHBOARD_JSON" << EOF
{
  "widgets": [
    {
      "type": "metric",
      "x": 0,
      "y": 0,
      "width": 12,
      "height": 6,
      "properties": {
        "metrics": [
          [ "AWS/Lambda", "Invocations", "FunctionName", "\${FunctionName}" ],
          [ ".", "Errors", ".", "." ],
          [ ".", "Throttles", ".", "." ]
        ],
        "view": "timeSeries",
        "stacked": false,
        "region": "$REGION_ESCAPED",
        "title": "Lambda Function Metrics for \${FunctionName}",
        "period": 300
      }
    },
    {
      "type": "metric",
      "x": 0,
      "y": 6,
      "width": 12,
      "height": 6,
      "properties": {
        "metrics": [
          [ "AWS/Lambda", "Duration", "FunctionName", "\${FunctionName}",
 { "stat": "Average" } ]
        ],
        "view": "timeSeries",
        "stacked": false,
        "region": "$REGION_ESCAPED",
        "title": "Duration for \${FunctionName}",
        "period": 300
      }
    },
    {
      "type": "metric",
```

```
      "x": 12,
      "y": 0,
      "width": 12,
      "height": 6,
      "properties": {
        "metrics": [
          [ "AWS/Lambda", "ConcurrentExecutions", "FunctionName", "\
${FunctionName}" ]
        ],
        "view": "timeSeries",
        "stacked": false,
        "region": "$REGION_ESCAPED",
        "title": "Concurrent Executions for \${FunctionName}",
        "period": 300
      }
    }
  ],
  "periodOverride": "auto",
  "variables": [
    {
      "type": "property",
      "id": "FunctionName",
      "property": "FunctionName",
      "label": "Lambda Function",
      "inputType": "select",
      "values": [
        {
          "value": "$FUNCTION_ESCAPED",
          "label": "$FUNCTION_ESCAPED"
        }
      ]
    }
  ]
}
EOF

# Validate JSON before sending
if ! jq empty "$DASHBOARD_JSON" 2>/dev/null; then
    handle_error "Invalid JSON generated for dashboard"
fi

# Create the dashboard using the JSON file
DASHBOARD_NAME="LambdaMetricsDashboard-$(date +%s)"
if ! DASHBOARD_RESULT=$(aws cloudwatch put-dashboard \
```

```
    --dashboard-name "$DASHBOARD_NAME" \
    --dashboard-body file://"$DASHBOARD_JSON" \
    --region "$REGION" 2>&1); then
    # If we created resources, clean them up
    if [ -n "${FUNCTION_NAME:-}" ]; then
        aws lambda delete-function --function-name "$FUNCTION_NAME" --region
 "$REGION" 2>/dev/null || true
        aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null || true
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi
    handle_error "Failed to create CloudWatch dashboard."
fi

# Display any validation messages but continue
if echo "$DASHBOARD_RESULT" | grep -q "DashboardValidationMessages"; then
    echo "Dashboard created with validation messages:"
    echo "$DASHBOARD_RESULT"
    echo "These validation messages are warnings and the dashboard should still
 function."
else
    echo "Dashboard created successfully!"
fi

# Verify the dashboard was created
echo "Verifying dashboard creation..."
if ! DASHBOARD_INFO=$(aws cloudwatch get-dashboard --dashboard-name
 "$DASHBOARD_NAME" --region "$REGION" 2>&1); then
    # If we created resources, clean them up
    if [ -n "${FUNCTION_NAME:-}" ]; then
        aws lambda delete-function --function-name "$FUNCTION_NAME" --region
 "$REGION" 2>/dev/null || true
        aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null || true
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi
    handle_error "Failed to verify dashboard creation."
fi

echo "Dashboard verification successful!"
```

```
echo "Dashboard details:"
echo "$DASHBOARD_INFO"

# List all dashboards to confirm
echo "Listing all dashboards:"
if ! DASHBOARDS=$(aws cloudwatch list-dashboards --region "$REGION" 2>&1); then
    # If we created resources, clean them up
    if [ -n "${FUNCTION_NAME:-}" ]; then
        aws lambda delete-function --function-name "$FUNCTION_NAME" --region
 "$REGION" 2>/dev/null || true
        aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null || true
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi
    handle_error "Failed to list dashboards."
fi
echo "$DASHBOARDS"

# Show instructions for accessing the dashboard
echo ""
echo "Dashboard created successfully! To access it:"
echo "1. Open the CloudWatch console at https://console.aws.amazon.com/
cloudwatch/"
echo "2. In the navigation pane, choose Dashboards"
echo "3. Select $DASHBOARD_NAME"
echo "4. You should see a dropdown menu labeled 'Lambda Function' at the top of
 the dashboard"
echo "5. Use this dropdown to select different Lambda functions and see their
 metrics"
echo ""

# Create a list of resources for cleanup
RESOURCES=("- CloudWatch Dashboard: $DASHBOARD_NAME")
if [ -n "${FUNCTION_NAME:-}" ]; then
    RESOURCES+=("- Lambda Function: $FUNCTION_NAME")
    RESOURCES+=("- IAM Role: $ROLE_NAME")
fi

# Prompt for cleanup with automatic yes
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
```

```
echo "Resources created:"
for resource in "${RESOURCES[@]}"; do
    echo "$resource"
done
echo ""
echo "Proceeding with automatic cleanup..."

CLEANUP_CHOICE="y"

if [[ "${CLEANUP_CHOICE,,}" == "y" ]]; then
    echo "Cleaning up resources..."

    # Delete the dashboard
    if aws cloudwatch delete-dashboards --dashboard-names "$DASHBOARD_NAME" --
region "$REGION" 2>/dev/null; then
        echo "Dashboard deleted successfully."
    else
        echo "WARNING: Failed to delete dashboard. You may need to delete it
 manually."
    fi

    # If we created a Lambda function, delete it and its role
    if [ -n "${FUNCTION_NAME:-}" ]; then
        echo "Deleting Lambda function..."
        if aws lambda delete-function --function-name "$FUNCTION_NAME" --region
 "$REGION" 2>/dev/null; then
            echo "Lambda function deleted successfully."
        else
            echo "WARNING: Failed to delete Lambda function. You may need to
 delete it manually."
        fi

        echo "Detaching role policy..."
        if aws iam detach-role-policy \
            --role-name "$ROLE_NAME" \
            --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>/dev/null; then
            echo "Role policy detached successfully."
        else
            echo "WARNING: Failed to detach role policy. You may need to detach
 it manually."
        fi

        echo "Deleting IAM role..."
```

```
        if aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null; then
            echo "IAM role deleted successfully."
        else
            echo "WARNING: Failed to delete IAM role. You may need to delete it
 manually."
        fi
    fi

    # Clean up the JSON file
    rm -f "$DASHBOARD_JSON"

    echo "Cleanup complete."
fi

echo "Script completed successfully!"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateFunction
  - CreateRole
  - DeleteDashboards
  - DeleteFunction
  - DeleteRole
  - DetachRolePolicy
  - GetCallerIdentity
  - GetDashboard
  - Invoke
  - ListDashboards
  - ListFunctions
  - PutDashboard

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Ottieni un token di sessione che richiede un token MFA con AWS STS utilizzando un AWS SDK

L'esempio di codice seguente mostra come ottenere un token di sessione che richiede un token MFA.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come Centro identità AWS IAM.

- Creare un ruolo IAM che conceda l'autorizzazione per elencare i bucket Amazon S3.
- Creare un utente IAM che abbia il permesso di assumere il ruolo solo quando vengono fornite le credenziali MFA.
- Registrare un dispositivo MFA per l'utente.
- Fornire le credenziali MFA per ottenere un token di sessione e utilizzare le credenziali temporanee per elencare i bucket S3.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

Creare un utente IAM, registrare un dispositivo MFA e creare un ruolo che conceda l'autorizzazione per consentire all'utente di elencare i bucket Amazon S3 solo quando si utilizzano le credenziali MFA.

```
def setup(iam_resource):
    """
    Creates a new user with no permissions.
    Creates a new virtual multi-factor authentication (MFA) device.
    Displays the QR code to seed the device.
```

```
    Asks for two codes from the MFA device.
    Registers the MFA device for the user.
    Creates an access key pair for the user.
    Creates an inline policy for the user that lets the user list Amazon S3
buckets,
    but only when MFA credentials are used.

    Any MFA device that can scan a QR code will work with this demonstration.
    Common choices are mobile apps like LastPass Authenticator,
    Microsoft Authenticator, or Google Authenticator.

    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
resource
                         that has permissions to create users, MFA devices, and
                         policies in the account.
    :return: The newly created user, user key, and virtual MFA device.
    """
    user = iam_resource.create_user(UserName=unique_name("user"))
    print(f"Created user {user.name}.")

    virtual_mfa_device = iam_resource.create_virtual_mfa_device(
        VirtualMFADeviceName=unique_name("mfa")
    )
    print(f"Created virtual MFA device {virtual_mfa_device.serial_number}")

    print(
        f"Showing the QR code for the device. Scan this in the MFA app of your "
        f"choice."
    )
    with open("qr.png", "wb") as qr_file:
        qr_file.write(virtual_mfa_device.qr_code_png)
    webbrowser.open(qr_file.name)

    print(f"Enter two consecutive code from your MFA device.")
    mfa_code_1 = input("Enter the first code: ")
    mfa_code_2 = input("Enter the second code: ")
    user.enable_mfa(
        SerialNumber=virtual_mfa_device.serial_number,
        AuthenticationCode1=mfa_code_1,
        AuthenticationCode2=mfa_code_2,
    )
    os.remove(qr_file.name)
    print(f"MFA device is registered with the user.")
```

```
    user_key = user.create_access_key_pair()
    print(f"Created access key pair for user.")

    print(f"Wait for user to be ready.", end="")
    progress_bar(10)

    user.create_policy(
        PolicyName=unique_name("user-policy"),
        PolicyDocument=json.dumps(
            {
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": "s3:ListAllMyBuckets",
                        "Resource": "arn:aws:s3:::*",
                        "Condition": {"Bool": {"aws:MultiFactorAuthPresent":
 True}},
                    }
                ],
            }
        ),
    )
    print(
        f"Created an inline policy for {user.name} that lets the user list
 buckets, "
        f"but only when MFA credentials are present."
    )

    print("Give AWS time to propagate these new resources and connections.",
 end="")
    progress_bar(10)

    return user, user_key, virtual_mfa_device
```

Recuperare le credenziali di sessione temporanee passando un token MFA e utilizzarle per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_with_session_token_with_mfa(mfa_serial_number, mfa_totp,
 sts_client):
```

```
    """
    Gets a session token with MFA credentials and uses the temporary session
    credentials to list Amazon S3 buckets.

    Requires an MFA device serial number and token.

    :param mfa_serial_number: The serial number of the MFA device. For a virtual
MFA
                              device, this is an Amazon Resource Name (ARN).
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
role.
    """
    if mfa_serial_number is not None:
        response = sts_client.get_session_token(
            SerialNumber=mfa_serial_number, TokenCode=mfa_totp
        )
    else:
        response = sts_client.get_session_token()
    temp_credentials = response["Credentials"]

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )

    print(f"Buckets for the account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

Elimina le risorse create per la demo.

```
def teardown(user, virtual_mfa_device):
    """
    Removes all resources created during setup.

    :param user: The demo user.
    :param role: The demo MFA device.
```

```
    """
    for user_pol in user.policies.all():
        user_pol.delete()
        print("Deleted inline user policy.")
    for key in user.access_keys.all():
        key.delete()
        print("Deleted user's access key.")
    for mfa in user.mfa_devices.all():
        mfa.disassociate()
    virtual_mfa_device.delete()
    user.delete()
    print(f"Deleted {user.name}.")
```

Esegui questo scenario utilizzando le funzioni definite in precedenza.

```
def usage_demo():
    """Drives the demonstration."""
    print("-" * 88)
    print(
        f"Welcome to the AWS Security Token Service assume role demo, "
        f"starring multi-factor authentication (MFA)!"
    )
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    user, user_key, virtual_mfa_device = setup(iam_resource)
    try:
        sts_client = boto3.client(
            "sts", aws_access_key_id=user_key.id,
 aws_secret_access_key=user_key.secret
        )
        try:
            print("Listing buckets without specifying MFA credentials.")
            list_buckets_with_session_token_with_mfa(None, None, sts_client)
        except ClientError as error:
            if error.response["Error"]["Code"] == "AccessDenied":
                print("Got expected AccessDenied error.")
        mfa_totp = input("Enter the code from your registered MFA device: ")
        list_buckets_with_session_token_with_mfa(
            virtual_mfa_device.serial_number, mfa_totp, sts_client
        )
```

```
    finally:
        teardown(user, virtual_mfa_device)
        print("Thanks for watching!")
```

- Per i dettagli sull'API, consulta GetSessionToken AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva alla gestione della configurazione

L'esempio di codice seguente mostra come:

- Crea un bucket Amazon S3
- Creare un argomento Amazon SNS.
- Creare un ruolo IAM per Config
- Configurare il registratore di configurazione Config
- Configura il canale di distribuzione Config
- Avvia il registratore di configurazione
- Verifica la configurazione di Config

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS Config Setup Script (v2)
# This script sets up AWS Config with the AWS CLI

# Error handling
set -e
LOGFILE="aws-config-setup-v2.log"
touch $LOGFILE
exec > >(tee -a $LOGFILE)
exec 2>&1

# Function to handle errors
handle_error() {
    echo "ERROR: An error occurred at line $1"
    echo "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Set trap for error handling
trap 'handle_error $LINENO' ERR

# Function to generate random identifier
generate_random_id() {
    echo $(openssl rand -hex 6)
}

# Function to check if command was successful
check_command() {
    if echo "$1" | grep -i "error" > /dev/null; then
        echo "ERROR: $1"
        return 1
    fi
    return 0
}

# Function to clean up resources
cleanup_resources() {
    if [ -n "$CONFIG_RECORDER_NAME" ]; then
        echo "Stopping configuration recorder..."
        aws configservice stop-configuration-recorder --configuration-recorder-
name "$CONFIG_RECORDER_NAME" 2>/dev/null || true
```

```
    fi

    # Check if we created a new delivery channel before trying to delete it
    if [ -n "$DELIVERY_CHANNEL_NAME" ] && [ "$CREATED_NEW_DELIVERY_CHANNEL" =
 "true" ]; then
        echo "Deleting delivery channel..."
        aws configservice delete-delivery-channel --delivery-channel-name
 "$DELIVERY_CHANNEL_NAME" 2>/dev/null || true
    fi

    if [ -n "$CONFIG_RECORDER_NAME" ] && [ "$CREATED_NEW_CONFIG_RECORDER" =
 "true" ]; then
        echo "Deleting configuration recorder..."
        aws configservice delete-configuration-recorder --configuration-recorder-
name "$CONFIG_RECORDER_NAME" 2>/dev/null || true
    fi

    if [ -n "$ROLE_NAME" ]; then
        if [ -n "$POLICY_NAME" ]; then
            echo "Detaching custom policy from role..."
            aws iam delete-role-policy --role-name "$ROLE_NAME" --policy-name
 "$POLICY_NAME" 2>/dev/null || true
        fi

        if [ -n "$MANAGED_POLICY_ARN" ]; then
            echo "Detaching managed policy from role..."
            aws iam detach-role-policy --role-name "$ROLE_NAME" --policy-arn
 "$MANAGED_POLICY_ARN" 2>/dev/null || true
        fi

        echo "Deleting IAM role..."
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi

    if [ -n "$SNS_TOPIC_ARN" ]; then
        echo "Deleting SNS topic..."
        aws sns delete-topic --topic-arn "$SNS_TOPIC_ARN" 2>/dev/null || true
    fi

    if [ -n "$S3_BUCKET_NAME" ]; then
        echo "Emptying S3 bucket..."
        aws s3 rm "s3://$S3_BUCKET_NAME" --recursive 2>/dev/null || true

        echo "Deleting S3 bucket..."
```

```
        if [ "$BUCKET_IS_SHARED" = "false" ]; then
            aws s3api delete-bucket --bucket "$S3_BUCKET_NAME" 2>/dev/null ||
 true
        fi
    fi
}

# Function to display created resources
display_resources() {
    echo ""
    echo "========================================="
    echo "CREATED RESOURCES"
    echo "========================================="
    echo "S3 Bucket: $S3_BUCKET_NAME"
    echo "SNS Topic ARN: $SNS_TOPIC_ARN"
    echo "IAM Role: $ROLE_NAME"
    if [ "$CREATED_NEW_CONFIG_RECORDER" = "true" ]; then
        echo "Configuration Recorder: $CONFIG_RECORDER_NAME (newly created)"
    else
        echo "Configuration Recorder: $CONFIG_RECORDER_NAME (existing)"
    fi
    if [ "$CREATED_NEW_DELIVERY_CHANNEL" = "true" ]; then
        echo "Delivery Channel: $DELIVERY_CHANNEL_NAME (newly created)"
    else
        echo "Delivery Channel: $DELIVERY_CHANNEL_NAME (existing)"
    fi
    echo "========================================="
}

# Get AWS account ID
echo "Getting AWS account ID..."
ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
if [ -z "$ACCOUNT_ID" ]; then
    echo "ERROR: Failed to get AWS account ID"
    exit 1
fi
echo "AWS Account ID: $ACCOUNT_ID"

# Generate random identifier for resources
RANDOM_ID=$(generate_random_id)
echo "Generated random identifier: $RANDOM_ID"

# Step 1: Create an S3 bucket
# Check for shared prereq bucket
```

```
PREREQ_BUCKET=$(aws cloudformation describe-stacks --stack-name tutorial-prereqs-
bucket \
    --query 'Stacks[0].Outputs[?OutputKey==`BucketName`].OutputValue' --output
 text 2>/dev/null)
if [ -n "$PREREQ_BUCKET" ] && [ "$PREREQ_BUCKET" != "None" ]; then
    S3_BUCKET_NAME="$PREREQ_BUCKET"
    BUCKET_IS_SHARED=true
    echo "Using shared bucket: $S3_BUCKET_NAME"
else
    BUCKET_IS_SHARED=false
    S3_BUCKET_NAME="configservice-${RANDOM_ID}"
    echo "Creating S3 bucket: $S3_BUCKET_NAME"
fi

# Get the current region
AWS_REGION=$(aws configure get region)
if [ -z "$AWS_REGION" ]; then
    AWS_REGION="us-east-1"  # Default to us-east-1 if no region is configured
fi
echo "Using AWS Region: $AWS_REGION"

# Create bucket with appropriate command based on region
if [ "$BUCKET_IS_SHARED" = "false" ]; then
    if [ "$AWS_REGION" = "us-east-1" ]; then
        BUCKET_RESULT=$(aws s3api create-bucket --bucket "$S3_BUCKET_NAME")
    else
        BUCKET_RESULT=$(aws s3api create-bucket --bucket "$S3_BUCKET_NAME" --
create-bucket-configuration LocationConstraint="$AWS_REGION")
    fi
    check_command "$BUCKET_RESULT"
    echo "S3 bucket created: $S3_BUCKET_NAME"

    aws s3api put-bucket-tagging --bucket "$S3_BUCKET_NAME" --tagging
 'TagSet=[{Key=project,Value=doc-smith},{Key=tutorial,Value=aws-config-gs}]'
    echo "Tags applied to S3 bucket"
else
    echo "Using shared bucket: $S3_BUCKET_NAME (skipping creation)"
fi

# Block public access for the bucket
aws s3api put-public-access-block \
    --bucket "$S3_BUCKET_NAME" \
    --public-access-block-configuration
 "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets
```

```
echo "Public access blocked for bucket"

# Step 2: Create an SNS topic
TOPIC_NAME="config-topic-${RANDOM_ID}"
echo "Creating SNS topic: $TOPIC_NAME"
SNS_RESULT=$(aws sns create-topic --name "$TOPIC_NAME" --tags
 Key=project,Value=doc-smith Key=tutorial,Value=aws-config-gs)
check_command "$SNS_RESULT"
SNS_TOPIC_ARN=$(echo "$SNS_RESULT" | grep -o 'arn:aws:sns:[^"]*')
echo "SNS topic created: $SNS_TOPIC_ARN"

# Step 3: Create an IAM role for AWS Config
ROLE_NAME="config-role-${RANDOM_ID}"
POLICY_NAME="config-delivery-permissions"
MANAGED_POLICY_ARN="arn:aws:iam::aws:policy/service-role/AWS_ConfigRole"

echo "Creating trust policy document..."
cat > config-trust-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "config.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

echo "Creating IAM role: $ROLE_NAME"
ROLE_RESULT=$(aws iam create-role --role-name "$ROLE_NAME" --assume-role-policy-
document file://config-trust-policy.json)
check_command "$ROLE_RESULT"
ROLE_ARN=$(echo "$ROLE_RESULT" | grep -o 'arn:aws:iam::[^"]*' | head -1)
echo "IAM role created: $ROLE_ARN"

aws iam tag-role --role-name "$ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-config-gs
echo "Tags applied to IAM role"

echo "Attaching AWS managed policy to role..."
```

```
ATTACH_RESULT=$(aws iam attach-role-policy --role-name "$ROLE_NAME" --policy-arn
 "$MANAGED_POLICY_ARN")
check_command "$ATTACH_RESULT"
echo "AWS managed policy attached"

echo "Creating custom policy document for S3 and SNS access..."
cat > config-delivery-permissions.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::${S3_BUCKET_NAME}/AWSLogs/${ACCOUNT_ID}/*",
      "Condition": {
        "StringLike": {
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketAcl"
      ],
      "Resource": "arn:aws:s3:::${S3_BUCKET_NAME}"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": "${SNS_TOPIC_ARN}"
    }
  ]
}
EOF

echo "Attaching custom policy to role..."
POLICY_RESULT=$(aws iam put-role-policy --role-name "$ROLE_NAME" --policy-name
 "$POLICY_NAME" --policy-document file://config-delivery-permissions.json)
check_command "$POLICY_RESULT"
```

```
echo "Custom policy attached"

# Wait for IAM role to propagate
echo "Waiting for IAM role to propagate (15 seconds)..."
sleep 15

# Step 4: Check if configuration recorder already exists
CONFIG_RECORDER_NAME="default"
CREATED_NEW_CONFIG_RECORDER="false"

echo "Checking for existing configuration recorder..."
EXISTING_RECORDERS=$(aws configservice describe-configuration-recorders 2>/dev/
null || echo "")
if echo "$EXISTING_RECORDERS" | grep -q "name"; then
    echo "Configuration recorder already exists. Will update it."
    # Get the name of the existing recorder
    CONFIG_RECORDER_NAME=$(echo "$EXISTING_RECORDERS" | grep -o '"name": "[^"]*"'
 | head -1 | cut -d'"' -f4)
    echo "Using existing configuration recorder: $CONFIG_RECORDER_NAME"
else
    echo "No existing configuration recorder found. Will create a new one."
    CREATED_NEW_CONFIG_RECORDER="true"
fi

echo "Creating configuration recorder configuration..."
cat > configurationRecorder.json << EOF
{
  "name": "${CONFIG_RECORDER_NAME}",
  "roleARN": "${ROLE_ARN}",
  "recordingMode": {
    "recordingFrequency": "CONTINUOUS"
  }
}
EOF

echo "Creating recording group configuration..."
cat > recordingGroup.json << EOF
{
  "allSupported": true,
  "includeGlobalResourceTypes": true
}
EOF

echo "Setting up configuration recorder..."
```

```
RECORDER_RESULT=$(aws configservice put-configuration-recorder --configuration-
recorder file://configurationRecorder.json --recording-group file://
recordingGroup.json)
check_command "$RECORDER_RESULT"
echo "Configuration recorder set up"

if [ "$CREATED_NEW_CONFIG_RECORDER" = "true" ]; then
    aws configservice tag-resource --resource-arn "arn:aws:config:
${AWS_REGION}:${ACCOUNT_ID}:config-recorder/${CONFIG_RECORDER_NAME}" --tags
 Key=project,Value=doc-smith Key=tutorial,Value=aws-config-gs
    echo "Tags applied to configuration recorder"
fi

# Step 5: Check if delivery channel already exists
DELIVERY_CHANNEL_NAME="default"
CREATED_NEW_DELIVERY_CHANNEL="false"

echo "Checking for existing delivery channel..."
EXISTING_CHANNELS=$(aws configservice describe-delivery-channels 2>/dev/null ||
 echo "")
if echo "$EXISTING_CHANNELS" | grep -q "name"; then
    echo "Delivery channel already exists."
    # Get the name of the existing channel
    DELIVERY_CHANNEL_NAME=$(echo "$EXISTING_CHANNELS" | grep -o '"name": "[^"]*"'
 | head -1 | cut -d'"' -f4)
    echo "Using existing delivery channel: $DELIVERY_CHANNEL_NAME"

    # Update the existing delivery channel
    echo "Creating delivery channel configuration for update..."
    cat > deliveryChannel.json << EOF
{
  "name": "${DELIVERY_CHANNEL_NAME}",
  "s3BucketName": "${S3_BUCKET_NAME}",
  "snsTopicARN": "${SNS_TOPIC_ARN}",
  "configSnapshotDeliveryProperties": {
    "deliveryFrequency": "Six_Hours"
  }
}
EOF

    echo "Updating delivery channel..."
    CHANNEL_RESULT=$(aws configservice put-delivery-channel --delivery-channel
 file://deliveryChannel.json)
    check_command "$CHANNEL_RESULT"
```

```
    echo "Delivery channel updated"
else
    echo "No existing delivery channel found. Will create a new one."
    CREATED_NEW_DELIVERY_CHANNEL="true"

    echo "Creating delivery channel configuration..."
    cat > deliveryChannel.json << EOF
{
  "name": "${DELIVERY_CHANNEL_NAME}",
  "s3BucketName": "${S3_BUCKET_NAME}",
  "snsTopicARN": "${SNS_TOPIC_ARN}",
  "configSnapshotDeliveryProperties": {
    "deliveryFrequency": "Six_Hours"
  }
}
EOF

    echo "Creating delivery channel..."
    CHANNEL_RESULT=$(aws configservice put-delivery-channel --delivery-channel
 file://deliveryChannel.json)
    check_command "$CHANNEL_RESULT"
    echo "Delivery channel created"

    aws configservice tag-resource --resource-arn "arn:aws:config:
${AWS_REGION}:${ACCOUNT_ID}:delivery-channel/${DELIVERY_CHANNEL_NAME}" --tags
 Key=project,Value=doc-smith Key=tutorial,Value=aws-config-gs
    echo "Tags applied to delivery channel"
fi

# Step 6: Start the configuration recorder
echo "Checking configuration recorder status..."
RECORDER_STATUS=$(aws configservice describe-configuration-recorder-status 2>/
dev/null || echo "")
if echo "$RECORDER_STATUS" | grep -q '"recording": true'; then
    echo "Configuration recorder is already running."
else
    echo "Starting configuration recorder..."
    START_RESULT=$(aws configservice start-configuration-recorder --
configuration-recorder-name "$CONFIG_RECORDER_NAME")
    check_command "$START_RESULT"
    echo "Configuration recorder started"
fi

# Step 7: Verify the AWS Config setup
```

```
echo "Verifying delivery channel..."
VERIFY_CHANNEL=$(aws configservice describe-delivery-channels)
check_command "$VERIFY_CHANNEL"
echo "$VERIFY_CHANNEL"

echo "Verifying configuration recorder..."
VERIFY_RECORDER=$(aws configservice describe-configuration-recorders)
check_command "$VERIFY_RECORDER"
echo "$VERIFY_RECORDER"

echo "Verifying configuration recorder status..."
VERIFY_STATUS=$(aws configservice describe-configuration-recorder-status)
check_command "$VERIFY_STATUS"
echo "$VERIFY_STATUS"

# Display created resources
display_resources

# Ask if user wants to clean up resources
echo ""
echo "========================================="
echo "CLEANUP CONFIRMATION"
echo "========================================="
echo "Do you want to clean up all created resources? (y/n): "
CLEANUP_CHOICE='y'

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    echo "Cleaning up resources..."
    cleanup_resources
    echo "Cleanup completed."
else
    echo "Resources will not be cleaned up. You can manually clean them up
 later."
fi

echo "Script completed successfully!"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateBucket
  - CreateRole

- CreateTopic
- DeleteBucket
- DeleteConfigurationRecorder
- DeleteDeliveryChannel
- DeleteRole
- DeleteRolePolicy
- DeleteTopic
- DescribeConfigurationRecorderStatus
- DescribeConfigurationRecorders
- DescribeDeliveryChannels
- DetachRolePolicy
- GetCallerIdentity
- PutConfigurationRecorder
- PutDeliveryChannel
- PutPublicAccessBlock
- PutRolePolicy
- Rm
- StartConfigurationRecorder
- StopConfigurationRecorder

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva ai registri dei container

L'esempio di codice seguente mostra come:

- Creazione di un'immagine Docker
- Crea un repository Amazon ECR
- Delete resources (Elimina risorse)

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon ECR Getting Started Script
# This script demonstrates the lifecycle of a Docker image in Amazon ECR

# Set up logging
LOG_FILE="ecr-tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "================================================="
echo "Amazon ECR Getting Started Tutorial"
echo "================================================="
echo "This script will:"
echo "1. Create a Docker image"
echo "2. Create an Amazon ECR repository"
echo "3. Authenticate to Amazon ECR"
echo "4. Push the image to Amazon ECR"
echo "5. Pull the image from Amazon ECR"
echo "6. Clean up resources (optional)"
echo "================================================="

# Check prerequisites
echo "Checking prerequisites..."

# Check if AWS CLI is installed
if ! command -v aws &> /dev/null; then
    echo "ERROR: AWS CLI is not installed. Please install it before running this
 script."
    echo "Visit https://docs.aws.amazon.com/cli/latest/userguide/install-
cliv2.html for installation instructions."
    exit 1
```

```
fi

# Check if AWS CLI is configured
if ! aws sts get-caller-identity &> /dev/null; then
    echo "ERROR: AWS CLI is not configured properly. Please run 'aws configure'
 to set up your credentials."
    exit 1
fi

# Check if Docker is installed
if ! command -v docker &> /dev/null; then
    echo "ERROR: Docker is not installed. Please install Docker before running
 this script."
    echo "Visit https://docs.docker.com/get-docker/ for installation
 instructions."
    exit 1
fi

# Check if Docker daemon is running
if ! docker info &> /dev/null; then
    echo "ERROR: Docker daemon is not running. Please start Docker and try
 again."
    exit 1
fi

echo "All prerequisites met."

# Initialize variables
REPO_URI=""
TIMEOUT_CMD="timeout 300"  # 5-minute timeout for long-running commands

# Function to handle errors
handle_error() {
    echo "ERROR: $1"
    echo "Check the log file for details: $LOG_FILE"

    echo "================================================="
    echo "Resources created:"
    echo "- Docker image: hello-world (local)"
    if [ -n "$REPO_URI" ]; then
        echo "- ECR Repository: hello-repository"
        echo "- ECR Image: $REPO_URI:latest"
    fi
    echo "================================================="
```

```
    echo "Attempting to clean up resources..."
    cleanup
    exit 1
}

# Function to clean up resources
cleanup() {
    echo "================================================="
    echo "Cleaning up resources..."

    # Delete the image from ECR if it exists
    if [ -n "$REPO_URI" ]; then
        echo "Deleting image from ECR repository..."
        aws ecr batch-delete-image --repository-name hello-repository --image-ids
 imageTag=latest || echo "Failed to delete image, it may not exist or may have
 already been deleted."
    fi

    # Delete the ECR repository if it exists
    if [ -n "$REPO_URI" ]; then
        echo "Deleting ECR repository..."
        aws ecr delete-repository --repository-name hello-repository --force ||
 echo "Failed to delete repository, it may not exist or may have already been
 deleted."
    fi

    # Remove local Docker image
    echo "Removing local Docker image..."
    docker rmi hello-world:latest 2>/dev/null || echo "Failed to remove local
 image, it may not exist or may have already been deleted."
    if [ -n "$REPO_URI" ]; then
        docker rmi "$REPO_URI:latest" 2>/dev/null || echo "Failed to remove
 tagged image, it may not exist or may have already been deleted."
    fi

    echo "Cleanup completed."
    echo "================================================="
}

# Step 1: Create a Docker image
echo "Step 1: Creating a Docker image"

# Create Dockerfile
```

```
echo "Creating Dockerfile..."
cat > Dockerfile << 'EOF'
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

# Install dependencies
RUN yum update -y && \
 yum install -y httpd

# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache
RUN echo 'mkdir -p /var/run/httpd' >> /root/run_apache.sh && \
 echo 'mkdir -p /var/lock/httpd' >> /root/run_apache.sh && \
 echo '/usr/sbin/httpd -D FOREGROUND' >> /root/run_apache.sh && \
 chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
EOF

# Build Docker image
echo "Building Docker image..."
$TIMEOUT_CMD docker build -t hello-world . || handle_error "Failed to build
 Docker image or operation timed out after 5 minutes"

# Verify image was created
echo "Verifying Docker image..."
docker images --filter reference=hello-world || handle_error "Failed to list
 Docker images"

echo "Docker image created successfully."

# Step 2: Create an Amazon ECR repository
echo "Step 2: Creating an Amazon ECR repository"

# Get AWS account ID
echo "Getting AWS account ID..."
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
if [[ -z "$AWS_ACCOUNT_ID" || "$AWS_ACCOUNT_ID" == *"error"* ]]; then
    handle_error "Failed to get AWS account ID. Make sure your AWS credentials
 are configured correctly."
fi
```

```
echo "AWS Account ID: $AWS_ACCOUNT_ID"

# Get current region
AWS_REGION=$(aws configure get region)
if [[ -z "$AWS_REGION" ]]; then
    AWS_REGION="us-east-1"  # Default to us-east-1 if no region is configured
    echo "No AWS region configured, defaulting to $AWS_REGION"
else
    echo "Using AWS region: $AWS_REGION"
fi

# Create ECR repository
echo "Creating ECR repository..."
REPO_RESULT=$(aws ecr create-repository --repository-name hello-repository --
tags key=project,value=doc-smith key=tutorial,value=amazon-elastic-container-
registry-gs)
if [[ -z "$REPO_RESULT" || "$REPO_RESULT" == *"error"* ]]; then
    handle_error "Failed to create ECR repository"
fi

# Extract repository URI
REPO_URI="$AWS_ACCOUNT_ID.dkr.ecr.$AWS_REGION.amazonaws.com/hello-repository"
echo "Repository URI: $REPO_URI"

# Step 3: Authenticate to Amazon ECR
echo "Step 3: Authenticating to Amazon ECR"

echo "Getting ECR login password..."
aws ecr get-login-password --region "$AWS_REGION" | docker login --username
 AWS --password-stdin "$AWS_ACCOUNT_ID.dkr.ecr.$AWS_REGION.amazonaws.com" ||
 handle_error "Failed to authenticate to ECR"

echo "Successfully authenticated to ECR."

# Step 4: Push the image to Amazon ECR
echo "Step 4: Pushing the image to Amazon ECR"

# Tag the image
echo "Tagging Docker image..."
docker tag hello-world:latest "$REPO_URI:latest" || handle_error "Failed to tag
 Docker image"

# Push the image with timeout
echo "Pushing image to ECR..."
```

```
$TIMEOUT_CMD docker push "$REPO_URI:latest" || handle_error "Failed to push image
 to ECR or operation timed out after 5 minutes"

echo "Successfully pushed image to ECR."

# Step 5: Pull the image from Amazon ECR
echo "Step 5: Pulling the image from Amazon ECR"

# Remove local image to ensure we're pulling from ECR
echo "Removing local tagged image..."
docker rmi "$REPO_URI:latest" || echo "Warning: Failed to remove local tagged
 image"

# Pull the image with timeout
echo "Pulling image from ECR..."
$TIMEOUT_CMD docker pull "$REPO_URI:latest" || handle_error "Failed to pull image
 from ECR or operation timed out after 5 minutes"

echo "Successfully pulled image from ECR."

# List resources created
echo "================================================="
echo "Resources created:"
echo "- Docker image: hello-world (local)"
echo "- ECR Repository: hello-repository"
echo "- ECR Image: $REPO_URI:latest"
echo "================================================="

# Ask user if they want to clean up resources
echo ""
echo "========================================="
echo "CLEANUP CONFIRMATION"
echo "========================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    # Step 6: Delete the image from ECR
    echo "Step 6: Deleting the image from ECR"

    DELETE_IMAGE_RESULT=$(aws ecr batch-delete-image --repository-name hello-
repository --image-ids imageTag=latest)
    if [[ -z "$DELETE_IMAGE_RESULT" || "$DELETE_IMAGE_RESULT" == *"error"* ]];
 then
```

```
        echo "Warning: Failed to delete image from ECR"
    else
        echo "Successfully deleted image from ECR."
    fi

    # Step 7: Delete the ECR repository
    echo "Step 7: Deleting the ECR repository"

    DELETE_REPO_RESULT=$(aws ecr delete-repository --repository-name hello-
repository --force)
    if [[ -z "$DELETE_REPO_RESULT" || "$DELETE_REPO_RESULT" == *"error"* ]]; then
        echo "Warning: Failed to delete ECR repository"
    else
        echo "Successfully deleted ECR repository."
    fi

    # Remove local Docker images
    echo "Removing local Docker images..."
    docker rmi hello-world:latest 2>/dev/null || echo "Warning: Failed to remove
 local image"

    echo "All resources have been cleaned up."
else
    echo "Resources were not cleaned up. You can manually clean up later with:"
    echo "aws ecr batch-delete-image --repository-name hello-repository --image-
ids imageTag=latest"
    echo "aws ecr delete-repository --repository-name hello-repository --force"
    echo "docker rmi hello-world:latest"
    echo "docker rmi $REPO_URI:latest"
fi

echo "================================================="
echo "Tutorial completed!"
echo "Log file: $LOG_FILE"
echo "================================================="
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - BatchDeleteImage
  - CreateRepository
  - DeleteRepository

- GetCallerIdentity
- GetLoginPassword

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva alla messaggistica sull'Internet of Things

L'esempio di codice seguente mostra come:

- Crea risorse IoT
- Configurazione del dispositivo
- Esecuzione dell'applicazione di esempio
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS IoT Core Getting Started Script
# This script creates AWS IoT resources, configures a device, and runs a sample
 application

# Set up logging
LOG_FILE="iot-core-setup.log"
echo "Starting AWS IoT Core setup at $(date)" > $LOG_FILE

# Function to log commands and their outputs
```

```
log_cmd() {
    echo "$(date): Running command: $1" >> $LOG_FILE
    eval "$1" 2>&1 | tee -a $LOG_FILE
    return ${PIPESTATUS[0]}
}

# Function to check for errors
check_error() {
    if [ $1 -ne 0 ]; then
        echo "ERROR: Command failed with exit code $1" | tee -a $LOG_FILE
        echo "Please check the log file $LOG_FILE for details" | tee -a $LOG_FILE
        cleanup_on_error
        exit $1
    fi
}

# Function to cleanup resources on error
cleanup_on_error() {
    echo "Error encountered. Attempting to clean up resources..." | tee -a
 $LOG_FILE
    echo "Resources created:" | tee -a $LOG_FILE
    if [ ! -z "$CERTIFICATE_ARN" ]; then
        echo "Certificate ARN: $CERTIFICATE_ARN" | tee -a $LOG_FILE
        if [ ! -z "$POLICY_NAME" ]; then
            log_cmd "aws iot detach-policy --policy-name $POLICY_NAME --target
 $CERTIFICATE_ARN"
        fi
        if [ ! -z "$THING_NAME" ]; then
            log_cmd "aws iot detach-thing-principal --thing-name $THING_NAME --
principal $CERTIFICATE_ARN"
        fi
        if [ ! -z "$CERTIFICATE_ID" ]; then
            log_cmd "aws iot update-certificate --certificate-id $CERTIFICATE_ID
 --new-status INACTIVE"
            log_cmd "aws iot delete-certificate --certificate-id $CERTIFICATE_ID"
        fi
    fi
    if [ ! -z "$THING_NAME" ]; then
        echo "Thing Name: $THING_NAME" | tee -a $LOG_FILE
        log_cmd "aws iot delete-thing --thing-name $THING_NAME"
    fi
    if [ ! -z "$POLICY_NAME" ]; then
        echo "Policy Name: $POLICY_NAME" | tee -a $LOG_FILE
        log_cmd "aws iot delete-policy --policy-name $POLICY_NAME"
```

```
    fi
    if [ ! -z "$SHARED_POLICY_NAME" ]; then
        echo "Shared Policy Name: $SHARED_POLICY_NAME" | tee -a $LOG_FILE
        log_cmd "aws iot delete-policy --policy-name $SHARED_POLICY_NAME"
    fi
}

# Generate unique identifiers
RANDOM_SUFFIX=$(openssl rand -hex 4)
THING_NAME="MyIoTThing-${RANDOM_SUFFIX}"
POLICY_NAME="MyIoTPolicy-${RANDOM_SUFFIX}"
SHARED_POLICY_NAME="SharedSubPolicy-${RANDOM_SUFFIX}"
CERTS_DIR="$HOME/certs"

echo "================================================" | tee -a $LOG_FILE
echo "AWS IoT Core Getting Started" | tee -a $LOG_FILE
echo "================================================" | tee -a $LOG_FILE
echo "This script will:" | tee -a $LOG_FILE
echo "1. Create AWS IoT resources (policy, thing, certificate)" | tee -a
 $LOG_FILE
echo "2. Configure your device" | tee -a $LOG_FILE
echo "3. Set up for running the sample application" | tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "Thing Name: $THING_NAME" | tee -a $LOG_FILE
echo "Policy Name: $POLICY_NAME" | tee -a $LOG_FILE
echo "Certificates Directory: $CERTS_DIR" | tee -a $LOG_FILE
echo "================================================" | tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE

# Get AWS account ID
echo "Getting AWS account ID..." | tee -a $LOG_FILE
ACCOUNT_ID=$(log_cmd "aws sts get-caller-identity --query Account --output text")
check_error $?

# Get AWS region
echo "Getting AWS region..." | tee -a $LOG_FILE
REGION=$(log_cmd "aws configure get region")
check_error $?
if [ -z "$REGION" ]; then
    echo "AWS region not configured. Please run 'aws configure' to set your
 region." | tee -a $LOG_FILE
    exit 1
fi
```

```
echo "Using AWS Account ID: $ACCOUNT_ID and Region: $REGION" | tee -a $LOG_FILE

# Step 1: Create AWS IoT Resources
echo "" | tee -a $LOG_FILE
echo "Step 1: Creating AWS IoT Resources..." | tee -a $LOG_FILE

# Create IoT policy
echo "Creating IoT policy document..." | tee -a $LOG_FILE
cat > iot-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:$REGION:$ACCOUNT_ID:client/test-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:$REGION:$ACCOUNT_ID:topic/test/topic"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:$REGION:$ACCOUNT_ID:topicfilter/test/topic"
      ]
    }
  ]
}
EOF
```

```
echo "Creating IoT policy: $POLICY_NAME..." | tee -a $LOG_FILE
log_cmd "aws iot create-policy --policy-name $POLICY_NAME --policy-
document file://iot-policy.json --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-iot-core-gs"
check_error $?

# Create IoT thing
echo "Creating IoT thing: $THING_NAME..." | tee -a $LOG_FILE
log_cmd "aws iot create-thing --thing-name $THING_NAME"
check_error $?

THING_ARN=$(aws iot describe-thing --thing-name "$THING_NAME" --query 'thingArn'
 --output text)
aws iot tag-resource --resource-arn "$THING_ARN" --tags Key=project,Value=doc-
smith Key=tutorial,Value=aws-iot-core-gs

# Create directory for certificates
echo "Creating certificates directory..." | tee -a $LOG_FILE
log_cmd "mkdir -p $CERTS_DIR"
check_error $?

# Create keys and certificate
echo "Creating keys and certificate..." | tee -a $LOG_FILE
CERT_OUTPUT=$(log_cmd "aws iot create-keys-and-certificate --set-as-active
 --certificate-pem-outfile $CERTS_DIR/device.pem.crt --public-key-outfile
 $CERTS_DIR/public.pem.key --private-key-outfile $CERTS_DIR/private.pem.key")
check_error $?

# Extract certificate ARN and ID
CERTIFICATE_ARN=$(echo "$CERT_OUTPUT" | grep "certificateArn" | cut -d'"' -f4)
CERTIFICATE_ID=$(echo "$CERTIFICATE_ARN" | cut -d/ -f2)

if [ -z "$CERTIFICATE_ARN" ] || [ -z "$CERTIFICATE_ID" ]; then
    echo "Failed to extract certificate ARN or ID" | tee -a $LOG_FILE
    cleanup_on_error
    exit 1
fi

echo "Certificate ARN: $CERTIFICATE_ARN" | tee -a $LOG_FILE
echo "Certificate ID: $CERTIFICATE_ID" | tee -a $LOG_FILE

# Attach policy to certificate
echo "Attaching policy to certificate..." | tee -a $LOG_FILE
```

```
log_cmd "aws iot attach-policy --policy-name $POLICY_NAME --target
 $CERTIFICATE_ARN"
check_error $?

# Attach certificate to thing
echo "Attaching certificate to thing..." | tee -a $LOG_FILE
log_cmd "aws iot attach-thing-principal --thing-name $THING_NAME --principal
 $CERTIFICATE_ARN"
check_error $?

# Download Amazon Root CA certificate
echo "Downloading Amazon Root CA certificate..." | tee -a $LOG_FILE
log_cmd "curl -s -o $CERTS_DIR/Amazon-root-CA-1.pem https://www.amazontrust.com/
repository/AmazonRootCA1.pem"
check_error $?

# Step 2: Configure Your Device
echo "" | tee -a $LOG_FILE
echo "Step 2: Configuring Your Device..." | tee -a $LOG_FILE

# Check if Git is installed
echo "Checking if Git is installed..." | tee -a $LOG_FILE
if ! command -v git &> /dev/null; then
    echo "Git is not installed. Please install Git and run this script again." |
 tee -a $LOG_FILE
    cleanup_on_error
    exit 1
fi

# Check if Python is installed
echo "Checking if Python is installed..." | tee -a $LOG_FILE
if ! command -v python3 &> /dev/null; then
    echo "Python 3 is not installed. Please install Python 3 and run this script
 again." | tee -a $LOG_FILE
    cleanup_on_error
    exit 1
fi

# Install AWS IoT Device SDK for Python
echo "Installing AWS IoT Device SDK for Python..." | tee -a $LOG_FILE
log_cmd "python3 -m pip install awsiotsdk"
check_error $?

# Clone the AWS IoT Device SDK for Python repository
```

```
echo "Cloning AWS IoT Device SDK for Python repository..." | tee -a $LOG_FILE
if [ ! -d "$HOME/aws-iot-device-sdk-python-v2" ]; then
    log_cmd "cd $HOME && git clone https://github.com/aws/aws-iot-device-sdk-
python-v2.git"
    check_error $?
else
    echo "AWS IoT Device SDK for Python repository already exists." | tee -a
 $LOG_FILE
fi

# Step 3: Get AWS IoT Endpoint
echo "" | tee -a $LOG_FILE
echo "Step 3: Getting AWS IoT Endpoint..." | tee -a $LOG_FILE

IOT_ENDPOINT=$(log_cmd "aws iot describe-endpoint --endpoint-type iot:Data-ATS --
query endpointAddress --output text")
check_error $?

echo "AWS IoT Endpoint: $IOT_ENDPOINT" | tee -a $LOG_FILE

# Create a shared subscription policy (optional)
echo "" | tee -a $LOG_FILE
echo "Creating shared subscription policy (optional)..." | tee -a $LOG_FILE

cat > shared-sub-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:$REGION:$ACCOUNT_ID:client/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:Receive"
      ],
      "Resource": [
```

```
        "arn:aws:iot:$REGION:$ACCOUNT_ID:topic/test/topic"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:$REGION:$ACCOUNT_ID:topicfilter/test/topic",
        "arn:aws:iot:$REGION:$ACCOUNT_ID:topicfilter/\$share/*/test/topic"
      ]
    }
  ]
}
EOF

log_cmd "aws iot create-policy --policy-name $SHARED_POLICY_NAME --policy-
document file://shared-sub-policy.json --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-iot-core-gs"
check_error $?

log_cmd "aws iot attach-policy --policy-name $SHARED_POLICY_NAME --target
 $CERTIFICATE_ARN"
check_error $?

# Summary of created resources
echo "" | tee -a $LOG_FILE
echo "================================================" | tee -a $LOG_FILE
echo "Setup Complete! Resources Created:" | tee -a $LOG_FILE
echo "================================================" | tee -a $LOG_FILE
echo "Thing Name: $THING_NAME" | tee -a $LOG_FILE
echo "Policy Name: $POLICY_NAME" | tee -a $LOG_FILE
echo "Shared Subscription Policy Name: $SHARED_POLICY_NAME" | tee -a $LOG_FILE
echo "Certificate ID: $CERTIFICATE_ID" | tee -a $LOG_FILE
echo "Certificate ARN: $CERTIFICATE_ARN" | tee -a $LOG_FILE
echo "Certificate Files Location: $CERTS_DIR" | tee -a $LOG_FILE
echo "AWS IoT Endpoint: $IOT_ENDPOINT" | tee -a $LOG_FILE
echo "================================================" | tee -a $LOG_FILE

# Instructions for running the sample application
echo "" | tee -a $LOG_FILE
echo "To run the sample application, execute:" | tee -a $LOG_FILE
echo "cd $HOME/aws-iot-device-sdk-python-v2/samples" | tee -a $LOG_FILE
```

```
echo "python3 pubsub.py \\" | tee -a $LOG_FILE
echo "  --endpoint $IOT_ENDPOINT \\" | tee -a $LOG_FILE
echo "  --ca_file $CERTS_DIR/Amazon-root-CA-1.pem \\" | tee -a $LOG_FILE
echo "  --cert $CERTS_DIR/device.pem.crt \\" | tee -a $LOG_FILE
echo "  --key $CERTS_DIR/private.pem.key" | tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "To run the shared subscription example, execute:" | tee -a $LOG_FILE
echo "cd $HOME/aws-iot-device-sdk-python-v2/samples" | tee -a $LOG_FILE
echo "python3 mqtt5_shared_subscription.py \\" | tee -a $LOG_FILE
echo "  --endpoint $IOT_ENDPOINT \\" | tee -a $LOG_FILE
echo "  --ca_file $CERTS_DIR/Amazon-root-CA-1.pem \\" | tee -a $LOG_FILE
echo "  --cert $CERTS_DIR/device.pem.crt \\" | tee -a $LOG_FILE
echo "  --key $CERTS_DIR/private.pem.key \\" | tee -a $LOG_FILE
echo "  --group_identifier consumer" | tee -a $LOG_FILE

# Ask if user wants to clean up resources
echo "" | tee -a $LOG_FILE
echo "================================================" | tee -a $LOG_FILE
echo "CLEANUP CONFIRMATION" | tee -a $LOG_FILE
echo "================================================" | tee -a $LOG_FILE
echo "Do you want to clean up all created resources? (y/n): " | tee -a $LOG_FILE
read -r CLEANUP_CHOICE

if [[ $CLEANUP_CHOICE =~ ^[Yy]$ ]]; then
    echo "Cleaning up resources..." | tee -a $LOG_FILE

    # Detach policies from certificate
    echo "Detaching policies from certificate..." | tee -a $LOG_FILE
    log_cmd "aws iot detach-policy --policy-name $POLICY_NAME --target
 $CERTIFICATE_ARN"
    log_cmd "aws iot detach-policy --policy-name $SHARED_POLICY_NAME --target
 $CERTIFICATE_ARN"

    # Detach certificate from thing
    echo "Detaching certificate from thing..." | tee -a $LOG_FILE
    log_cmd "aws iot detach-thing-principal --thing-name $THING_NAME --principal
 $CERTIFICATE_ARN"

    # Update certificate status to INACTIVE
    echo "Setting certificate to inactive..." | tee -a $LOG_FILE
    log_cmd "aws iot update-certificate --certificate-id $CERTIFICATE_ID --new-
status INACTIVE"

    # Delete certificate
```

```
    echo "Deleting certificate..." | tee -a $LOG_FILE
    log_cmd "aws iot delete-certificate --certificate-id $CERTIFICATE_ID"

    # Delete thing
    echo "Deleting thing..." | tee -a $LOG_FILE
    log_cmd "aws iot delete-thing --thing-name $THING_NAME"

    # Delete policies
    echo "Deleting policies..." | tee -a $LOG_FILE
    log_cmd "aws iot delete-policy --policy-name $POLICY_NAME"
    log_cmd "aws iot delete-policy --policy-name $SHARED_POLICY_NAME"

    echo "Cleanup complete!" | tee -a $LOG_FILE
else
    echo "Resources were not cleaned up. You can manually clean them up later." |
 tee -a $LOG_FILE
    echo "To clean up resources, run the following commands:" | tee -a $LOG_FILE
    echo "aws iot detach-policy --policy-name $POLICY_NAME --target
 $CERTIFICATE_ARN" | tee -a $LOG_FILE
    echo "aws iot detach-policy --policy-name $SHARED_POLICY_NAME --target
 $CERTIFICATE_ARN" | tee -a $LOG_FILE
    echo "aws iot detach-thing-principal --thing-name $THING_NAME --principal
 $CERTIFICATE_ARN" | tee -a $LOG_FILE
    echo "aws iot update-certificate --certificate-id $CERTIFICATE_ID --new-
status INACTIVE" | tee -a $LOG_FILE
    echo "aws iot delete-certificate --certificate-id $CERTIFICATE_ID" | tee -a
 $LOG_FILE
    echo "aws iot delete-thing --thing-name $THING_NAME" | tee -a $LOG_FILE
    echo "aws iot delete-policy --policy-name $POLICY_NAME" | tee -a $LOG_FILE
    echo "aws iot delete-policy --policy-name $SHARED_POLICY_NAME" | tee -a
 $LOG_FILE
fi

echo "" | tee -a $LOG_FILE
echo "Script execution completed. See $LOG_FILE for details." | tee -a $LOG_FILE
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachPolicy
  - AttachThingPrincipal
  - CreateKeysAndCertificate

- CreatePolicy
- CreateThing
- DeleteCertificate
- DeletePolicy
- DeleteThing
- DescribeEndpoint
- DetachPolicy
- DetachThingPrincipal
- GetCallerIdentity
- UpdateCertificate

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva ai feature store di machine learning

L'esempio di codice seguente mostra come:

- Impostare autorizzazioni IAM
- Crea un ruolo di SageMaker esecuzione
- Creazione di gruppi di funzionalità
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon SageMaker Feature Store Tutorial Script - Version 3
# This script demonstrates how to use Amazon SageMaker Feature Store with AWS CLI

# Setup logging
LOG_FILE="sagemaker-featurestore-tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting SageMaker Feature Store tutorial script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"
echo ""

# Track created resources for cleanup
CREATED_RESOURCES=()

# Function to handle errors
handle_error() {
    echo "ERROR: $1"
    echo "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Function to check command status
check_status() {
    if echo "$1" | grep -i "error" > /dev/null; then
        handle_error "$1"
    fi
}

# Function to wait for feature group to be created
wait_for_feature_group() {
    local feature_group_name=$1
    local status="Creating"

    echo "Waiting for feature group ${feature_group_name} to be created..."

    while [ "$status" = "Creating" ]; do
        sleep 5
        status=$(aws sagemaker describe-feature-group \
            --feature-group-name "${feature_group_name}" \
            --query 'FeatureGroupStatus' \
```

```
            --output text)
        echo "Current status: ${status}"

        if [ "$status" = "Failed" ]; then
            handle_error "Feature group ${feature_group_name} creation failed"
        fi
    done

    echo "Feature group ${feature_group_name} is now ${status}"
}

# Function to clean up resources
cleanup_resources() {
    echo "Cleaning up resources..."

    # Clean up in reverse order
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        resource="${CREATED_RESOURCES[$i]}"
        resource_type=$(echo "$resource" | cut -d: -f1)
        resource_name=$(echo "$resource" | cut -d: -f2)

        echo "Deleting $resource_type: $resource_name"

        case "$resource_type" in
            "FeatureGroup")
                aws sagemaker delete-feature-group --feature-group-name
 "$resource_name"
                ;;
            "S3Bucket")
                echo "Emptying S3 bucket: $resource_name"
                aws s3 rm "s3://$resource_name" --recursive 2>/dev/null
                echo "Deleting S3 bucket: $resource_name"
                aws s3api delete-bucket --bucket "$resource_name" 2>/dev/null
                ;;
            "IAMRole")
                echo "Detaching policies from role: $resource_name"
                aws iam detach-role-policy --role-name "$resource_name" --policy-
arn "arn:aws:iam::aws:policy/AmazonSageMakerFullAccess" 2>/dev/null
                aws iam detach-role-policy --role-name "$resource_name" --policy-
arn "arn:aws:iam::aws:policy/AmazonS3FullAccess" 2>/dev/null
                echo "Deleting IAM role: $resource_name"
                aws iam delete-role --role-name "$resource_name" 2>/dev/null
                ;;
            *)
```

```
                echo "Unknown resource type: $resource_type"
                ;;
        esac
    done
}

# Function to create SageMaker execution role
create_sagemaker_role() {
    local role_name="SageMakerFeatureStoreRole-$(openssl rand -hex 4)"

    echo "Creating SageMaker execution role: $role_name" >&2

    # Create trust policy document
    local trust_policy='{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "sagemaker.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    }'

    # Create the role
    local role_result=$(aws iam create-role \
        --role-name "$role_name" \
        --assume-role-policy-document "$trust_policy" \
        --description "SageMaker execution role for Feature Store tutorial" 2>&1)

    if echo "$role_result" | grep -i "error" > /dev/null; then
        handle_error "Failed to create IAM role: $role_result"
    fi

    echo "Role created successfully" >&2
    CREATED_RESOURCES+=("IAMRole:$role_name")

    # Tag the role
    echo "Tagging IAM role..." >&2
    aws iam tag-role --role-name "$role_name" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=sagemaker-featurestore 2>&1
```

```
    # Attach necessary policies
    echo "Attaching policies to role..." >&2

    # SageMaker execution policy
    local policy1_result=$(aws iam attach-role-policy \
        --role-name "$role_name" \
        --policy-arn "arn:aws:iam::aws:policy/AmazonSageMakerFullAccess" 2>&1)

    if echo "$policy1_result" | grep -i "error" > /dev/null; then
        handle_error "Failed to attach SageMaker policy: $policy1_result"
    fi

    # S3 access policy
    local policy2_result=$(aws iam attach-role-policy \
        --role-name "$role_name" \
        --policy-arn "arn:aws:iam::aws:policy/AmazonS3FullAccess" 2>&1)

    if echo "$policy2_result" | grep -i "error" > /dev/null; then
        handle_error "Failed to attach S3 policy: $policy2_result"
    fi

    # Get account ID for role ARN
    local account_id=$(aws sts get-caller-identity --query Account --output text)
    local role_arn="arn:aws:iam::${account_id}:role/${role_name}"

    echo "Role ARN: $role_arn" >&2
    echo "Waiting 10 seconds for role to propagate..." >&2
    sleep 10

    # Return only the role ARN to stdout
    echo "$role_arn"
}

# Handle SageMaker execution role
ROLE_ARN=""

if [ -z "$1" ]; then
    echo "Creating SageMaker execution role automatically..."
    ROLE_ARN=$(create_sagemaker_role)
    if [ -z "$ROLE_ARN" ]; then
        handle_error "Failed to create SageMaker execution role"
    fi
else
    ROLE_ARN="$1"
```

```
    # Validate the role ARN
    ROLE_NAME=$(echo "$ROLE_ARN" | sed 's/.*role\///')
    ROLE_CHECK=$(aws iam get-role --role-name "$ROLE_NAME" 2>&1)
    if echo "$ROLE_CHECK" | grep -i "error" > /dev/null; then
        echo "Creating a new role automatically..."
        ROLE_ARN=$(create_sagemaker_role)
        if [ -z "$ROLE_ARN" ]; then
            handle_error "Failed to create SageMaker execution role"
        fi
    fi
fi

# Handle cleanup option
AUTO_CLEANUP=""
if [ -n "$2" ]; then
    AUTO_CLEANUP="$2"
fi

# Generate a random identifier for resource names
RANDOM_ID=$(openssl rand -hex 4)
echo "Using random identifier: $RANDOM_ID"

# Set variables
ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
check_status "$ACCOUNT_ID"
echo "Account ID: $ACCOUNT_ID"

# Get current region
REGION=$(aws configure get region)
if [ -z "$REGION" ]; then
    REGION="us-east-1"
    echo "No default region configured, using: $REGION"
else
    echo "Using region: $REGION"
fi
# Check for shared prereq bucket
PREREQ_BUCKET=$(aws cloudformation describe-stacks --stack-name tutorial-prereqs-
bucket \
    --query 'Stacks[0].Outputs[?OutputKey==`BucketName`].OutputValue' --output
 text 2>/dev/null)
if [ -n "$PREREQ_BUCKET" ] && [ "$PREREQ_BUCKET" != "None" ]; then
    S3_BUCKET_NAME="$PREREQ_BUCKET"
    BUCKET_IS_SHARED=true
```

```
    echo "Using shared bucket: $S3_BUCKET_NAME"
else
    BUCKET_IS_SHARED=false
    S3_BUCKET_NAME="sagemaker-featurestore-${RANDOM_ID}-${ACCOUNT_ID}"
fi
PREFIX="featurestore-tutorial"
CURRENT_TIME=$(date +%s)

echo "Creating S3 bucket: $S3_BUCKET_NAME"
# Create bucket in current region (skip if using shared bucket)
if [ "$BUCKET_IS_SHARED" = "false" ]; then
if [ "$REGION" = "us-east-1" ]; then
    BUCKET_RESULT=$(aws s3api create-bucket --bucket "$S3_BUCKET_NAME" \
        --region "$REGION" 2>&1)
else
    BUCKET_RESULT=$(aws s3api create-bucket --bucket "$S3_BUCKET_NAME" \
        --region "$REGION" \
        --create-bucket-configuration LocationConstraint="$REGION" 2>&1)
fi

if echo "$BUCKET_RESULT" | grep -i "error" > /dev/null; then
    echo "Failed to create S3 bucket: $BUCKET_RESULT"
    exit 1
fi

echo "$BUCKET_RESULT"
CREATED_RESOURCES+=("S3Bucket:$S3_BUCKET_NAME")

# Tag the S3 bucket
echo "Tagging S3 bucket: $S3_BUCKET_NAME"
aws s3api put-bucket-tagging --bucket "$S3_BUCKET_NAME" --tagging
 'TagSet=[{Key=project,Value=doc-smith},{Key=tutorial,Value=sagemaker-
featurestore}]' 2>&1

# Block public access to the bucket
BLOCK_RESULT=$(aws s3api put-public-access-block \
    --bucket "$S3_BUCKET_NAME" \
    --public-access-block-configuration
 "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets
 2>&1)

if echo "$BLOCK_RESULT" | grep -i "error" > /dev/null; then
    echo "Failed to block public access to S3 bucket: $BLOCK_RESULT"
    cleanup_resources
```

```
    exit 1
fi
else
    echo "Using shared bucket (skipping creation)"
fi

# Create feature groups
echo "Creating feature groups..."

# Create customers feature group
CUSTOMERS_FEATURE_GROUP_NAME="customers-feature-group-${RANDOM_ID}"
echo "Creating customers feature group: $CUSTOMERS_FEATURE_GROUP_NAME"

CUSTOMERS_RESPONSE=$(aws sagemaker create-feature-group \
    --feature-group-name "$CUSTOMERS_FEATURE_GROUP_NAME" \
    --record-identifier-feature-name "customer_id" \
    --event-time-feature-name "EventTime" \
    --feature-definitions '[
        {"FeatureName": "customer_id", "FeatureType": "Integral"},
        {"FeatureName": "name", "FeatureType": "String"},
        {"FeatureName": "age", "FeatureType": "Integral"},
        {"FeatureName": "address", "FeatureType": "String"},
        {"FeatureName": "membership_type", "FeatureType": "String"},
        {"FeatureName": "EventTime", "FeatureType": "Fractional"}
    ]' \
    --online-store-config '{"EnableOnlineStore": true}' \
    --offline-store-config '{
        "S3StorageConfig": {
            "S3Uri": "s3://'${S3_BUCKET_NAME}'/'${PREFIX}'"
        },
        "DisableGlueTableCreation": false
    }' \
    --role-arn "$ROLE_ARN" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=sagemaker-featurestore
 2>&1)

if echo "$CUSTOMERS_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to create customers feature group: $CUSTOMERS_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$CUSTOMERS_RESPONSE"
CREATED_RESOURCES+=("FeatureGroup:$CUSTOMERS_FEATURE_GROUP_NAME")
```

```
# Create orders feature group
ORDERS_FEATURE_GROUP_NAME="orders-feature-group-${RANDOM_ID}"
echo "Creating orders feature group: $ORDERS_FEATURE_GROUP_NAME"

ORDERS_RESPONSE=$(aws sagemaker create-feature-group \
    --feature-group-name "$ORDERS_FEATURE_GROUP_NAME" \
    --record-identifier-feature-name "customer_id" \
    --event-time-feature-name "EventTime" \
    --feature-definitions '[
        {"FeatureName": "customer_id", "FeatureType": "Integral"},
        {"FeatureName": "order_id", "FeatureType": "String"},
        {"FeatureName": "order_date", "FeatureType": "String"},
        {"FeatureName": "product", "FeatureType": "String"},
        {"FeatureName": "quantity", "FeatureType": "Integral"},
        {"FeatureName": "amount", "FeatureType": "Fractional"},
        {"FeatureName": "EventTime", "FeatureType": "Fractional"}
    ]' \
    --online-store-config '{"EnableOnlineStore": true}' \
    --offline-store-config '{
        "S3StorageConfig": {
            "S3Uri": "s3://'${S3_BUCKET_NAME}'/'${PREFIX}'"
        },
        "DisableGlueTableCreation": false
    }' \
    --role-arn "$ROLE_ARN" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=sagemaker-featurestore
 2>&1)

if echo "$ORDERS_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to create orders feature group: $ORDERS_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$ORDERS_RESPONSE"
CREATED_RESOURCES+=("FeatureGroup:$ORDERS_FEATURE_GROUP_NAME")

# Wait for feature groups to be created
wait_for_feature_group "$CUSTOMERS_FEATURE_GROUP_NAME"
wait_for_feature_group "$ORDERS_FEATURE_GROUP_NAME"

# Ingest data into feature groups
echo "Ingesting data into feature groups..."
```

```
# Ingest customer data
echo "Ingesting customer data..."
CUSTOMER1_RESPONSE=$(aws sagemaker-featurestore-runtime put-record \
    --feature-group-name "$CUSTOMERS_FEATURE_GROUP_NAME" \
    --record '[
        {"FeatureName": "customer_id", "ValueAsString": "573291"},
        {"FeatureName": "name", "ValueAsString": "John Doe"},
        {"FeatureName": "age", "ValueAsString": "35"},
        {"FeatureName": "address", "ValueAsString": "123 Main St"},
        {"FeatureName": "membership_type", "ValueAsString": "premium"},
        {"FeatureName": "EventTime", "ValueAsString": "'${CURRENT_TIME}'"}
    ]' 2>&1)

if echo "$CUSTOMER1_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to ingest customer 1 data: $CUSTOMER1_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$CUSTOMER1_RESPONSE"

CUSTOMER2_RESPONSE=$(aws sagemaker-featurestore-runtime put-record \
    --feature-group-name "$CUSTOMERS_FEATURE_GROUP_NAME" \
    --record '[
        {"FeatureName": "customer_id", "ValueAsString": "109382"},
        {"FeatureName": "name", "ValueAsString": "Jane Smith"},
        {"FeatureName": "age", "ValueAsString": "28"},
        {"FeatureName": "address", "ValueAsString": "456 Oak Ave"},
        {"FeatureName": "membership_type", "ValueAsString": "standard"},
        {"FeatureName": "EventTime", "ValueAsString": "'${CURRENT_TIME}'"}
    ]' 2>&1)

if echo "$CUSTOMER2_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to ingest customer 2 data: $CUSTOMER2_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$CUSTOMER2_RESPONSE"

# Ingest order data
echo "Ingesting order data..."
ORDER1_RESPONSE=$(aws sagemaker-featurestore-runtime put-record \
```

```
    --feature-group-name "$ORDERS_FEATURE_GROUP_NAME" \
    --record '[
        {"FeatureName": "customer_id", "ValueAsString": "573291"},
        {"FeatureName": "order_id", "ValueAsString": "ORD-001"},
        {"FeatureName": "order_date", "ValueAsString": "2023-01-15"},
        {"FeatureName": "product", "ValueAsString": "Laptop"},
        {"FeatureName": "quantity", "ValueAsString": "1"},
        {"FeatureName": "amount", "ValueAsString": "1299.99"},
        {"FeatureName": "EventTime", "ValueAsString": "'${CURRENT_TIME}'"}
    ]' 2>&1)

if echo "$ORDER1_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to ingest order 1 data: $ORDER1_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$ORDER1_RESPONSE"

ORDER2_RESPONSE=$(aws sagemaker-featurestore-runtime put-record \
    --feature-group-name "$ORDERS_FEATURE_GROUP_NAME" \
    --record '[
        {"FeatureName": "customer_id", "ValueAsString": "109382"},
        {"FeatureName": "order_id", "ValueAsString": "ORD-002"},
        {"FeatureName": "order_date", "ValueAsString": "2023-01-20"},
        {"FeatureName": "product", "ValueAsString": "Smartphone"},
        {"FeatureName": "quantity", "ValueAsString": "1"},
        {"FeatureName": "amount", "ValueAsString": "899.99"},
        {"FeatureName": "EventTime", "ValueAsString": "'${CURRENT_TIME}'"}
    ]' 2>&1)

if echo "$ORDER2_RESPONSE" | grep -i "error" > /dev/null; then
    echo "Failed to ingest order 2 data: $ORDER2_RESPONSE"
    cleanup_resources
    exit 1
fi

echo "$ORDER2_RESPONSE"

# Retrieve records from feature groups
echo "Retrieving records from feature groups..."

# Get a single customer record
echo "Getting customer record with ID 573291:"
```

```
CUSTOMER_RECORD=$(aws sagemaker-featurestore-runtime get-record \
    --feature-group-name "$CUSTOMERS_FEATURE_GROUP_NAME" \
    --record-identifier-value-as-string "573291" 2>&1)

if echo "$CUSTOMER_RECORD" | grep -i "error" > /dev/null; then
    echo "Failed to get customer record: $CUSTOMER_RECORD"
    cleanup_resources
    exit 1
fi

echo "$CUSTOMER_RECORD"

# Get multiple records using batch-get-record
echo "Getting multiple records using batch-get-record:"
BATCH_RECORDS=$(aws sagemaker-featurestore-runtime batch-get-record \
    --identifiers '[
        {
            "FeatureGroupName": "'${CUSTOMERS_FEATURE_GROUP_NAME}'",
            "RecordIdentifiersValueAsString": ["573291", "109382"]
        },
        {
            "FeatureGroupName": "'${ORDERS_FEATURE_GROUP_NAME}'",
            "RecordIdentifiersValueAsString": ["573291", "109382"]
        }
    ]' 2>&1)

if echo "$BATCH_RECORDS" | grep -i "error" > /dev/null && ! echo "$BATCH_RECORDS"
 | grep -i "Records" > /dev/null; then
    echo "Failed to get batch records: $BATCH_RECORDS"
    cleanup_resources
    exit 1
fi

echo "$BATCH_RECORDS"

# List feature groups
echo "Listing feature groups:"
FEATURE_GROUPS=$(aws sagemaker list-feature-groups 2>&1)

if echo "$FEATURE_GROUPS" | grep -i "error" > /dev/null; then
    echo "Failed to list feature groups: $FEATURE_GROUPS"
    cleanup_resources
    exit 1
fi
```

```
echo "$FEATURE_GROUPS"

# Display summary of created resources
echo ""
echo "============================================"
echo "TUTORIAL COMPLETED SUCCESSFULLY!"
echo "============================================"
echo "Resources created:"
echo "- S3 Bucket: $S3_BUCKET_NAME"
echo "- Customers Feature Group: $CUSTOMERS_FEATURE_GROUP_NAME"
echo "- Orders Feature Group: $ORDERS_FEATURE_GROUP_NAME"
if [[ " ${CREATED_RESOURCES[@]} " =~ " IAMRole:" ]]; then
    echo "- IAM Role: $(echo "${CREATED_RESOURCES[@]}" | grep -o 'IAMRole:
[^[:space:]]*' | cut -d: -f2)"
fi
echo ""
echo "You can now:"
echo "1. View your feature groups in the SageMaker console"
echo "2. Query the offline store using Amazon Athena"
echo "3. Use the feature groups in your ML workflows"
echo "============================================"
echo ""

# Handle cleanup
if [ "$AUTO_CLEANUP" = "y" ]; then
    echo "Auto-cleanup enabled. Starting cleanup..."
    cleanup_resources
    echo "Cleanup completed."
elif [ "$AUTO_CLEANUP" = "n" ]; then
    echo "Auto-cleanup disabled. Resources will remain in your account."
    echo "To clean up later, run this script again with cleanup option 'y'"
else
    echo "============================================"
    echo "CLEANUP CONFIRMATION"
    echo "============================================"
    echo "Do you want to clean up all created resources? (y/n): "
    read -r CLEANUP_CHOICE

    if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
        echo "Starting cleanup..."
        cleanup_resources
        echo "Cleanup completed."
    else
```

```
        echo "Skipping cleanup. Resources will remain in your account."
        echo "To clean up later, delete the following resources:"
        echo "- Feature Groups: $CUSTOMERS_FEATURE_GROUP_NAME,
 $ORDERS_FEATURE_GROUP_NAME"
        echo "- S3 Bucket: $S3_BUCKET_NAME"
        if [[ " ${CREATED_RESOURCES[@]} " =~ " IAMRole:" ]]; then
            echo "- IAM Role: $(echo "${CREATED_RESOURCES[@]}" | grep -o
 'IAMRole:[^[:space:]]*' | cut -d: -f2)"
        fi
        echo ""
        echo "Estimated ongoing cost: ~$0.01 per month for online store"
    fi
fi

echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateBucket
  - CreateFeatureGroup
  - CreateRole
  - DeleteBucket
  - DeleteFeatureGroup
  - DeleteRole
  - DescribeFeatureGroup
  - DetachRolePolicy
  - GetCallerIdentity
  - GetRole
  - ListFeatureGroups
  - PutPublicAccessBlock
  - Rm

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva ai cluster Kubernetes gestiti

L'esempio di codice seguente mostra come:

- Crea un VPC per il tuo cluster EKS
- Crea ruoli IAM per il tuo cluster EKS
- Creazione di un cluster EKS
- Configura kubectl per comunicare con il tuo cluster
- Creazione di un gruppo di nodi gestito
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon EKS Cluster Creation Script (v2)
# This script creates an Amazon EKS cluster with a managed node group using the
 AWS CLI

# Set up logging
LOG_FILE="eks-cluster-creation-v2.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon EKS cluster creation script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Error handling function
handle_error() {
    echo "ERROR: $1"
    echo "Attempting to clean up resources..."
```

```
    cleanup_resources
    exit 1
}

# Function to check command success
check_command() {
    if [ $? -ne 0 ] || echo "$1" | grep -i "error" > /dev/null; then
        handle_error "$1"
    fi
}

# Function to check if kubectl is installed
check_kubectl() {
    if ! command -v kubectl &> /dev/null; then
        echo "WARNING: kubectl is not installed or not in your PATH."
        echo ""
        echo "To install kubectl, follow these instructions based on your
 operating system:"
        echo ""
        echo "For Linux:"
        echo "  1. Download the latest release:"
        echo "     curl -LO \"https://dl.k8s.io/release/\$(curl -L -s https://
dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl\""
        echo ""
        echo "  2. Make the kubectl binary executable:"
        echo "     chmod +x ./kubectl"
        echo ""
        echo "  3. Move the binary to your PATH:"
        echo "     sudo mv ./kubectl /usr/local/bin/kubectl"
        echo ""
        echo "For macOS:"
        echo "  1. Using Homebrew:"
        echo "     brew install kubectl"
        echo "     or"
        echo "  2. Using curl:"
        echo "     curl -LO \"https://dl.k8s.io/release/\$(curl -L -s https://
dl.k8s.io/release/stable.txt)/bin/darwin/amd64/kubectl\""
        echo "     chmod +x ./kubectl"
        echo "     sudo mv ./kubectl /usr/local/bin/kubectl"
        echo ""
        echo "For Windows:"
        echo "  1. Using curl:"
        echo "     curl -LO \"https://dl.k8s.io/release/v1.28.0/bin/windows/
amd64/kubectl.exe\""
```

```
        echo "     Add the binary to your PATH"
        echo "     or"
        echo "  2. Using Chocolatey:"
        echo "     choco install kubernetes-cli"
        echo ""
        echo "After installation, verify with: kubectl version --client"
        echo ""
        return 1
    fi
    return 0
}

# Generate a random identifier for resource names
RANDOM_ID=$(LC_ALL=C tr -dc 'a-z0-9' < /dev/urandom | fold -w 6 | head -n 1)
STACK_NAME="eks-vpc-stack-${RANDOM_ID}"
CLUSTER_NAME="eks-cluster-${RANDOM_ID}"
NODEGROUP_NAME="eks-nodegroup-${RANDOM_ID}"
CLUSTER_ROLE_NAME="EKSClusterRole-${RANDOM_ID}"
NODE_ROLE_NAME="EKSNodeRole-${RANDOM_ID}"

echo "Using the following resource names:"
echo "- VPC Stack: $STACK_NAME"
echo "- EKS Cluster: $CLUSTER_NAME"
echo "- Node Group: $NODEGROUP_NAME"
echo "- Cluster IAM Role: $CLUSTER_ROLE_NAME"
echo "- Node IAM Role: $NODE_ROLE_NAME"

# Array to track created resources for cleanup
declare -a CREATED_RESOURCES

# Function to clean up resources
cleanup_resources() {
    echo "Cleaning up resources in reverse order..."

    # Check if node group exists and delete it
    if aws eks list-nodegroups --cluster-name "$CLUSTER_NAME" --query
 "nodegroups[?contains(@,'$NODEGROUP_NAME')]" --output text 2>/dev/null | grep -q
 "$NODEGROUP_NAME"; then
        echo "Deleting node group: $NODEGROUP_NAME"
        aws eks delete-nodegroup --cluster-name "$CLUSTER_NAME" --nodegroup-name
 "$NODEGROUP_NAME"
        echo "Waiting for node group deletion to complete..."
        aws eks wait nodegroup-deleted --cluster-name "$CLUSTER_NAME" --
nodegroup-name "$NODEGROUP_NAME"
```

```
        echo "Node group deleted successfully."
    fi

    # Check if cluster exists and delete it
    if aws eks describe-cluster --name "$CLUSTER_NAME" 2>/dev/null; then
        echo "Deleting cluster: $CLUSTER_NAME"
        aws eks delete-cluster --name "$CLUSTER_NAME"
        echo "Waiting for cluster deletion to complete (this may take several
 minutes)..."
        aws eks wait cluster-deleted --name "$CLUSTER_NAME"
        echo "Cluster deleted successfully."
    fi

    # Check if CloudFormation stack exists and delete it
    if aws cloudformation describe-stacks --stack-name "$STACK_NAME" 2>/dev/null;
 then
        echo "Deleting CloudFormation stack: $STACK_NAME"
        aws cloudformation delete-stack --stack-name "$STACK_NAME"
        echo "Waiting for CloudFormation stack deletion to complete..."
        aws cloudformation wait stack-delete-complete --stack-name "$STACK_NAME"
        echo "CloudFormation stack deleted successfully."
    fi

    # Clean up IAM roles
    if aws iam get-role --role-name "$NODE_ROLE_NAME" 2>/dev/null; then
        echo "Detaching policies from node role: $NODE_ROLE_NAME"
        aws iam detach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSWorkerNodePolicy --role-name "$NODE_ROLE_NAME"
        aws iam detach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEC2ContainerRegistryReadOnly --role-name "$NODE_ROLE_NAME"
        aws iam detach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKS_CNI_Policy --role-name "$NODE_ROLE_NAME"
        echo "Deleting node role: $NODE_ROLE_NAME"
        aws iam delete-role --role-name "$NODE_ROLE_NAME"
        echo "Node role deleted successfully."
    fi

    if aws iam get-role --role-name "$CLUSTER_ROLE_NAME" 2>/dev/null; then
        echo "Detaching policies from cluster role: $CLUSTER_ROLE_NAME"
        aws iam detach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSClusterPolicy --role-name "$CLUSTER_ROLE_NAME"
        echo "Deleting cluster role: $CLUSTER_ROLE_NAME"
        aws iam delete-role --role-name "$CLUSTER_ROLE_NAME"
        echo "Cluster role deleted successfully."
```

```
    fi

    echo "Cleanup complete."
}

# Trap to ensure cleanup on script exit
trap 'echo "Script interrupted. Cleaning up resources..."; cleanup_resources;
 exit 1' SIGINT SIGTERM

# Verify AWS CLI configuration
echo "Verifying AWS CLI configuration..."
AWS_ACCOUNT_INFO=$(aws sts get-caller-identity)
check_command "$AWS_ACCOUNT_INFO"
echo "AWS CLI is properly configured."

# Step 1: Create VPC using CloudFormation
echo "Step 1: Creating VPC with CloudFormation..."
echo "Creating CloudFormation stack: $STACK_NAME"

# Create the CloudFormation stack
CF_CREATE_OUTPUT=$(aws cloudformation create-stack \
  --stack-name "$STACK_NAME" \
  --template-url https://s3.us-west-2.amazonaws.com/amazon-eks/
cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=eks-gs)
check_command "$CF_CREATE_OUTPUT"
CREATED_RESOURCES+=("CloudFormation Stack: $STACK_NAME")

echo "Waiting for CloudFormation stack to complete (this may take a few
 minutes)..."
aws cloudformation wait stack-create-complete --stack-name "$STACK_NAME"
if [ $? -ne 0 ]; then
    handle_error "CloudFormation stack creation failed"
fi
echo "CloudFormation stack created successfully."

# Step 2: Create IAM roles for EKS
echo "Step 2: Creating IAM roles for EKS..."

# Create cluster role trust policy
echo "Creating cluster role trust policy..."
cat > eks-cluster-role-trust-policy.json << EOF
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create cluster role
echo "Creating cluster IAM role: $CLUSTER_ROLE_NAME"
CLUSTER_ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$CLUSTER_ROLE_NAME" \
  --assume-role-policy-document file://"eks-cluster-role-trust-policy.json")
check_command "$CLUSTER_ROLE_OUTPUT"
aws iam tag-role --role-name "$CLUSTER_ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=eks-gs
CREATED_RESOURCES+=("IAM Role: $CLUSTER_ROLE_NAME")

# Attach policy to cluster role
echo "Attaching EKS cluster policy to role..."
ATTACH_CLUSTER_POLICY_OUTPUT=$(aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy \
  --role-name "$CLUSTER_ROLE_NAME")
check_command "$ATTACH_CLUSTER_POLICY_OUTPUT"

# Create node role trust policy
echo "Creating node role trust policy..."
cat > node-role-trust-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
EOF

# Create node role
echo "Creating node IAM role: $NODE_ROLE_NAME"
NODE_ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$NODE_ROLE_NAME" \
  --assume-role-policy-document file://"node-role-trust-policy.json")
check_command "$NODE_ROLE_OUTPUT"
aws iam tag-role --role-name "$NODE_ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=eks-gs
CREATED_RESOURCES+=("IAM Role: $NODE_ROLE_NAME")

# Attach policies to node role
echo "Attaching EKS node policies to role..."
ATTACH_NODE_POLICY1_OUTPUT=$(aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name "$NODE_ROLE_NAME")
check_command "$ATTACH_NODE_POLICY1_OUTPUT"

ATTACH_NODE_POLICY2_OUTPUT=$(aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name "$NODE_ROLE_NAME")
check_command "$ATTACH_NODE_POLICY2_OUTPUT"

ATTACH_NODE_POLICY3_OUTPUT=$(aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
  --role-name "$NODE_ROLE_NAME")
check_command "$ATTACH_NODE_POLICY3_OUTPUT"

# Step 3: Get VPC and subnet information
echo "Step 3: Getting VPC and subnet information..."

VPC_ID=$(aws cloudformation describe-stacks \
  --stack-name "$STACK_NAME" \
  --query "Stacks[0].Outputs[?OutputKey=='VpcId'].OutputValue" \
  --output text)
if [ -z "$VPC_ID" ]; then
    handle_error "Failed to get VPC ID from CloudFormation stack"
fi
echo "VPC ID: $VPC_ID"

SUBNET_IDS=$(aws cloudformation describe-stacks \
  --stack-name "$STACK_NAME" \
  --query "Stacks[0].Outputs[?OutputKey=='SubnetIds'].OutputValue" \
```

```
  --output text)
if [ -z "$SUBNET_IDS" ]; then
    handle_error "Failed to get Subnet IDs from CloudFormation stack"
fi
echo "Subnet IDs: $SUBNET_IDS"

SECURITY_GROUP_ID=$(aws cloudformation describe-stacks \
  --stack-name "$STACK_NAME" \
  --query "Stacks[0].Outputs[?OutputKey=='SecurityGroups'].OutputValue" \
  --output text)
if [ -z "$SECURITY_GROUP_ID" ]; then
    handle_error "Failed to get Security Group ID from CloudFormation stack"
fi
echo "Security Group ID: $SECURITY_GROUP_ID"

# Step 4: Create EKS cluster
echo "Step 4: Creating EKS cluster: $CLUSTER_NAME"

CLUSTER_ROLE_ARN=$(aws iam get-role --role-name "$CLUSTER_ROLE_NAME" --query
 "Role.Arn" --output text)
if [ -z "$CLUSTER_ROLE_ARN" ]; then
    handle_error "Failed to get Cluster Role ARN"
fi

echo "Creating EKS cluster (this will take 10-15 minutes)..."
CREATE_CLUSTER_OUTPUT=$(aws eks create-cluster \
  --name "$CLUSTER_NAME" \
  --role-arn "$CLUSTER_ROLE_ARN" \
  --resources-vpc-config
 subnetIds="$SUBNET_IDS",securityGroupIds="$SECURITY_GROUP_ID" \
  --tags Key=project,Value=doc-smith,Key=tutorial,Value=eks-gs)
check_command "$CREATE_CLUSTER_OUTPUT"
CREATED_RESOURCES+=("EKS Cluster: $CLUSTER_NAME")

echo "Waiting for EKS cluster to become active (this may take 10-15 minutes)..."
aws eks wait cluster-active --name "$CLUSTER_NAME"
if [ $? -ne 0 ]; then
    handle_error "Cluster creation failed or timed out"
fi
echo "EKS cluster is now active."

# Step 5: Configure kubectl
echo "Step 5: Configuring kubectl to communicate with the cluster..."
```

```
# Check if kubectl is installed
if ! check_kubectl; then
    echo "Will skip kubectl configuration steps but continue with the script."
    echo "You can manually configure kubectl later with: aws eks update-
kubeconfig --name \"$CLUSTER_NAME\""
else
    UPDATE_KUBECONFIG_OUTPUT=$(aws eks update-kubeconfig --name "$CLUSTER_NAME")
    check_command "$UPDATE_KUBECONFIG_OUTPUT"
    echo "kubectl configured successfully."

    # Test kubectl configuration
    echo "Testing kubectl configuration..."
    KUBECTL_TEST_OUTPUT=$(kubectl get svc 2>&1)
    if [ $? -ne 0 ]; then
        echo "Warning: kubectl configuration test failed. This might be due to
 permissions or network issues."
        echo "Error details: $KUBECTL_TEST_OUTPUT"
        echo "Continuing with script execution..."
    else
        echo "$KUBECTL_TEST_OUTPUT"
        echo "kubectl configuration test successful."
    fi
fi

# Step 6: Create managed node group
echo "Step 6: Creating managed node group: $NODEGROUP_NAME"

NODE_ROLE_ARN=$(aws iam get-role --role-name "$NODE_ROLE_NAME" --query "Role.Arn"
 --output text)
if [ -z "$NODE_ROLE_ARN" ]; then
    handle_error "Failed to get Node Role ARN"
fi

# Convert comma-separated subnet IDs to space-separated for the create-nodegroup
 command
SUBNET_IDS_ARRAY=(${SUBNET_IDS//,/ })

echo "Creating managed node group (this will take 5-10 minutes)..."
CREATE_NODEGROUP_OUTPUT=$(aws eks create-nodegroup \
  --cluster-name "$CLUSTER_NAME" \
  --nodegroup-name "$NODEGROUP_NAME" \
  --node-role "$NODE_ROLE_ARN" \
  --subnets "${SUBNET_IDS_ARRAY[@]}" \
  --tags Key=project,Value=doc-smith,Key=tutorial,Value=eks-gs)
```

```
check_command "$CREATE_NODEGROUP_OUTPUT"
CREATED_RESOURCES+=("EKS Node Group: $NODEGROUP_NAME")

echo "Waiting for node group to become active (this may take 5-10 minutes)..."
aws eks wait nodegroup-active --cluster-name "$CLUSTER_NAME" --nodegroup-name
 "$NODEGROUP_NAME"
if [ $? -ne 0 ]; then
    handle_error "Node group creation failed or timed out"
fi
echo "Node group is now active."

# Step 7: Verify nodes
echo "Step 7: Verifying nodes..."
echo "Waiting for nodes to register with the cluster (this may take a few
 minutes)..."
sleep 60  # Give nodes more time to register

# Check if kubectl is installed before attempting to use it
if ! check_kubectl; then
    echo "Cannot verify nodes without kubectl. Skipping this step."
    echo "You can manually verify nodes after installing kubectl with: kubectl
 get nodes"
else
    NODES_OUTPUT=$(kubectl get nodes 2>&1)
    if [ $? -ne 0 ]; then
        echo "Warning: Unable to get nodes. This might be due to permissions or
 the nodes are still registering."
        echo "Error details: $NODES_OUTPUT"
        echo "Continuing with script execution..."
    else
        echo "$NODES_OUTPUT"
        echo "Nodes verified successfully."
    fi
fi

# Step 8: View resources
echo "Step 8: Viewing cluster resources..."

echo "Cluster information:"
CLUSTER_INFO=$(aws eks describe-cluster --name "$CLUSTER_NAME")
echo "$CLUSTER_INFO"

echo "Node group information:"
```

```
NODEGROUP_INFO=$(aws eks describe-nodegroup --cluster-name "$CLUSTER_NAME" --
nodegroup-name "$NODEGROUP_NAME")
echo "$NODEGROUP_INFO"

echo "Kubernetes resources:"
if ! check_kubectl; then
    echo "Cannot list Kubernetes resources without kubectl. Skipping this step."
    echo "You can manually list resources after installing kubectl with: kubectl
 get all --all-namespaces"
else
    KUBE_RESOURCES=$(kubectl get all --all-namespaces 2>&1)
    if [ $? -ne 0 ]; then
        echo "Warning: Unable to get Kubernetes resources. This might be due to
 permissions."
        echo "Error details: $KUBE_RESOURCES"
        echo "Continuing with script execution..."
    else
        echo "$KUBE_RESOURCES"
    fi
fi

# Display summary of created resources
echo ""
echo "==========================================="
echo "RESOURCES CREATED"
echo "==========================================="
for resource in "${CREATED_RESOURCES[@]}"; do
    echo "- $resource"
done
echo "==========================================="

# Prompt for cleanup
echo ""
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "${CLEANUP_CHOICE,,}" == "y" ]]; then
    cleanup_resources
else
    echo "Resources will not be cleaned up. You can manually clean them up
 later."
```

```
    echo "To clean up resources, run the following commands:"
    echo "1. Delete node group: aws eks delete-nodegroup --cluster-name
 $CLUSTER_NAME --nodegroup-name $NODEGROUP_NAME"
    echo "2. Wait for node group deletion: aws eks wait nodegroup-deleted --
cluster-name $CLUSTER_NAME --nodegroup-name $NODEGROUP_NAME"
    echo "3. Delete cluster: aws eks delete-cluster --name $CLUSTER_NAME"
    echo "4. Wait for cluster deletion: aws eks wait cluster-deleted --name
 $CLUSTER_NAME"
    echo "5. Delete CloudFormation stack: aws cloudformation delete-stack --
stack-name $STACK_NAME"
    echo "6. Detach and delete IAM roles for the node group and cluster"
fi

echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateCluster
  - CreateNodegroup
  - CreateRole
  - CreateStack
  - DeleteCluster
  - DeleteNodegroup
  - DeleteRole
  - DeleteStack
  - DescribeCluster
  - DescribeNodegroup
  - DescribeStacks
  - DetachRolePolicy
  - GetCallerIdentity
  - GetRole
  - ListNodegroups
  - UpdateKubeconfig

- Aspetta (CloudFormation)
- Attendi (Amazon EKS)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva allo streaming gestito

L'esempio di codice seguente mostra come:

- Crea un cluster MSK
- Crea autorizzazioni IAM per l'accesso MSK
- Crea una macchina client
- Ottieni broker bootstrap
- Configura la macchina client
- Crea un argomento e produce/consume dei dati
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon MSK Getting Started Tutorial Script - Version 8
# This script automates the steps in the Amazon MSK Getting Started tutorial
# It creates an MSK cluster, sets up IAM permissions, creates a client machine,
# and configures the client to interact with the cluster
```

```
# Set up logging
LOG_FILE="msk_tutorial_$(date +%Y%m%d_%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon MSK Getting Started Tutorial Script - Version 8"
echo "Logging to $LOG_FILE"
echo "============================================="

# Function to handle errors
handle_error() {
    echo "ERROR: $1"
    echo "Resources created so far:"
    if [ -n "$CLUSTER_ARN" ]; then echo "- MSK Cluster: $CLUSTER_ARN"; fi
    if [ -n "$POLICY_ARN" ]; then echo "- IAM Policy: $POLICY_ARN"; fi
    if [ -n "$ROLE_NAME" ]; then echo "- IAM Role: $ROLE_NAME"; fi
    if [ -n "$INSTANCE_PROFILE_NAME" ]; then echo "- IAM Instance Profile:
 $INSTANCE_PROFILE_NAME"; fi
    if [ -n "$CLIENT_SG_ID" ]; then echo "- Client Security Group:
 $CLIENT_SG_ID"; fi
    if [ -n "$INSTANCE_ID" ]; then echo "- EC2 Instance: $INSTANCE_ID"; fi
    if [ -n "$KEY_NAME" ]; then echo "- Key Pair: $KEY_NAME"; fi

    echo "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Function to check if a resource exists
resource_exists() {
    local resource_type="$1"
    local resource_id="$2"

    case "$resource_type" in
        "cluster")
            aws kafka describe-cluster --cluster-arn "$resource_id" &>/dev/null
            ;;
        "policy")
            aws iam get-policy --policy-arn "$resource_id" &>/dev/null
            ;;
        "role")
            aws iam get-role --role-name "$resource_id" &>/dev/null
            ;;
        "instance-profile")
```

```
            aws iam get-instance-profile --instance-profile-name "$resource_id"
 &>/dev/null
            ;;
        "security-group")
            aws ec2 describe-security-groups --group-ids "$resource_id" &>/dev/
null
            ;;
        "instance")
            aws ec2 describe-instances --instance-ids "$resource_id" --query
 'Reservations[0].Instances[0].State.Name' --output text | grep -v "terminated"
 &>/dev/null
            ;;
        "key-pair")
            aws ec2 describe-key-pairs --key-names "$resource_id" &>/dev/null
            ;;
    esac
}

# Function to remove security group references
remove_security_group_references() {
    local sg_id="$1"

    if [ -z "$sg_id" ]; then
        echo "No security group ID provided for reference removal"
        return
    fi

    echo "Removing security group references for $sg_id"

    # Get all security groups in the VPC that might reference our client security
 group
    local vpc_security_groups=$(aws ec2 describe-security-groups \
        --filters "Name=vpc-id,Values=$DEFAULT_VPC_ID" \
        --query 'SecurityGroups[].GroupId' \
        --output text 2>/dev/null)

    if [ -n "$vpc_security_groups" ]; then
        for other_sg in $vpc_security_groups; do
            if [ "$other_sg" != "$sg_id" ]; then
                echo "Checking security group $other_sg for references to $sg_id"

                # Get the security group details in JSON format
                local sg_details=$(aws ec2 describe-security-groups \
                    --group-ids "$other_sg" \
```

```
                    --output json 2>/dev/null)

                if [ -n "$sg_details" ]; then
                    # Check if our security group is referenced in inbound rules
                    local has_inbound_ref=$(echo "$sg_details" | grep -o
 "\"GroupId\": \"$sg_id\"" | head -1)

                    if [ -n "$has_inbound_ref" ]; then
                        echo "Found inbound rules in $other_sg referencing
 $sg_id, removing them..."

                        # Try to remove common rule types
                        echo "Attempting to remove all-traffic rule"
                        aws ec2 revoke-security-group-ingress \
                            --group-id "$other_sg" \
                            --protocol all \
                            --source-group "$sg_id" 2>/dev/null || echo "No all-
traffic rule to remove"

                        # Try to remove TCP rules on common ports
                        for port in 22 80 443 9092 9094 9096; do
                            aws ec2 revoke-security-group-ingress \
                                --group-id "$other_sg" \
                                --protocol tcp \
                                --port "$port" \
                                --source-group "$sg_id" 2>/dev/null || true
                        done

                        # Try to remove UDP rules
                        aws ec2 revoke-security-group-ingress \
                            --group-id "$other_sg" \
                            --protocol udp \
                            --source-group "$sg_id" 2>/dev/null || true
                    fi

                    # Check for outbound rules (less common but possible)
                    local has_outbound_ref=$(echo "$sg_details" | grep -A 20
 "IpPermissionsEgress" | grep -o "\"GroupId\": \"$sg_id\"" | head -1)

                    if [ -n "$has_outbound_ref" ]; then
                        echo "Found outbound rules in $other_sg referencing
 $sg_id, removing them..."

                        aws ec2 revoke-security-group-egress \
```

```
                            --group-id "$other_sg" \
                            --protocol all \
                            --source-group "$sg_id" 2>/dev/null || echo "No
 outbound all-traffic rule to remove"
                    fi
                fi
            fi
        done
    fi

    echo "Completed security group reference removal for $sg_id"
}

# Function to clean up resources
cleanup_resources() {
    echo "Cleaning up resources..."

    # Delete EC2 instance if it exists
    if [ -n "$INSTANCE_ID" ] && resource_exists "instance" "$INSTANCE_ID"; then
        echo "Terminating EC2 instance: $INSTANCE_ID"
        aws ec2 terminate-instances --instance-ids "$INSTANCE_ID" || echo "Failed
 to terminate instance"
        echo "Waiting for instance to terminate..."
        aws ec2 wait instance-terminated --instance-ids "$INSTANCE_ID" || echo
 "Failed to wait for instance termination"
    fi

    # Delete MSK cluster first (to remove dependencies on security group)
    if [ -n "$CLUSTER_ARN" ] && resource_exists "cluster" "$CLUSTER_ARN"; then
        echo "Deleting MSK cluster: $CLUSTER_ARN"
        aws kafka delete-cluster --cluster-arn "$CLUSTER_ARN" || echo "Failed to
 delete cluster"

        # Wait a bit for the cluster deletion to start
        echo "Waiting 30 seconds for cluster deletion to begin..."
        sleep 30
    fi

    # Remove security group references before attempting deletion
    if [ -n "$CLIENT_SG_ID" ] && resource_exists "security-group"
 "$CLIENT_SG_ID"; then
        remove_security_group_references "$CLIENT_SG_ID"

        echo "Deleting security group: $CLIENT_SG_ID"
```

```
        # Try multiple times with longer delays to ensure dependencies are
removed
        for i in {1..10}; do
            if aws ec2 delete-security-group --group-id "$CLIENT_SG_ID"; then
                echo "Security group deleted successfully"
                break
            fi
            echo "Failed to delete security group (attempt $i/10), retrying in 30
seconds..."
            sleep 30
        done
    fi

    # Delete key pair if it exists
    if [ -n "$KEY_NAME" ] && resource_exists "key-pair" "$KEY_NAME"; then
        echo "Deleting key pair: $KEY_NAME"
        aws ec2 delete-key-pair --key-name "$KEY_NAME" || echo "Failed to delete
key pair"
    fi

    # Remove role from instance profile
    if [ -n "$ROLE_NAME" ] && [ -n "$INSTANCE_PROFILE_NAME" ] && resource_exists
"instance-profile" "$INSTANCE_PROFILE_NAME"; then
        echo "Removing role from instance profile"
        aws iam remove-role-from-instance-profile \
            --instance-profile-name "$INSTANCE_PROFILE_NAME" \
            --role-name "$ROLE_NAME" || echo "Failed to remove role from instance
profile"
    fi

    # Delete instance profile
    if [ -n "$INSTANCE_PROFILE_NAME" ] && resource_exists "instance-profile"
"$INSTANCE_PROFILE_NAME"; then
        echo "Deleting instance profile: $INSTANCE_PROFILE_NAME"
        aws iam delete-instance-profile \
            --instance-profile-name "$INSTANCE_PROFILE_NAME" || echo "Failed to
delete instance profile"
    fi

    # Detach policy from role
    if [ -n "$ROLE_NAME" ] && [ -n "$POLICY_ARN" ] && resource_exists "role"
"$ROLE_NAME"; then
        echo "Detaching policy from role"
        aws iam detach-role-policy \
```

```
            --role-name "$ROLE_NAME" \
            --policy-arn "$POLICY_ARN" || echo "Failed to detach policy"
    fi

    # Delete role
    if [ -n "$ROLE_NAME" ] && resource_exists "role" "$ROLE_NAME"; then
        echo "Deleting role: $ROLE_NAME"
        aws iam delete-role --role-name "$ROLE_NAME" || echo "Failed to delete
 role"
    fi

    # Delete policy
    if [ -n "$POLICY_ARN" ] && resource_exists "policy" "$POLICY_ARN"; then
        echo "Deleting policy: $POLICY_ARN"
        aws iam delete-policy --policy-arn "$POLICY_ARN" || echo "Failed to
 delete policy"
    fi

    echo "Cleanup completed"
}

# Function to find a suitable subnet and instance type combination
find_suitable_subnet_and_instance_type() {
    local vpc_id="$1"
    local -a subnet_array=("${!2}")

    # List of instance types to try, in order of preference
    local instance_types=("t3.micro" "t2.micro" "t3.small" "t2.small")

    echo "Finding suitable subnet and instance type combination..."

    for instance_type in "${instance_types[@]}"; do
        echo "Trying instance type: $instance_type"

        for subnet_id in "${subnet_array[@]}"; do
            # Get the availability zone for this subnet
            local az=$(aws ec2 describe-subnets \
                --subnet-ids "$subnet_id" \
                --query 'Subnets[0].AvailabilityZone' \
                --output text)

            echo "  Checking subnet $subnet_id in AZ $az"

            # Check if this instance type is available in this AZ
```

```
            local available=$(aws ec2 describe-instance-type-offerings \
                --location-type availability-zone \
                --filters "Name=location,Values=$az" "Name=instance-type,Values=
$instance_type" \
                --query 'InstanceTypeOfferings[0].InstanceType' \
                --output text 2>/dev/null)

            if [ "$available" = "$instance_type" ]; then
                echo "  # Found suitable combination: $instance_type in $az
 (subnet: $subnet_id)"
                SELECTED_SUBNET_ID="$subnet_id"
                SELECTED_INSTANCE_TYPE="$instance_type"
                return 0
            else
                echo "  # $instance_type not available in $az"
            fi
        done
    done

    echo "ERROR: Could not find any suitable subnet and instance type
 combination"
    return 1
}

# Generate unique identifiers
RANDOM_SUFFIX=$(LC_ALL=C tr -dc 'a-z0-9' < /dev/urandom | fold -w 8 | head -n 1)
CLUSTER_NAME="MSKTutorialCluster-${RANDOM_SUFFIX}"
POLICY_NAME="msk-tutorial-policy-${RANDOM_SUFFIX}"
ROLE_NAME="msk-tutorial-role-${RANDOM_SUFFIX}"
INSTANCE_PROFILE_NAME="msk-tutorial-profile-${RANDOM_SUFFIX}"
SG_NAME="MSKClientSecurityGroup-${RANDOM_SUFFIX}"

echo "Using the following resource names:"
echo "- Cluster Name: $CLUSTER_NAME"
echo "- Policy Name: $POLICY_NAME"
echo "- Role Name: $ROLE_NAME"
echo "- Instance Profile Name: $INSTANCE_PROFILE_NAME"
echo "- Security Group Name: $SG_NAME"
echo "================================================"

# Step 1: Create an MSK Provisioned cluster
echo "Step 1: Creating MSK Provisioned cluster"

# Get the default VPC ID first
```

```
echo "Getting default VPC..."
DEFAULT_VPC_ID=$(aws ec2 describe-vpcs \
    --filters "Name=is-default,Values=true" \
    --query "Vpcs[0].VpcId" \
    --output text)

if [ -z "$DEFAULT_VPC_ID" ] || [ "$DEFAULT_VPC_ID" = "None" ]; then
    handle_error "Could not find default VPC. Please ensure you have a default
 VPC in your region."
fi

echo "Default VPC ID: $DEFAULT_VPC_ID"

# Get available subnets in the default VPC
echo "Getting available subnets in the default VPC..."
SUBNETS=$(aws ec2 describe-subnets \
    --filters "Name=vpc-id,Values=$DEFAULT_VPC_ID" "Name=default-for-
az,Values=true" \
    --query "Subnets[0:3].SubnetId" \
    --output text)

# Convert space-separated subnet IDs to an array
read -r -a SUBNET_ARRAY <<< "$SUBNETS"

if [ ${#SUBNET_ARRAY[@]} -lt 3 ]; then
    handle_error "Not enough subnets available in the default VPC. Need at least
 3 subnets, found ${#SUBNET_ARRAY[@]}."
fi

# Get default security group for the default VPC
echo "Getting default security group for the default VPC..."
DEFAULT_SG=$(aws ec2 describe-security-groups \
    --filters "Name=group-name,Values=default" "Name=vpc-id,Values=
$DEFAULT_VPC_ID" \
    --query "SecurityGroups[0].GroupId" \
    --output text)

if [ -z "$DEFAULT_SG" ] || [ "$DEFAULT_SG" = "None" ]; then
    handle_error "Could not find default security group for VPC $DEFAULT_VPC_ID"
fi

echo "Creating MSK cluster: $CLUSTER_NAME"
echo "Using VPC: $DEFAULT_VPC_ID"
echo "Using subnets: ${SUBNET_ARRAY[0]} ${SUBNET_ARRAY[1]} ${SUBNET_ARRAY[2]}"
```

```
echo "Using security group: $DEFAULT_SG"

# Create the MSK cluster with proper error handling
CLUSTER_RESPONSE=$(aws kafka create-cluster \
    --cluster-name "$CLUSTER_NAME" \
    --broker-node-group-info "{\"InstanceType\": \"kafka.t3.small\",
 \"ClientSubnets\": [\"${SUBNET_ARRAY[0]}\", \"${SUBNET_ARRAY[1]}\",
 \"${SUBNET_ARRAY[2]}\"], \"SecurityGroups\": [\"$DEFAULT_SG\"]}" \
    --kafka-version "3.6.0" \
    --number-of-broker-nodes 3 \
    --encryption-info "{\"EncryptionInTransit\": {\"InCluster\": true,
 \"ClientBroker\": \"TLS\"}}" \
    --tags project=doc-smith,tutorial=amazon-managed-streaming-for-apache-kafka-
gs 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create MSK cluster: $CLUSTER_RESPONSE"
fi

# Extract the cluster ARN using grep
CLUSTER_ARN=$(echo "$CLUSTER_RESPONSE" | grep -o '"ClusterArn": "[^"]*' | cut -
d'"' -f4)

if [ -z "$CLUSTER_ARN" ]; then
    handle_error "Failed to extract cluster ARN from response: $CLUSTER_RESPONSE"
fi

echo "MSK cluster creation initiated. ARN: $CLUSTER_ARN"
echo "Waiting for cluster to become active (this may take 15-20 minutes)..."

# Wait for the cluster to become active
while true; do
    CLUSTER_STATUS=$(aws kafka describe-cluster --cluster-arn "$CLUSTER_ARN" --
query "ClusterInfo.State" --output text 2>/dev/null)

    if [ $? -ne 0 ]; then
        echo "Failed to get cluster status. Retrying in 30 seconds..."
        sleep 30
        continue
    fi

    echo "Current cluster status: $CLUSTER_STATUS"
```

```
    if [ "$CLUSTER_STATUS" = "ACTIVE" ]; then
        echo "Cluster is now active!"
        break
    elif [ "$CLUSTER_STATUS" = "FAILED" ]; then
        handle_error "Cluster creation failed"
    fi

    echo "Still waiting for cluster to become active... (checking again in 60
 seconds)"
    sleep 60
done

echo "============================================="
# Step 2: Create an IAM role granting access to create topics on the Amazon MSK
 cluster
echo "Step 2: Creating IAM policy and role"

# Get account ID and region
ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
REGION=$(aws configure get region)
if [ -z "$REGION" ]; then
    REGION=$(aws ec2 describe-availability-zones --query
 'AvailabilityZones[0].RegionName' --output text)
fi

if [ -z "$ACCOUNT_ID" ] || [ -z "$REGION" ]; then
    handle_error "Could not determine AWS account ID or region"
fi

echo "Account ID: $ACCOUNT_ID"
echo "Region: $REGION"

# Create IAM policy
echo "Creating IAM policy: $POLICY_NAME"
POLICY_DOCUMENT="{
    \"Version\": \"2012-10-17\",
    \"Statement\": [
        {
            \"Effect\": \"Allow\",
            \"Action\": [
                \"kafka-cluster:Connect\",
                \"kafka-cluster:AlterCluster\",
                \"kafka-cluster:DescribeCluster\"
            ],
```

```
            \"Resource\": [
                \"$CLUSTER_ARN\"
            ]
        },
        {
            \"Effect\": \"Allow\",
            \"Action\": [
                \"kafka-cluster:*Topic*\",
                \"kafka-cluster:WriteData\",
                \"kafka-cluster:ReadData\"
            ],
            \"Resource\": [
                \"arn:aws:kafka:$REGION:$ACCOUNT_ID:topic/$CLUSTER_NAME/*\"
            ]
        },
        {
            \"Effect\": \"Allow\",
            \"Action\": [
                \"kafka-cluster:AlterGroup\",
                \"kafka-cluster:DescribeGroup\"
            ],
            \"Resource\": [
                \"arn:aws:kafka:$REGION:$ACCOUNT_ID:group/$CLUSTER_NAME/*\"
            ]
        }
    ]
}"

POLICY_RESPONSE=$(aws iam create-policy \
    --policy-name "$POLICY_NAME" \
    --policy-document "$POLICY_DOCUMENT" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create IAM policy: $POLICY_RESPONSE"
fi

# Extract the policy ARN using grep
POLICY_ARN=$(echo "$POLICY_RESPONSE" | grep -o '"Arn": "[^"]*' | cut -d'"' -f4)

if [ -z "$POLICY_ARN" ]; then
    handle_error "Failed to extract policy ARN from response: $POLICY_RESPONSE"
fi
```

```
aws iam tag-policy --policy-arn "$POLICY_ARN" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=amazon-managed-
streaming-for-apache-kafka-gs

echo "IAM policy created. ARN: $POLICY_ARN"

# Create IAM role for EC2
echo "Creating IAM role: $ROLE_NAME"
TRUST_POLICY="{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"ec2.amazonaws.com\"},\"Action\":\"sts:AssumeRole
\"}]}"

ROLE_RESPONSE=$(aws iam create-role \
    --role-name "$ROLE_NAME" \
    --assume-role-policy-document "$TRUST_POLICY" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create IAM role: $ROLE_RESPONSE"
fi

echo "IAM role created: $ROLE_NAME"
aws iam tag-role --role-name "$ROLE_NAME" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=amazon-managed-
streaming-for-apache-kafka-gs

# Attach policy to role
echo "Attaching policy to role"
ATTACH_RESPONSE=$(aws iam attach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "$POLICY_ARN" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to attach policy to role: $ATTACH_RESPONSE"
fi

echo "Policy attached to role"

# Create instance profile and add role to it
echo "Creating instance profile: $INSTANCE_PROFILE_NAME"
PROFILE_RESPONSE=$(aws iam create-instance-profile \
    --instance-profile-name "$INSTANCE_PROFILE_NAME" 2>&1)
```

```
# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create instance profile: $PROFILE_RESPONSE"
fi

echo "Instance profile created"

echo "Adding role to instance profile"
ADD_ROLE_RESPONSE=$(aws iam add-role-to-instance-profile \
    --instance-profile-name "$INSTANCE_PROFILE_NAME" \
    --role-name "$ROLE_NAME" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to add role to instance profile: $ADD_ROLE_RESPONSE"
fi

echo "Role added to instance profile"

# Wait a moment for IAM propagation
echo "Waiting 10 seconds for IAM propagation..."
sleep 10

echo "============================================="

# Step 3: Create a client machine
echo "Step 3: Creating client machine"

# Find a suitable subnet and instance type combination
if ! find_suitable_subnet_and_instance_type "$DEFAULT_VPC_ID" SUBNET_ARRAY[@];
 then
    handle_error "Could not find a suitable subnet and instance type combination"
fi

echo "Selected subnet: $SELECTED_SUBNET_ID"
echo "Selected instance type: $SELECTED_INSTANCE_TYPE"

# Verify the subnet is in the same VPC we're using
SUBNET_VPC_ID=$(aws ec2 describe-subnets \
    --subnet-ids "$SELECTED_SUBNET_ID" \
    --query 'Subnets[0].VpcId' \
    --output text)

if [ "$SUBNET_VPC_ID" != "$DEFAULT_VPC_ID" ]; then
```

```
    handle_error "Subnet VPC ($SUBNET_VPC_ID) does not match default VPC
 ($DEFAULT_VPC_ID)"
fi

echo "VPC ID: $SUBNET_VPC_ID"

# Get security group ID from the MSK cluster
echo "Getting security group ID from the MSK cluster"
MSK_SG_ID=$(aws kafka describe-cluster \
    --cluster-arn "$CLUSTER_ARN" \
    --query 'ClusterInfo.BrokerNodeGroupInfo.SecurityGroups[0]' \
    --output text)

if [ -z "$MSK_SG_ID" ] || [ "$MSK_SG_ID" = "None" ]; then
    handle_error "Failed to get security group ID from cluster"
fi

echo "MSK security group ID: $MSK_SG_ID"

# Create security group for client
echo "Creating security group for client: $SG_NAME"
CLIENT_SG_RESPONSE=$(aws ec2 create-security-group \
    --group-name "$SG_NAME" \
    --description "Security group for MSK client" \
    --vpc-id "$DEFAULT_VPC_ID" \
    --tag-specifications 'ResourceType=security-
group,Tags=[{Key=project,Value=doc-smith},{Key=tutorial,Value=amazon-managed-
streaming-for-apache-kafka-gs}]' 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create security group: $CLIENT_SG_RESPONSE"
fi

# Extract the security group ID using grep
CLIENT_SG_ID=$(echo "$CLIENT_SG_RESPONSE" | grep -o '"GroupId": "[^"]*' | cut -
d'"' -f4)

if [ -z "$CLIENT_SG_ID" ]; then
    handle_error "Failed to extract security group ID from response:
 $CLIENT_SG_RESPONSE"
fi

echo "Client security group created. ID: $CLIENT_SG_ID"
```

```
# Allow SSH access to client from your IP only
echo "Getting your public IP address"
MY_IP=$(curl -s https://checkip.amazonaws.com 2>/dev/null)

if [ -z "$MY_IP" ]; then
    echo "Warning: Could not determine your IP address. Using 0.0.0.0/0 (not
 recommended for production)"
    MY_IP="0.0.0.0/0"
else
    MY_IP="$MY_IP/32"
    echo "Your IP address: $MY_IP"
fi

echo "Adding SSH ingress rule to client security group"
SSH_RULE_RESPONSE=$(aws ec2 authorize-security-group-ingress \
    --group-id "$CLIENT_SG_ID" \
    --protocol tcp \
    --port 22 \
    --cidr "$MY_IP" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    echo "Warning: Failed to add SSH ingress rule: $SSH_RULE_RESPONSE"
    echo "You may need to manually add SSH access to security group
 $CLIENT_SG_ID"
fi

echo "SSH ingress rule added"

# Update MSK security group to allow traffic from client security group
echo "Adding ingress rule to MSK security group to allow traffic from client"
MSK_RULE_RESPONSE=$(aws ec2 authorize-security-group-ingress \
    --group-id "$MSK_SG_ID" \
    --protocol all \
    --source-group "$CLIENT_SG_ID" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    echo "Warning: Failed to add ingress rule to MSK security group:
 $MSK_RULE_RESPONSE"
    echo "You may need to manually add ingress rule to security group $MSK_SG_ID"
fi
```

```
echo "Ingress rule added to MSK security group"

# Create key pair
KEY_NAME="MSKKeyPair-${RANDOM_SUFFIX}"
echo "Creating key pair: $KEY_NAME"
KEY_RESPONSE=$(aws ec2 create-key-pair --key-name "$KEY_NAME" --tag-
specifications 'ResourceType=key-pair,Tags=[{Key=project,Value=doc-smith},
{Key=tutorial,Value=amazon-managed-streaming-for-apache-kafka-gs}]' --query
 'KeyMaterial' --output text 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to create key pair: $KEY_RESPONSE"
fi

# Save the private key to a file
KEY_FILE="${KEY_NAME}.pem"
echo "$KEY_RESPONSE" > "$KEY_FILE"
chmod 400 "$KEY_FILE"

echo "Key pair created and saved to $KEY_FILE"

# Get the latest Amazon Linux 2 AMI
echo "Getting latest Amazon Linux 2 AMI ID"
AMI_ID=$(aws ec2 describe-images \
    --owners amazon \
    --filters "Name=name,Values=amzn2-ami-hvm-*-x86_64-gp2"
 "Name=state,Values=available" \
    --query "sort_by(Images, &CreationDate)[-1].ImageId" \
    --output text 2>/dev/null)

if [ -z "$AMI_ID" ] || [ "$AMI_ID" = "None" ]; then
    handle_error "Failed to get Amazon Linux 2 AMI ID"
fi

echo "Using AMI ID: $AMI_ID"

# Launch EC2 instance with the selected subnet and instance type
echo "Launching EC2 instance"
echo "Instance type: $SELECTED_INSTANCE_TYPE"
echo "Subnet: $SELECTED_SUBNET_ID"

INSTANCE_RESPONSE=$(aws ec2 run-instances \
    --image-id "$AMI_ID" \
```

```
    --instance-type "$SELECTED_INSTANCE_TYPE" \
    --key-name "$KEY_NAME" \
    --security-group-ids "$CLIENT_SG_ID" \
    --subnet-id "$SELECTED_SUBNET_ID" \
    --iam-instance-profile "Name=$INSTANCE_PROFILE_NAME" \
    --tag-specifications
 "ResourceType=instance,Tags=[{Key=Name,Value=MSKTutorialClient-
${RANDOM_SUFFIX}},{Key=project,Value=doc-smith},{Key=tutorial,Value=amazon-
managed-streaming-for-apache-kafka-gs}]" 2>&1)

# Check if the command was successful
if [ $? -ne 0 ]; then
    handle_error "Failed to launch EC2 instance: $INSTANCE_RESPONSE"
fi

# Extract the instance ID using grep
INSTANCE_ID=$(echo "$INSTANCE_RESPONSE" | grep -o '"InstanceId": "[^"]*' | head
 -1 | cut -d'"' -f4)

if [ -z "$INSTANCE_ID" ]; then
    handle_error "Failed to extract instance ID from response:
 $INSTANCE_RESPONSE"
fi

echo "EC2 instance launched successfully. ID: $INSTANCE_ID"
echo "Waiting for instance to be running..."

# Wait for the instance to be running
aws ec2 wait instance-running --instance-ids "$INSTANCE_ID"

if [ $? -ne 0 ]; then
    handle_error "Instance failed to reach running state"
fi

# Wait a bit more for the instance to initialize
echo "Instance is running. Waiting 30 seconds for initialization..."
sleep 30

# Get public DNS name of instance
CLIENT_DNS=$(aws ec2 describe-instances \
    --instance-ids "$INSTANCE_ID" \
    --query 'Reservations[0].Instances[0].PublicDnsName' \
    --output text)
```

```
if [ -z "$CLIENT_DNS" ] || [ "$CLIENT_DNS" = "None" ]; then
    echo "Warning: Could not get public DNS name for instance. Trying public
 IP..."
    CLIENT_DNS=$(aws ec2 describe-instances \
        --instance-ids "$INSTANCE_ID" \
        --query 'Reservations[0].Instances[0].PublicIpAddress' \
        --output text)

    if [ -z "$CLIENT_DNS" ] || [ "$CLIENT_DNS" = "None" ]; then
        handle_error "Failed to get public DNS name or IP address for instance"
    fi
fi

echo "Client instance DNS/IP: $CLIENT_DNS"
echo "============================================="
# Get bootstrap brokers with improved logic
echo "Getting bootstrap brokers"
MAX_RETRIES=10
RETRY_COUNT=0
BOOTSTRAP_BROKERS=""
AUTH_METHOD=""

while [ -z "$BOOTSTRAP_BROKERS" ] || [ "$BOOTSTRAP_BROKERS" = "None" ]; do
    # Get the full bootstrap brokers response
    BOOTSTRAP_RESPONSE=$(aws kafka get-bootstrap-brokers \
        --cluster-arn "$CLUSTER_ARN" 2>/dev/null)

    if [ $? -eq 0 ] && [ -n "$BOOTSTRAP_RESPONSE" ]; then
        # Try to get IAM authentication brokers first using grep
        BOOTSTRAP_BROKERS=$(echo "$BOOTSTRAP_RESPONSE" | grep -o
 '"BootstrapBrokerStringSaslIam": "[^"]*' | cut -d'"' -f4)
        if [ -n "$BOOTSTRAP_BROKERS" ]; then
            AUTH_METHOD="IAM"
        else
            # Fall back to TLS authentication
            BOOTSTRAP_BROKERS=$(echo "$BOOTSTRAP_RESPONSE" | grep -o
 '"BootstrapBrokerStringTls": "[^"]*' | cut -d'"' -f4)
            if [ -n "$BOOTSTRAP_BROKERS" ]; then
                AUTH_METHOD="TLS"
            fi
        fi
    fi

    RETRY_COUNT=$((RETRY_COUNT + 1))
```

```
    if [ "$RETRY_COUNT" -ge "$MAX_RETRIES" ]; then
        echo "Warning: Could not get bootstrap brokers after $MAX_RETRIES
 attempts."
        echo "You may need to manually retrieve them later using:"
        echo "aws kafka get-bootstrap-brokers --cluster-arn $CLUSTER_ARN"
        BOOTSTRAP_BROKERS="BOOTSTRAP_BROKERS_NOT_AVAILABLE"
        AUTH_METHOD="UNKNOWN"
        break
    fi

    if [ -z "$BOOTSTRAP_BROKERS" ] || [ "$BOOTSTRAP_BROKERS" = "None" ]; then
        echo "Bootstrap brokers not available yet. Retrying in 30 seconds...
 (Attempt $RETRY_COUNT/$MAX_RETRIES)"
        sleep 30
    fi
done

echo "Bootstrap brokers: $BOOTSTRAP_BROKERS"
echo "Authentication method: $AUTH_METHOD"
echo "============================================="

# Create setup script for the client machine
echo "Creating setup script for the client machine"
cat > setup_client.sh << 'EOF'
#!/bin/bash

# Set up logging
LOG_FILE="client_setup_$(date +%Y%m%d_%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting client setup"
echo "============================================="

# Install Java
echo "Installing Java"
sudo yum -y install java-11

# Set environment variables
echo "Setting up environment variables"
export KAFKA_VERSION="3.6.0"
echo "KAFKA_VERSION=$KAFKA_VERSION"

# Download and extract Apache Kafka
```

```
echo "Downloading Apache Kafka"
wget https://archive.apache.org/dist/kafka/$KAFKA_VERSION/kafka_2.13-
$KAFKA_VERSION.tgz
if [ $? -ne 0 ]; then
    echo "Failed to download Kafka. Trying alternative mirror..."
    wget https://www.apache.org/dyn/closer.cgi?path=/kafka/$KAFKA_VERSION/
kafka_2.13-$KAFKA_VERSION.tgz
fi

echo "Extracting Kafka"
tar -xzf kafka_2.13-$KAFKA_VERSION.tgz
export KAFKA_ROOT=$(pwd)/kafka_2.13-$KAFKA_VERSION
echo "KAFKA_ROOT=$KAFKA_ROOT"

# Download the MSK IAM authentication package (needed for both IAM and TLS)
echo "Downloading MSK IAM authentication package"
cd $KAFKA_ROOT/libs
wget https://github.com/aws/aws-msk-iam-auth/releases/latest/download/aws-msk-
iam-auth-1.1.6-all.jar
if [ $? -ne 0 ]; then
    echo "Failed to download specific version. Trying to get latest version..."
    LATEST_VERSION=$(curl -s https://api.github.com/repos/aws/aws-msk-iam-auth/
releases/latest | grep -o '"tag_name": "[^"]*' | cut -d'"' -f4)
    wget https://github.com/aws/aws-msk-iam-auth/releases/download/
$LATEST_VERSION/aws-msk-iam-auth-$LATEST_VERSION-all.jar
    if [ $? -ne 0 ]; then
        echo "Failed to download IAM auth package. Please check the URL and try
 again."
        exit 1
    fi
    export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-$LATEST_VERSION-all.jar
else
    export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-1.1.6-all.jar
fi
echo "CLASSPATH=$CLASSPATH"

# Create client properties file based on authentication method
echo "Creating client properties file"
cd $KAFKA_ROOT/config

# The AUTH_METHOD_PLACEHOLDER will be replaced by the script
AUTH_METHOD="AUTH_METHOD_PLACEHOLDER"

if [ "$AUTH_METHOD" = "IAM" ]; then
```

```
    echo "Configuring for IAM authentication"
    cat > client.properties << 'EOT'
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required;
sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandler
EOT
elif [ "$AUTH_METHOD" = "TLS" ]; then
    echo "Configuring for TLS authentication"
    cat > client.properties << 'EOT'
security.protocol=SSL
EOT
else
    echo "Unknown authentication method. Creating basic TLS configuration."
    cat > client.properties << 'EOT'
security.protocol=SSL
EOT
fi

echo "Client setup completed"
echo "============================================="

# Create a script to set up environment variables
cat > ~/setup_env.sh << 'EOT'
#!/bin/bash
export KAFKA_VERSION="3.6.0"
export KAFKA_ROOT=~/kafka_2.13-$KAFKA_VERSION
export CLASSPATH=$KAFKA_ROOT/libs/aws-msk-iam-auth-1.1.6-all.jar
export BOOTSTRAP_SERVER="BOOTSTRAP_SERVER_PLACEHOLDER"
export AUTH_METHOD="AUTH_METHOD_PLACEHOLDER"

echo "Environment variables set:"
echo "KAFKA_VERSION=$KAFKA_VERSION"
echo "KAFKA_ROOT=$KAFKA_ROOT"
echo "CLASSPATH=$CLASSPATH"
echo "BOOTSTRAP_SERVER=$BOOTSTRAP_SERVER"
echo "AUTH_METHOD=$AUTH_METHOD"
EOT

chmod +x ~/setup_env.sh

echo "Created environment setup script: ~/setup_env.sh"
echo "Run 'source ~/setup_env.sh' to set up your environment"
EOF
```

```
# Replace placeholders in the setup script
if [ -n "$BOOTSTRAP_BROKERS" ] && [ "$BOOTSTRAP_BROKERS" != "None" ] &&
 [ "$BOOTSTRAP_BROKERS" != "BOOTSTRAP_BROKERS_NOT_AVAILABLE" ]; then
    sed -i "s|BOOTSTRAP_SERVER_PLACEHOLDER|$BOOTSTRAP_BROKERS|g" setup_client.sh
else
    # If bootstrap brokers are not available, provide instructions to get them
    sed -i "s|BOOTSTRAP_SERVER_PLACEHOLDER|\$(aws kafka get-bootstrap-brokers
 --cluster-arn $CLUSTER_ARN --query 'BootstrapBrokerStringTls' --output text)|g"
 setup_client.sh
fi

# Replace auth method placeholder
sed -i "s|AUTH_METHOD_PLACEHOLDER|$AUTH_METHOD|g" setup_client.sh

echo "Setup script created"
echo "============================================="

# Display summary of created resources
echo ""
echo "============================================="
echo "RESOURCE SUMMARY"
echo "============================================="
echo "MSK Cluster ARN: $CLUSTER_ARN"
echo "MSK Cluster Name: $CLUSTER_NAME"
echo "Authentication Method: $AUTH_METHOD"
echo "IAM Policy ARN: $POLICY_ARN"
echo "IAM Role Name: $ROLE_NAME"
echo "IAM Instance Profile: $INSTANCE_PROFILE_NAME"
echo "Client Security Group: $CLIENT_SG_ID"
echo "EC2 Instance ID: $INSTANCE_ID"
echo "EC2 Instance Type: $SELECTED_INSTANCE_TYPE"
echo "EC2 Instance DNS: $CLIENT_DNS"
echo "Key Pair: $KEY_NAME (saved to $KEY_FILE)"
echo "Bootstrap Brokers: $BOOTSTRAP_BROKERS"
echo "============================================="
echo ""

# Instructions for connecting to the instance and setting up the client
echo "NEXT STEPS:"
echo "1. Connect to your EC2 instance:"
echo "   ssh -i $KEY_FILE ec2-user@$CLIENT_DNS"
echo ""
echo "2. Upload the setup script to your instance:"
```

```
echo "   scp -i $KEY_FILE setup_client.sh ec2-user@$CLIENT_DNS:~/"
echo ""
echo "3. Run the setup script on your instance:"
echo "   ssh -i $KEY_FILE ec2-user@$CLIENT_DNS 'chmod +x ~/setup_client.sh && ~/
setup_client.sh'"
echo ""
echo "4. Source the environment setup script:"
echo "   source ~/setup_env.sh"
echo ""

# Provide different instructions based on authentication method
if [ "$AUTH_METHOD" = "IAM" ]; then
    echo "5. Create a Kafka topic (using IAM authentication):"
    echo "   \$KAFKA_ROOT/bin/kafka-topics.sh --create \\"
    echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
    echo "     --command-config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --replication-factor 3 \\"
    echo "     --partitions 1 \\"
    echo "     --topic MSKTutorialTopic"
    echo ""
    echo "6. Start a producer:"
    echo "   \$KAFKA_ROOT/bin/kafka-console-producer.sh \\"
    echo "     --broker-list \$BOOTSTRAP_SERVER \\"
    echo "     --producer.config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --topic MSKTutorialTopic"
    echo ""
    echo "7. Start a consumer:"
    echo "   \$KAFKA_ROOT/bin/kafka-console-consumer.sh \\"
    echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
    echo "     --consumer.config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --topic MSKTutorialTopic \\"
    echo "     --from-beginning"
elif [ "$AUTH_METHOD" = "TLS" ]; then
    echo "5. Create a Kafka topic (using TLS authentication):"
    echo "   \$KAFKA_ROOT/bin/kafka-topics.sh --create \\"
    echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
    echo "     --command-config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --replication-factor 3 \\"
    echo "     --partitions 1 \\"
    echo "     --topic MSKTutorialTopic"
    echo ""
    echo "6. Start a producer:"
    echo "   \$KAFKA_ROOT/bin/kafka-console-producer.sh \\"
    echo "     --broker-list \$BOOTSTRAP_SERVER \\"
```

```
    echo "     --producer.config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --topic MSKTutorialTopic"
    echo ""
    echo "7. Start a consumer:"
    echo "   \$KAFKA_ROOT/bin/kafka-console-consumer.sh \\"
    echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
    echo "     --consumer.config \$KAFKA_ROOT/config/client.properties \\"
    echo "     --topic MSKTutorialTopic \\"
    echo "     --from-beginning"
else
    echo "5. Manually retrieve bootstrap brokers and configure authentication:"
    echo "   aws kafka get-bootstrap-brokers --cluster-arn $CLUSTER_ARN"
fi

echo ""
echo "8. Verify the topic was created:"
echo "   \$KAFKA_ROOT/bin/kafka-topics.sh --list \\"
echo "     --bootstrap-server \$BOOTSTRAP_SERVER \\"
echo "     --command-config \$KAFKA_ROOT/config/client.properties"
echo "============================================="
echo ""

# Ask if user wants to clean up resources
echo ""
echo "=========================================="
echo "CLEANUP CONFIRMATION"
echo "=========================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ $CLEANUP_CHOICE =~ ^[Yy]$ ]]; then
    cleanup_resources
    echo "All resources have been cleaned up."
else
    echo "Resources will not be cleaned up. You can manually clean them up
 later."
    echo "To clean up resources, run this script again and choose 'y' when
 prompted."
fi

echo "Script completed successfully!"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AddRoleToInstanceProfile
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress
  - CreateCluster
  - CreateInstanceProfile
  - CreateKeyPair
  - CreatePolicy
  - CreateRole
  - CreateSecurityGroup
  - DeleteCluster
  - DeleteInstanceProfile
  - DeleteKeyPair
  - DeletePolicy
  - DeleteRole
  - DeleteSecurityGroup
  - DescribeAvailabilityZones
  - DescribeCluster
  - DescribeImages
  - DescribeInstanceTypeOfferings
  - DescribeInstances
  - DescribeKeyPairs
  - DescribeSecurityGroups
  - DescribeSubnets
  - DescribeVpcs
  - DetachRolePolicy
  - GetBootstrapBrokers
  - GetCallerIdentity
  - GetInstanceProfile
  - GetPolicy

- GetRole
- RemoveRoleFromInstanceProfile
- RevokeSecurityGroupEgress
- RevokeSecurityGroupIngress
- RunInstances
- TerminateInstances
- Attendi

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva alle notifiche push

L'esempio di codice seguente mostra come:

- Creazione di un'applicazione
- Abilita i canali di notifica push
- Invia una notifica push
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS End User Messaging Push Getting Started Script
# This script creates an AWS End User Messaging Push application and demonstrates
```

```
# how to enable push notification channels and send a test message.
#
# Prerequisites:
# - AWS CLI installed and configured
# - Appropriate IAM permissions for Pinpoint operations
#
# Usage: ./aws-end-user-messaging-gs.sh [--auto-cleanup]

set -uo pipefail

# Security: Set secure umask for created files
umask 0077

# Set up logging
LOG_DIR="./aws-eump-logs"
mkdir -p "$LOG_DIR"
LOG_FILE="$LOG_DIR/aws-end-user-messaging-push-script-$(date +%Y%m%d-%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting AWS End User Messaging Push setup script..."
echo "Logging to $LOG_FILE"
echo "Timestamp: $(date)"

# Security: Track created resources for cleanup
declare -a TEMP_FILES=()
declare -a AWS_RESOURCES=()

# Cleanup function with improved security
cleanup() {
    local exit_code=$?
    echo "Cleaning up temporary resources..."

    # Remove temporary files securely
    for temp_file in "${TEMP_FILES[@]+"${TEMP_FILES[@]}"}"; do
        if [ -f "$temp_file" ]; then
            shred -vfz -n 3 "$temp_file" 2>/dev/null || rm -f "$temp_file"
        fi
    done

    # Optionally delete AWS resources
    if [ "${DELETE_AWS_RESOURCES:-false}" = "true" ]; then
        for resource in "${AWS_RESOURCES[@]+"${AWS_RESOURCES[@]}"}"; do
            echo "Deleting AWS resource: $resource"
            aws pinpoint delete-app --application-id "$resource" 2>/dev/null || \
```

```
                echo "Warning: Failed to delete application $resource"
        done
    fi

    exit "$exit_code"
}

trap cleanup EXIT INT TERM

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2
    local ignore_error=${3:-false}

    if echo "$output" | grep -qi "error\|exception\|fail"; then
        echo "ERROR: Command failed: $cmd" >&2
        echo "Error details: $output" >&2

        if [ "$ignore_error" = "true" ]; then
            echo "Ignoring error and continuing..." >&2
            return 1
        else
            return 2
        fi
    fi

    return 0
}

# Function to validate AWS CLI is configured
validate_aws_cli() {
    echo "Validating AWS CLI configuration..."

    # Check if AWS CLI is installed
    if ! command -v aws &> /dev/null; then
        echo "ERROR: AWS CLI is not installed. Please install it first." >&2
        echo "Visit: https://docs.aws.amazon.com/cli/latest/userguide/getting-
started-install.html" >&2
        return 1
    fi

    # Check AWS CLI version
    AWS_VERSION=$(aws --version 2>&1 | head -n1)
```

```
    echo "AWS CLI version: $AWS_VERSION"

    # Verify credentials are set (check for credential env vars or config file)
    if ! aws sts get-caller-identity &> /dev/null; then
        echo "ERROR: AWS CLI credentials are not configured or invalid." >&2
        echo "Please configure credentials via environment variables, credential
 file, or 'aws configure'" >&2
        return 1
    fi

    # Get current AWS identity and region
    CALLER_IDENTITY=$(aws sts get-caller-identity)
    CURRENT_REGION=$(aws configure get region 2>/dev/null || echo "us-east-1")
    echo "AWS CLI configured for:"
    echo "$CALLER_IDENTITY"
    echo "Current region: $CURRENT_REGION"
    echo ""

    return 0
}

# Function to check if jq is available for JSON parsing
check_json_tools() {
    if command -v jq &> /dev/null; then
        USE_JQ=true
        echo "jq is available for JSON parsing"
    else
        USE_JQ=false
        echo "jq is not available, using grep for JSON parsing"
        echo "Consider installing jq for better JSON handling: https://
stedolan.github.io/jq/"
    fi
}

# Function to extract JSON values safely
extract_json_value() {
    local json=$1
    local key=$2

    if [ "$USE_JQ" = "true" ]; then
        echo "$json" | jq -r ".ApplicationResponse.$key // empty" 2>/dev/null ||
 echo ""
    else
        # Fallback to grep method with better validation
```

```
        echo "$json" | grep -o "\"$key\": \"[^\"]*" | cut -d'"' -f4 | head -n1 ||
 echo ""
    fi
}

# Function to validate required IAM permissions
validate_permissions() {
    echo "Validating IAM permissions..."

    # Test basic Pinpoint permissions
    if ! aws pinpoint get-apps > /dev/null 2>&1; then
        echo "WARNING: Unable to list Pinpoint applications." >&2
        echo "Please ensure you have appropriate IAM permissions for Pinpoint
 operations." >&2
        echo "Required permissions:" >&2
        echo "  - mobiletargeting:GetApps" >&2
        echo "  - mobiletargeting:CreateApp" >&2
        echo "  - mobiletargeting:DeleteApp" >&2
        echo "  - mobiletargeting:UpdateGcmChannel" >&2
        echo "  - mobiletargeting:UpdateApnsChannel" >&2
        echo "  - mobiletargeting:SendMessages" >&2
    else
        echo "Basic Pinpoint permissions validated."
    fi
}

# Function to validate input parameters
validate_input() {
    local app_name=$1

    # Validate app name length and characters
    if [ ${#app_name} -gt 64 ]; then
        echo "ERROR: Application name exceeds maximum length of 64 characters"
 >&2
        return 1
    fi

    if ! [[ "$app_name" =~ ^[a-zA-Z0-9_-]+$ ]]; then
        echo "ERROR: Application name contains invalid characters" >&2
        return 1
    fi

    return 0
}
```

```
# Function to create secure temporary files
create_temp_file() {
    local temp_file
    temp_file=$(mktemp) || {
        echo "ERROR: Failed to create temporary file" >&2
        return 1
    }
    chmod 600 "$temp_file"
    TEMP_FILES+=("$temp_file")
    echo "$temp_file"
}

# Validate prerequisites
if ! validate_aws_cli; then
    exit 1
fi

check_json_tools
validate_permissions

# Generate a random suffix for resource names to avoid conflicts
RANDOM_SUFFIX=$(LC_ALL=C tr -dc 'a-z0-9' < /dev/urandom | fold -w 8 | head -n1)
APP_NAME="PushNotificationApp-${RANDOM_SUFFIX}"

# Validate input
if ! validate_input "$APP_NAME"; then
    exit 1
fi

echo "Creating application with name: $APP_NAME"

# Step 1: Create an application
echo "Executing: aws pinpoint create-app --create-application-request Name=
${APP_NAME}"
CREATE_APP_OUTPUT=$(aws pinpoint create-app --create-application-request "Name=
${APP_NAME},tags={project=doc-smith,tutorial=aws-end-user-messaging-gs}" 2>&1)

if ! check_error "$CREATE_APP_OUTPUT" "create-app"; then
    exit 1
fi

echo "Application created successfully:"
echo "$CREATE_APP_OUTPUT"
```

```
# Extract the application ID from the output
APP_ID=$(extract_json_value "$CREATE_APP_OUTPUT" "Id")

if [ -z "$APP_ID" ] || [ "$APP_ID" = "null" ]; then
    echo "ERROR: Failed to extract application ID from output" >&2
    echo "Output was: $CREATE_APP_OUTPUT" >&2
    exit 1
fi

echo "Application ID: $APP_ID"
AWS_RESOURCES+=("$APP_ID")

# Step 2: Enable FCM (GCM) channel with a sample API key
echo ""
echo "========================================="
echo "ENABLING FCM (GCM) CHANNEL"
echo "========================================="
echo "Note: This is using a placeholder API key for demonstration purposes only."
echo "In a production environment, you should use your actual FCM API key from
 Firebase Console."
echo ""
echo "IMPORTANT: The following command will likely fail because we're using a
 placeholder API key."
echo "This is expected behavior for this demonstration script."

echo "Executing: aws pinpoint update-gcm-channel --application-id $APP_ID --gcm-
channel-request ..."
UPDATE_GCM_OUTPUT=$(aws pinpoint update-gcm-channel \
    --application-id "$APP_ID" \
    --gcm-channel-request '{"Enabled": true, "ApiKey": "sample-fcm-api-key-for-
demo-only"}' 2>&1)

# We'll ignore this specific error since we're using a placeholder API key
if check_error "$UPDATE_GCM_OUTPUT" "update-gcm-channel" "true"; then
    echo "FCM channel enabled successfully:"
    echo "$UPDATE_GCM_OUTPUT"
else
    echo "As expected, FCM channel update failed with the placeholder API key."
    echo "Error details: $UPDATE_GCM_OUTPUT"
    echo ""
    echo "To enable FCM in production:"
    echo "1. Go to Firebase Console (https://console.firebase.google.com/)"
    echo "2. Create or select your project"
```

```
    echo "3. Go to Project Settings > Cloud Messaging"
    echo "4. Copy the Server Key"
    echo "5. Replace 'sample-fcm-api-key-for-demo-only' with your actual Server
 Key"
fi

# Step 3: Try to enable APNS channel (this will also fail without real
 certificates)
echo ""
echo "=========================================="
echo "ENABLING APNS CHANNEL (OPTIONAL)"
echo "=========================================="
echo "Attempting to enable APNS channel with placeholder certificate..."
echo "This will also fail without real APNS certificates, which is expected."

# Create a placeholder APNS configuration
echo "Executing: aws pinpoint update-apns-channel --application-id $APP_ID --
apns-channel-request ..."
UPDATE_APNS_OUTPUT=$(aws pinpoint update-apns-channel \
    --application-id "$APP_ID" \
    --apns-channel-request '{"Enabled": true, "Certificate": "placeholder-
certificate", "PrivateKey": "placeholder-private-key"}' 2>&1)

if check_error "$UPDATE_APNS_OUTPUT" "update-apns-channel" "true"; then
    echo "APNS channel enabled successfully:"
    echo "$UPDATE_APNS_OUTPUT"
else
    echo "As expected, APNS channel update failed with placeholder certificates."
    echo "Error details: $UPDATE_APNS_OUTPUT"
    echo ""
    echo "To enable APNS in production:"
    echo "1. Generate APNS certificates from Apple Developer Console"
    echo "2. Convert certificates to PEM format"
    echo "3. Use the actual certificate and private key in the update-apns-
channel command"
fi

# Step 4: Create message files for different platforms
echo ""
echo "=========================================="
echo "CREATING MESSAGE FILES"
echo "=========================================="

# Create FCM message file securely
```

```
GCM_MESSAGE_FILE=$(create_temp_file)
echo "Creating FCM message file..."
cat > "$GCM_MESSAGE_FILE" << 'EOF'
{
  "Addresses": {
    "SAMPLE-DEVICE-TOKEN-FCM": {
      "ChannelType": "GCM"
    }
  },
  "MessageConfiguration": {
    "GCMMessage": {
      "Action": "OPEN_APP",
      "Body": "Hello from AWS End User Messaging Push! This is an FCM
 notification.",
      "Priority": "normal",
      "SilentPush": false,
      "Title": "My First FCM Push Notification",
      "TimeToLive": 30,
      "Data": {
        "key1": "value1",
        "key2": "value2"
      }
    }
  }
}
EOF

# Create APNS message file securely
APNS_MESSAGE_FILE=$(create_temp_file)
echo "Creating APNS message file..."
cat > "$APNS_MESSAGE_FILE" << 'EOF'
{
  "Addresses": {
    "SAMPLE-DEVICE-TOKEN-APNS": {
      "ChannelType": "APNS"
    }
  },
  "MessageConfiguration": {
    "APNSMessage": {
      "Action": "OPEN_APP",
      "Body": "Hello from AWS End User Messaging Push! This is an APNS
 notification.",
      "Priority": "normal",
      "SilentPush": false,
```

```
      "Title": "My First APNS Push Notification",
      "TimeToLive": 30,
      "Badge": 1,
      "Sound": "default"
    }
  }
}
EOF

echo "Message files created:"
echo "- FCM message file (for FCM/Android)"
echo "- APNS message file (for APNS/iOS)"
echo ""
echo "Note: These messages use placeholder device tokens and will not actually be
 delivered."
echo "To send real messages, you would need to replace the sample device tokens
 with actual ones."

# Step 5: Demonstrate how to send messages (this will fail with placeholder
 tokens)
echo ""
echo "============================================"
echo "DEMONSTRATING MESSAGE SENDING"
echo "============================================"
echo "Attempting to send FCM message (will fail with placeholder token)..."

echo "Executing: aws pinpoint send-messages --application-id $APP_ID --message-
request file://<gcm-message>"
SEND_FCM_OUTPUT=$(aws pinpoint send-messages \
    --application-id "$APP_ID" \
    --message-request "file://$GCM_MESSAGE_FILE" 2>&1)

if check_error "$SEND_FCM_OUTPUT" "send-messages (FCM)" "true"; then
    echo "FCM message sent successfully:"
    echo "$SEND_FCM_OUTPUT"
else
    echo "As expected, FCM message sending failed with placeholder token."
    echo "Error details: $SEND_FCM_OUTPUT"
fi

echo ""
echo "Attempting to send APNS message (will fail with placeholder token)..."
```

```
echo "Executing: aws pinpoint send-messages --application-id $APP_ID --message-
request file://<apns-message>"
SEND_APNS_OUTPUT=$(aws pinpoint send-messages \
    --application-id "$APP_ID" \
    --message-request "file://$APNS_MESSAGE_FILE" 2>&1)

if check_error "$SEND_APNS_OUTPUT" "send-messages (APNS)" "true"; then
    echo "APNS message sent successfully:"
    echo "$SEND_APNS_OUTPUT"
else
    echo "As expected, APNS message sending failed with placeholder token."
    echo "Error details: $SEND_APNS_OUTPUT"
fi

# Step 6: Show application details
echo ""
echo "========================================="
echo "APPLICATION DETAILS"
echo "========================================="
echo "Retrieving application details..."

echo "Executing: aws pinpoint get-app --application-id $APP_ID"
GET_APP_OUTPUT=$(aws pinpoint get-app --application-id "$APP_ID" 2>&1)
if check_error "$GET_APP_OUTPUT" "get-app"; then
    echo "Application details:"
    echo "$GET_APP_OUTPUT"
fi

# Display summary of created resources
echo ""
echo "========================================="
echo "RESOURCES CREATED"
echo "========================================="
echo "AWS Resources:"
for resource in "${AWS_RESOURCES[@]+"${AWS_RESOURCES[@]}"}"; do
    echo "- Application: $resource"
done

echo ""
echo "Files created:"
echo "- $LOG_FILE (script log)"

# Auto-cleanup information
echo ""
```

```
echo "============================================="
echo "CLEANUP INFORMATION"
echo "============================================="
echo "This script created AWS resources that may incur charges."
echo "AWS resources will be automatically cleaned up on script exit."
echo ""
echo "To manually delete resources later, use:"
echo "  aws pinpoint delete-app --application-id $APP_ID"

# Set flag to delete AWS resources on cleanup
DELETE_AWS_RESOURCES=true

echo ""
echo "============================================="
echo "SCRIPT COMPLETED SUCCESSFULLY"
echo "============================================="
echo "This script demonstrated:"
echo "1. Creating an AWS End User Messaging Push application"
echo "2. Attempting to enable FCM and APNS channels (with placeholder
 credentials)"
echo "3. Creating message templates for different platforms"
echo "4. Demonstrating message sending commands (with placeholder tokens)"
echo "5. Retrieving application details"
echo "6. Proper cleanup of resources"
echo ""
echo "Security best practices implemented:"
echo "- Secure temporary file handling with restricted permissions"
echo "- Input validation for application names"
echo "- Error handling with proper exit codes"
echo "- Credential validation before AWS operations"
echo "- Automatic cleanup on script exit"
echo "- Secure file destruction for sensitive data"
echo ""
echo "For production use:"
echo "- Replace placeholder API keys with real FCM server keys"
echo "- Replace placeholder certificates with real APNS certificates"
echo "- Replace placeholder device tokens with real device tokens"
echo "- Use AWS IAM roles instead of long-term credentials"
echo "- Implement comprehensive error handling"
echo "- Store sensitive credentials in AWS Secrets Manager or Parameter Store"
echo "- Enable CloudTrail for audit logging"
echo "- Use VPC endpoints for private AWS API access"
echo ""
echo "Log file: $LOG_FILE"
```

```
echo "Script completed at: $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - CreateApp
  - DeleteApp
  - GetApp
  - GetApps
  - GetCallerIdentity
  - SendMessages
  - UpdateApnsChannel
  - UpdateGcmChannel

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva ai motori di ricerca e analisi

L'esempio di codice seguente mostra come:

- Crea un dominio di OpenSearch servizio
- Carica i dati sul tuo dominio
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Amazon OpenSearch Service Getting Started Script - Version 8 Fixed
# This script creates an OpenSearch domain, uploads data, searches documents, and
 cleans up resources
# Based on the tested and working 4-tutorial-final.md

# SECURITY IMPROVEMENTS IN THIS VERSION:
# 1. Removed hardcoded passwords - now generated securely
# 2. Improved access policy to use principle of least privilege
# 3. Added credential masking in logs
# 4. Removed credentials from command output
# 5. Added input validation for AWS region
# 6. Improved error handling and validation
# 7. Added secure temporary file handling
# 8. Removed unnecessary curl insecurity flags
# 9. Added better secret management practices
# 10. Improved resource tagging for better governance

set -e  # Exit on any error

# Set up logging with restricted permissions
LOG_FILE="opensearch_tutorial_v8_fixed.log"
touch "$LOG_FILE"
chmod 600 "$LOG_FILE"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon OpenSearch Service tutorial script v8-fixed at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Track if domain was successfully created
DOMAIN_CREATED=false
DOMAIN_ACTIVE=false

# Secure temporary directory
TEMP_DIR=$(mktemp -d)
trap 'rm -rf "$TEMP_DIR"' EXIT
chmod 700 "$TEMP_DIR"

# Error handling function
handle_error() {
    echo "ERROR: $1"
    echo "Attempting to clean up resources..."
```

```
    cleanup_resources
    exit 1
}

# Function to clean up resources
cleanup_resources() {
    echo "Cleaning up resources..."

    if [[ "$DOMAIN_CREATED" == "true" ]]; then
        echo "Checking if domain $DOMAIN_NAME exists before attempting to
 delete..."

        # Check if domain exists before trying to delete
        if aws opensearch describe-domain --domain-name "$DOMAIN_NAME" > /dev/
null 2>&1; then
            echo "Domain $DOMAIN_NAME exists. Proceeding with deletion."
            aws opensearch delete-domain --domain-name "$DOMAIN_NAME"
            echo "Domain deletion initiated. This may take several minutes to
 complete."
        else
            echo "Domain $DOMAIN_NAME does not exist or is not accessible. No
 deletion needed."
        fi
    else
        echo "No domain was successfully created. Nothing to clean up."
    fi

    # Securely clean up sensitive files
    if [[ -d "$TEMP_DIR" ]]; then
        shred -vfz -n 3 "$TEMP_DIR"/* 2>/dev/null || true
        rm -rf "$TEMP_DIR"
    fi
}

# Generate a random identifier for resource names to avoid conflicts
RANDOM_ID=$(openssl rand -hex 4)
DOMAIN_NAME="movies-${RANDOM_ID}"
MASTER_USER="master-user"

# Generate secure password using openssl instead of hardcoding
MASTER_PASSWORD=$(openssl rand -base64 32 | tr -d "=+/" | cut -c1-25)

# Mask password in logs
echo "Using domain name: $DOMAIN_NAME"
```

```
echo "Using master username: $MASTER_USER"
echo "Using master password: [REDACTED]"

# Get AWS account ID (matches tutorial)
echo "Retrieving AWS account ID..."
ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
if [[ $? -ne 0 ]] || [[ -z "$ACCOUNT_ID" ]]; then
    handle_error "Failed to retrieve AWS account ID. Please check your AWS
 credentials."
fi
echo "AWS Account ID: $ACCOUNT_ID"

# Get current region (matches tutorial)
echo "Retrieving current AWS region..."
AWS_REGION=$(aws configure get region)
if [[ -z "$AWS_REGION" ]]; then
    AWS_REGION="us-east-1"
    echo "No region found in AWS config, defaulting to $AWS_REGION"
else
    echo "Using AWS region: $AWS_REGION"
fi

# Validate AWS region format
if ! [[ "$AWS_REGION" =~ ^[a-z]{2}-[a-z]+-[0-9]$ ]]; then
    handle_error "Invalid AWS region format: $AWS_REGION"
fi

# Step 1: Create an OpenSearch Service Domain
echo "Creating OpenSearch Service domain..."
echo "This may take 15-30 minutes to complete."

# SECURITY IMPROVED: Use least privilege access policy
# This policy restricts access to specific actions and should be further
 restricted to specific principals in production
# Create the domain using CloudFormation (handles the 15-20 min creation wait)
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
STACK_NAME="tutorial-opensearch-${RANDOM_ID}"

echo "Creating domain $DOMAIN_NAME via CloudFormation..."
echo "This typically takes 15-20 minutes. CloudFormation will wait for
 completion."
aws cloudformation deploy \
  --template-file "$SCRIPT_DIR/cfn-opensearch-domain.yaml" \
  --stack-name "$STACK_NAME" \
```

```
  --parameter-overrides \
    DomainName="$DOMAIN_NAME" \
    MasterUserName="$MASTER_USER" \
    MasterUserPassword="$MASTER_PASSWORD" \
  --tags project=doc-smith tutorial=opensearch-service-gs \
  --no-fail-on-empty-changeset 2>&1 || handle_error "CloudFormation stack
 creation failed"

DOMAIN_CREATED=true
echo "Domain created successfully via CloudFormation."

# Get domain endpoint from stack outputs
DOMAIN_ENDPOINT=$(aws cloudformation describe-stacks \
  --stack-name "$STACK_NAME" \
  --query 'Stacks[0].Outputs[?OutputKey==`DomainEndpoint`].OutputValue' \
  --output text)

if [[ -z "$DOMAIN_ENDPOINT" ]] || [[ "$DOMAIN_ENDPOINT" == "None" ]]; then
    handle_error "Failed to get domain endpoint from CloudFormation outputs"
fi

echo "Domain endpoint: $DOMAIN_ENDPOINT"

# Wait for fine-grained access control to be fully ready
echo "Waiting for fine-grained access control to initialize..."
sleep 120

# Verify variables are set correctly (matches tutorial)
echo "Verifying configuration..."
echo "Domain endpoint: $DOMAIN_ENDPOINT"
echo "Master user: $MASTER_USER"
echo "Password set: $(if [ -n "$MASTER_PASSWORD" ]; then echo "Yes"; else echo
 "No"; fi)"

# Step 2: Upload Data to the Domain
echo "Preparing to upload data to the domain..."

# Create a file for the single document (matches tutorial exactly)
echo "Creating single document JSON file..."
SINGLE_MOVIE_FILE="$TEMP_DIR/single_movie.json"
cat > "$SINGLE_MOVIE_FILE" << 'EOF'
{
  "director": "Burton, Tim",
  "genre": ["Comedy","Sci-Fi"],
```

```
  "year": 1996,
  "actor": ["Jack Nicholson","Pierce Brosnan","Sarah Jessica Parker"],
  "title": "Mars Attacks!"
}
EOF
chmod 600 "$SINGLE_MOVIE_FILE"

# Create a file for bulk documents (matches tutorial exactly)
echo "Creating bulk documents JSON file..."
BULK_MOVIES_FILE="$TEMP_DIR/bulk_movies.json"
cat > "$BULK_MOVIES_FILE" << 'EOF'
{ "index" : { "_index": "movies", "_id" : "2" } }
{"director": "Frankenheimer, John", "genre": ["Drama", "Mystery", "Thriller",
 "Crime"], "year": 1962, "actor": ["Lansbury, Angela", "Sinatra, Frank", "Leigh,
 Janet", "Harvey, Laurence", "Silva, Henry", "Frees, Paul", "Gregory, James",
 "Bissell, Whit", "McGiver, John", "Parrish, Leslie", "Edwards, James", "Flowers,
 Bess", "Dhiegh, Khigh", "Payne, Julie", "Kleeb, Helen", "Gray, Joe", "Nalder,
 Reggie", "Stevens, Bert", "Masters, Michael", "Lowell, Tom"], "title": "The
 Manchurian Candidate"}
{ "index" : { "_index": "movies", "_id" : "3" } }
{"director": "Baird, Stuart", "genre": ["Action", "Crime", "Thriller"], "year":
 1998, "actor": ["Downey Jr., Robert", "Jones, Tommy Lee", "Snipes, Wesley",
 "Pantoliano, Joe", "Jacob, Irène", "Nelligan, Kate", "Roebuck, Daniel",
 "Malahide, Patrick", "Richardson, LaTanya", "Wood, Tom", "Kosik, Thomas",
 "Stellate, Nick", "Minkoff, Robert", "Brown, Spitfire", "Foster, Reese",
 "Spielbauer, Bruce", "Mukherji, Kevin", "Cray, Ed", "Fordham, David", "Jett,
 Charlie"], "title": "U.S. Marshals"}
{ "index" : { "_index": "movies", "_id" : "4" } }
{"director": "Ray, Nicholas", "genre": ["Drama", "Romance"], "year": 1955,
 "actor": ["Hopper, Dennis", "Wood, Natalie", "Dean, James", "Mineo, Sal",
 "Backus, Jim", "Platt, Edward", "Ray, Nicholas", "Hopper, William", "Allen,
 Corey", "Birch, Paul", "Hudson, Rochelle", "Doran, Ann", "Hicks, Chuck", "Leigh,
 Nelson", "Williams, Robert", "Wessel, Dick", "Bryar, Paul", "Sessions, Almira",
 "McMahon, David", "Peters Jr., House"], "title": "Rebel Without a Cause"}
EOF
chmod 600 "$BULK_MOVIES_FILE"

# Check if curl is installed
if ! command -v curl &> /dev/null; then
    echo "Warning: curl is not installed. Skipping data upload and search steps."
    echo "You can manually upload the data later using the commands in the
 tutorial."
else
    # IMPROVED: Test authentication with multiple approaches
```

```
    echo "Testing authentication with the OpenSearch domain..."
    echo "This test checks if fine-grained access control is ready for data
operations."

    # Create credentials file for secure handling
    CREDENTIALS_FILE="$TEMP_DIR/.credentials"
    echo "$MASTER_USER:$MASTER_PASSWORD" > "$CREDENTIALS_FILE"
    chmod 600 "$CREDENTIALS_FILE"

    # Test 1: Basic authentication with root endpoint
    echo "Testing basic authentication with root endpoint..."
    AUTH_TEST_RESULT=$(curl -s -w "\nHTTP_CODE:%{http_code}" \
        --netrc-file "$CREDENTIALS_FILE" \
        --request GET \
        "https://${DOMAIN_ENDPOINT}/" 2>&1)

    echo "Basic auth test result:"
    echo "$AUTH_TEST_RESULT"

    # Extract HTTP status code
    HTTP_CODE=$(echo "$AUTH_TEST_RESULT" | grep "HTTP_CODE:" | cut -d: -f2)

    # Function to check if HTTP code is 2xx
    is_success_code() {
        local code=$1
        if [[ "$code" =~ ^2[0-9][0-9]$ ]]; then
            return 0
        else
            return 1
        fi
    }

    # Check if basic authentication test was successful (200 or 2xx status codes)
    if is_success_code "$HTTP_CODE"; then
        echo "# Basic authentication test successful! (HTTP $HTTP_CODE)"
        AUTH_SUCCESS=true
        AUTH_METHOD="basic"
    else
        echo "Basic authentication failed with HTTP code: $HTTP_CODE"

        # Test 2: Try cluster health endpoint
        echo "Testing with cluster health endpoint..."
        HEALTH_TEST_RESULT=$(curl -s -w "\nHTTP_CODE:%{http_code}" \
            --netrc-file "$CREDENTIALS_FILE" \
```

```
            --request GET \
            "https://${DOMAIN_ENDPOINT}/_cluster/health" 2>&1)

        echo "Cluster health test result:"
        echo "$HEALTH_TEST_RESULT"

        HEALTH_HTTP_CODE=$(echo "$HEALTH_TEST_RESULT" | grep "HTTP_CODE:" | cut -
d: -f2)

        if is_success_code "$HEALTH_HTTP_CODE"; then
            echo "# Cluster health authentication test successful! (HTTP
 $HEALTH_HTTP_CODE)"
            AUTH_SUCCESS=true
            AUTH_METHOD="basic"
        else
            echo "Cluster health authentication also failed with HTTP code:
 $HEALTH_HTTP_CODE"

            # Check for specific error patterns
            if echo "$AUTH_TEST_RESULT" | grep -q "anonymous is not authorized";
 then
                echo "Error: Request is being treated as anonymous
 (authentication not working)"
            elif echo "$AUTH_TEST_RESULT" | grep -q "Unauthorized"; then
                echo "Error: Authentication credentials rejected"
            elif echo "$AUTH_TEST_RESULT" | grep -q "Forbidden"; then
                echo "Error: Authentication succeeded but access is forbidden"
            fi

            echo "Waiting additional time and retrying with exponential
 backoff..."

            # Retry authentication test with exponential backoff
            AUTH_RETRY_COUNT=0
            MAX_AUTH_RETRIES=5
            WAIT_TIME=60
            AUTH_SUCCESS=false

            while [[ $AUTH_RETRY_COUNT -lt $MAX_AUTH_RETRIES ]]; do
                echo "Retrying authentication test (attempt $((AUTH_RETRY_COUNT
+1))/$MAX_AUTH_RETRIES) after ${WAIT_TIME} seconds..."
                sleep $WAIT_TIME

                # Try both endpoints
```

```
                AUTH_TEST_RESULT=$(curl -s -w "\nHTTP_CODE:%{http_code}" \
                    --netrc-file "$CREDENTIALS_FILE" \
                    --request GET \
                    "https://${DOMAIN_ENDPOINT}/" 2>&1)

                HTTP_CODE=$(echo "$AUTH_TEST_RESULT" | grep "HTTP_CODE:" | cut -
d: -f2)

                echo "Retry result (HTTP $HTTP_CODE):"
                echo "$AUTH_TEST_RESULT"

                if is_success_code "$HTTP_CODE"; then
                    echo "# Authentication test successful after retry! (HTTP
 $HTTP_CODE)"
                    AUTH_SUCCESS=true
                    AUTH_METHOD="basic"
                    break
                fi

                # Also try cluster health
                HEALTH_TEST_RESULT=$(curl -s -w "\nHTTP_CODE:%{http_code}" \
                    --netrc-file "$CREDENTIALS_FILE" \
                    --request GET \
                    "https://${DOMAIN_ENDPOINT}/_cluster/health" 2>&1)

                HEALTH_HTTP_CODE=$(echo "$HEALTH_TEST_RESULT" | grep "HTTP_CODE:"
 | cut -d: -f2)

                if is_success_code "$HEALTH_HTTP_CODE"; then
                    echo "# Cluster health authentication successful after retry!
 (HTTP $HEALTH_HTTP_CODE)"
                    AUTH_SUCCESS=true
                    AUTH_METHOD="basic"
                    break
                fi

                AUTH_RETRY_COUNT=$((AUTH_RETRY_COUNT+1))
                # Exponential backoff: double the wait time each retry (max 10
 minutes)
                WAIT_TIME=$((WAIT_TIME * 2))
                if [[ $WAIT_TIME -gt 600 ]]; then
                    WAIT_TIME=600
                fi
            done
```

```
        fi
    fi

    # Proceed with data operations if authentication is working
    if [[ "$AUTH_SUCCESS" == "true" ]]; then
        echo "Authentication successful using $AUTH_METHOD method. Proceeding
 with data operations."

        # Upload single document (matches tutorial exactly)
        echo "Uploading single document..."
        UPLOAD_RESULT=$(curl -s -w "\nHTTP_CODE:%{http_code}" \
            --netrc-file "$CREDENTIALS_FILE" \
            --request PUT \
            --header 'Content-Type: application/json' \
            --data @"$SINGLE_MOVIE_FILE" \
            "https://${DOMAIN_ENDPOINT}/movies/_doc/1" 2>&1)

        echo "Upload response:"
        echo "$UPLOAD_RESULT"

        UPLOAD_HTTP_CODE=$(echo "$UPLOAD_RESULT" | grep "HTTP_CODE:" | cut -d: -
f2)
        if is_success_code "$UPLOAD_HTTP_CODE" && echo "$UPLOAD_RESULT" | grep -q
 '"result"'; then
            echo "# Single document uploaded successfully! (HTTP
 $UPLOAD_HTTP_CODE)"
        else
            echo "# Warning: Single document upload may have failed (HTTP
 $UPLOAD_HTTP_CODE)"
        fi

        # Upload bulk documents (matches tutorial exactly)
        echo "Uploading bulk documents..."
        BULK_RESULT=$(curl -s -w "\nHTTP_CODE:%{http_code}" \
            --netrc-file "$CREDENTIALS_FILE" \
            --request POST \
            --header 'Content-Type: application/x-ndjson' \
            --data-binary @"$BULK_MOVIES_FILE" \
            "https://${DOMAIN_ENDPOINT}/movies/_bulk" 2>&1)

        echo "Bulk upload response:"
        echo "$BULK_RESULT"

        BULK_HTTP_CODE=$(echo "$BULK_RESULT" | grep "HTTP_CODE:" | cut -d: -f2)
```

```
        if is_success_code "$BULK_HTTP_CODE" && echo "$BULK_RESULT" | grep -q
 '"errors": false'; then
            echo "# Bulk documents uploaded successfully! (HTTP $BULK_HTTP_CODE)"
        else
            echo "# Warning: Bulk document upload may have failed (HTTP
 $BULK_HTTP_CODE)"
        fi

        # Wait a moment for indexing
        echo "Waiting for documents to be indexed..."
        sleep 5

        # Step 3: Search Documents (matches tutorial exactly)
        echo "Searching for documents containing 'mars'..."
        SEARCH_RESULT=$(curl -s -w "\nHTTP_CODE:%{http_code}" \
            --netrc-file "$CREDENTIALS_FILE" \
            --request GET \
            "https://${DOMAIN_ENDPOINT}/movies/_search?q=mars&pretty=true" 2>&1)

        SEARCH_HTTP_CODE=$(echo "$SEARCH_RESULT" | grep "HTTP_CODE:" | cut -d: -
f2)
        echo "Search results for 'mars' (HTTP $SEARCH_HTTP_CODE):"
        echo "$SEARCH_RESULT"

        echo "Searching for documents containing 'rebel'..."
        REBEL_SEARCH=$(curl -s -w "\nHTTP_CODE:%{http_code}" \
            --netrc-file "$CREDENTIALS_FILE" \
            --request GET \
            "https://${DOMAIN_ENDPOINT}/movies/_search?q=rebel&pretty=true" 2>&1)

        REBEL_HTTP_CODE=$(echo "$REBEL_SEARCH" | grep "HTTP_CODE:" | cut -d: -f2)
        echo "Search results for 'rebel' (HTTP $REBEL_HTTP_CODE):"
        echo "$REBEL_SEARCH"

        # Verify search results
        if is_success_code "$SEARCH_HTTP_CODE" && echo "$SEARCH_RESULT" | grep -q
 '"hits"'; then
            echo "# Search functionality is working!"
        else
            echo "# Warning: Search may not be working properly."
        fi

    else
        echo ""
```

```
        echo "========================================="
        echo "AUTHENTICATION TROUBLESHOOTING"
        echo "========================================="
        echo "Authentication failed after all retries. This may be due to:"
        echo "1. Fine-grained access control not fully initialized (most common)"
        echo "2. Domain configuration issues"
        echo "3. Network connectivity issues"
        echo "4. AWS credentials or permissions issues"
        echo ""
        echo "DOMAIN CONFIGURATION DEBUG:"
        echo "Let's check the domain configuration..."

        # Debug domain configuration
        DOMAIN_CONFIG=$(aws opensearch describe-domain --domain-
name "$DOMAIN_NAME" --query 'DomainStatus.{AdvancedSecurityOptions:
 AdvancedSecurityOptions, AccessPolicies: AccessPolicies}' --output json 2>&1)
        echo "Domain configuration:"
        echo "$DOMAIN_CONFIG"

        echo ""
        echo "MANUAL TESTING COMMANDS:"
        echo "You can try these commands manually in 10-15 minutes:"
        echo ""
        echo "# Test basic authentication:"
        echo "curl --user \"${MASTER_USER}:[PASSWORD]\" \"https://
${DOMAIN_ENDPOINT}/\""
        echo ""
        echo "# Test cluster health:"
        echo "curl --user \"${MASTER_USER}:[PASSWORD]\" \"https://
${DOMAIN_ENDPOINT}/_cluster/health\""
        echo ""
        echo "# Upload single document:"
        echo "curl --user \"${MASTER_USER}:[PASSWORD]\" --request PUT --
header 'Content-Type: application/json' --data @single_movie.json \"https://
${DOMAIN_ENDPOINT}/movies/_doc/1\""
        echo ""
        echo "# Search for documents:"
        echo "curl --user \"${MASTER_USER}:[PASSWORD]\" \"https://
${DOMAIN_ENDPOINT}/movies/_search?q=mars&pretty=true\""
        echo ""
        echo "TROUBLESHOOTING TIPS:"
        echo "- Wait 10-15 more minutes and try the manual commands"
        echo "- Check AWS CloudTrail logs for authentication errors"
        echo "- Verify your AWS region is correct: $AWS_REGION"
```

```
        echo "- Ensure your AWS credentials have OpenSearch permissions"
        echo "- Try accessing OpenSearch Dashboards to verify the master user
 works"
        echo ""
        echo "Skipping data upload and search operations for now."
        echo "The domain is created and accessible via OpenSearch Dashboards."
    fi

    # Securely remove credentials file
    shred -vfz -n 3 "$CREDENTIALS_FILE" 2>/dev/null || true
fi

# Display OpenSearch Dashboards URL (matches tutorial)
echo ""
echo "==========================================="
echo "OPENSEARCH DASHBOARDS ACCESS"
echo "==========================================="
echo "OpenSearch Dashboards URL: https://${DOMAIN_ENDPOINT}/_dashboards/"
echo "Username: $MASTER_USER"
echo "Password: [REDACTED - stored securely]"
echo ""
echo "You can access OpenSearch Dashboards using the credentials provided during
 domain creation."
echo "If you uploaded data successfully, you can create an index pattern for
 'movies'."
echo ""

# Summary of created resources
echo ""
echo "==========================================="
echo "RESOURCES CREATED"
echo "==========================================="
echo "OpenSearch Domain Name: $DOMAIN_NAME"
echo "OpenSearch Domain Endpoint: $DOMAIN_ENDPOINT"
echo "AWS Region: $AWS_REGION"
echo "Master Username: $MASTER_USER"
echo ""
echo "ESTIMATED COST: ~$0.038/hour (~$0.91/day) until deleted"
echo ""
echo "Make sure to save the domain name and endpoint for future reference."
echo ""

# Auto-confirm cleanup: Yes, clean up all resources
echo ""
```

```
echo "============================================="
echo "CLEANUP CONFIRMATION"
echo "============================================="
echo "Automatically cleaning up all created resources..."
CLEANUP_CHOICE="y"

if [[ "${CLEANUP_CHOICE,,}" == "y" ]]; then
    echo "Cleaning up resources..."
    if [ -n "${STACK_NAME:-}" ]; then
        echo "Deleting CloudFormation stack $STACK_NAME..."
        aws cloudformation delete-stack --stack-name "$STACK_NAME"
        echo "# Stack deletion initiated. This may take several minutes."
        echo "  Monitor: aws cloudformation describe-stacks --stack-name
 $STACK_NAME"
    else
        aws opensearch delete-domain --domain-name "$DOMAIN_NAME" 2>/dev/null
        echo "# Domain deletion initiated."
    fi
else
    echo "Resources will NOT be deleted automatically."
    echo ""
    echo "To delete later:"
    echo "  aws cloudformation delete-stack --stack-name ${STACK_NAME:-tutorial-
opensearch-XXXX}"
    echo ""
    echo "# IMPORTANT: Keeping these resources will incur ongoing AWS charges!"
fi

# Clean up temporary files (handled by trap)
echo "Cleaning up temporary files..."

# Disable the trap since we're handling cleanup manually
trap - EXIT

# Final cleanup
rm -rf "$TEMP_DIR" 2>/dev/null || true

echo ""
echo "============================================="
echo "SCRIPT COMPLETED SUCCESSFULLY"
echo "============================================="
echo "Script completed at $(date)"
echo "All output has been logged to: $LOG_FILE"
echo ""
```

```
echo "Next steps:"
echo "1. Access OpenSearch Dashboards at: https://${DOMAIN_ENDPOINT}/
_dashboards/"
echo "2. Create visualizations and dashboards"
echo "3. Explore the OpenSearch API"
echo "4. Remember to delete resources when done to avoid charges"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - CreateDomain
  - DeleteDomain
  - DescribeDomain
  - GetCallerIdentity

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Guida introduttiva ai firewall per applicazioni Web

L'esempio di codice seguente mostra come:

- Creazione di un ACL Web
- Aggiungi una regola di corrispondenza delle stringhe
- Aggiungi regole gestite
- Configurazione della registrazione
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS WAF Getting Started Script
# This script creates a Web ACL with a string match rule and AWS Managed Rules,
# associates it with a CloudFront distribution, and then cleans up all resources.

set -euo pipefail

# Security: Restrict file permissions
umask 077

# Set up logging with secure file handling
LOG_FILE="waf-tutorial.log"
touch "$LOG_FILE"
chmod 600 "$LOG_FILE"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "================================================="
echo "AWS WAF Getting Started Tutorial"
echo "================================================="
echo "This script will create AWS WAF resources and associate"
echo "them with a CloudFront distribution."
echo ""

# Maximum number of retries for operations
MAX_RETRIES=3

# Function to handle errors securely
handle_error() {
    local error_msg="$1"
    echo "ERROR: $error_msg" >&2
    echo "Check the log file for details: $LOG_FILE" >&2
    cleanup_resources
    exit 1
}

# Function to validate AWS CLI JSON output
validate_json() {
    local json_string="$1"
    if ! echo "$json_string" | jq empty 2>/dev/null; then
        return 1
    fi
    return 0
```

```
}

# Function to safely extract JSON values
extract_json_value() {
    local json_string="$1"
    local key_path="$2"

    if ! validate_json "$json_string"; then
        return 1
    fi

    echo "$json_string" | jq -r "$key_path" 2>/dev/null || return 1
}

# Function to clean up resources securely
cleanup_resources() {
    echo ""
    echo "=================================================="
    echo "CLEANING UP RESOURCES"
    echo "=================================================="

    if [ -n "${DISTRIBUTION_ID:-}" ] && [ -n "${WEB_ACL_ARN:-}" ]; then
        echo "Disassociating Web ACL from CloudFront distribution..."
        local account_id
        account_id=$(aws sts get-caller-identity --query Account --output text
 2>/dev/null) || account_id=""

        if [ -z "$account_id" ]; then
            echo "Warning: Could not retrieve AWS account ID"
            return
        fi

        local disassociate_result
        disassociate_result=$(aws wafv2 disassociate-web-acl \
            --resource-arn "arn:aws:cloudfront::${account_id}:distribution/
${DISTRIBUTION_ID}" \
            --region us-east-1 2>&1) || true

        if echo "$disassociate_result" | grep -qi "error"; then
            echo "Warning: Failed to disassociate Web ACL: $disassociate_result"
        else
            echo "Web ACL disassociated successfully."
        fi
    fi
```

```
   if [ -n "${WEB_ACL_ID:-}" ] && [ -n "${WEB_ACL_NAME:-}" ]; then
       echo "Deleting Web ACL..."

       local get_result
       get_result=$(aws wafv2 get-web-acl \
           --name "$WEB_ACL_NAME" \
           --scope CLOUDFRONT \
           --id "$WEB_ACL_ID" \
           --region us-east-1 2>&1) || true

       if echo "$get_result" | grep -qi "error"; then
           echo "Warning: Failed to get Web ACL for deletion: $get_result"
           echo "You may need to manually delete the Web ACL using the AWS
Console."
       else
           local latest_token
           latest_token=$(extract_json_value "$get_result" '.WebACL.LockToken'
2>/dev/null) || latest_token=""

           if [ -n "$latest_token" ]; then
               local delete_result
               delete_result=$(aws wafv2 delete-web-acl \
                   --name "$WEB_ACL_NAME" \
                   --scope CLOUDFRONT \
                   --id "$WEB_ACL_ID" \
                   --lock-token "$latest_token" \
                   --region us-east-1 2>&1) || true

               if echo "$delete_result" | grep -qi "error"; then
                   echo "Warning: Failed to delete Web ACL: $delete_result"
                   echo "You may need to manually delete the Web ACL using the
AWS Console."
               else
                   echo "Web ACL deleted successfully."
               fi
           else
               echo "Warning: Could not extract lock token for deletion. You may
need to manually delete the Web ACL."
           fi
       fi
   fi

   echo "Cleanup process completed."
```

```
}

# Security: Trap EXIT to ensure cleanup on any exit
trap cleanup_resources EXIT

# Generate a random identifier for resource names using secure method
RANDOM_ID=$(openssl rand -hex 4) || handle_error "Failed to generate random ID"
WEB_ACL_NAME="MyWebACL-${RANDOM_ID}"
METRIC_NAME="MyWebACLMetrics-${RANDOM_ID}"

echo "Using Web ACL name: $WEB_ACL_NAME"

# Step 1: Create a Web ACL
echo ""
echo "=================================================="
echo "STEP 1: Creating Web ACL"
echo "=================================================="

local create_result
create_result=$(aws wafv2 create-web-acl \
    --name "$WEB_ACL_NAME" \
    --scope "CLOUDFRONT" \
    --default-action Allow={} \
    --visibility-config
 "SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=
$METRIC_NAME" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=aws-waf-gs \
    --region us-east-1 2>&1) || handle_error "Failed to create Web ACL"

if ! validate_json "$create_result"; then
    handle_error "Invalid JSON response from create-web-acl"
fi

# Extract Web ACL ID, ARN, and Lock Token from the response
WEB_ACL_ID=$(extract_json_value "$create_result" '.Summary.Id') || handle_error
 "Failed to extract Web ACL ID"
WEB_ACL_ARN=$(extract_json_value "$create_result" '.Summary.ARN') || handle_error
 "Failed to extract Web ACL ARN"
LOCK_TOKEN=$(extract_json_value "$create_result" '.Summary.LockToken') ||
 handle_error "Failed to extract Lock Token"

echo "Web ACL created successfully with ID: $WEB_ACL_ID"
echo "Lock Token: [REDACTED]"
```

```
# Step 2: Add a String Match Rule
echo ""
echo "=================================================="
echo "STEP 2: Adding String Match Rule"
echo "=================================================="

for ((i=1; i<=MAX_RETRIES; i++)); do
    echo "Attempt $i to add string match rule..."

    # Get the latest lock token before updating
    local get_result
    get_result=$(aws wafv2 get-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope CLOUDFRONT \
        --id "$WEB_ACL_ID" \
        --region us-east-1 2>&1) || true

    if echo "$get_result" | grep -qi "error"; then
        echo "Warning: Failed to get Web ACL for update: $get_result"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to get Web ACL after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    if ! validate_json "$get_result"; then
        echo "Warning: Invalid JSON response from get-web-acl"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Invalid JSON response after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    local latest_token
    latest_token=$(extract_json_value "$get_result" '.WebACL.LockToken' 2>/dev/
null) || true

    if [ -z "$latest_token" ]; then
        echo "Warning: Could not extract lock token for update"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to extract lock token after $MAX_RETRIES
 attempts"
```

```
        fi
        sleep 2
        continue
    fi

    local update_result
    update_result=$(aws wafv2 update-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope "CLOUDFRONT" \
        --id "$WEB_ACL_ID" \
        --lock-token "$latest_token" \
        --default-action Allow={} \
        --rules '[{
            "Name": "UserAgentRule",
            "Priority": 0,
            "Statement": {
                "ByteMatchStatement": {
                    "SearchString": "MyAgent",
                    "FieldToMatch": {
                        "SingleHeader": {
                            "Name": "user-agent"
                        }
                    },
                    "TextTransformations": [
                        {
                            "Priority": 0,
                            "Type": "NONE"
                        }
                    ],
                    "PositionalConstraint": "EXACTLY"
                }
            },
            "Action": {
                "Count": {}
            },
            "VisibilityConfig": {
                "SampledRequestsEnabled": true,
                "CloudWatchMetricsEnabled": true,
                "MetricName": "UserAgentRuleMetric"
            }
        }]' \
        --visibility-config
 "SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=
$METRIC_NAME" \
```

```
        --region us-east-1 2>&1) || true

    if echo "$update_result" | grep -qi "WAFOptimisticLockException"; then
        echo "Optimistic lock exception encountered. Will retry with new lock
 token."
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to add string match rule after $MAX_RETRIES
 attempts"
        fi
        sleep 2
        continue
    elif echo "$update_result" | grep -qi "error"; then
        handle_error "Failed to add string match rule: $update_result"
    else
        echo "String match rule added successfully."
        break
    fi
done

# Step 3: Add AWS Managed Rules
echo ""
echo "=================================================="
echo "STEP 3: Adding AWS Managed Rules"
echo "=================================================="

for ((i=1; i<=MAX_RETRIES; i++)); do
    echo "Attempt $i to add AWS Managed Rules..."

    # Get the latest lock token before updating
    local get_result
    get_result=$(aws wafv2 get-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope CLOUDFRONT \
        --id "$WEB_ACL_ID" \
        --region us-east-1 2>&1) || true

    if echo "$get_result" | grep -qi "error"; then
        echo "Warning: Failed to get Web ACL for update: $get_result"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to get Web ACL after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi
```

```
    if ! validate_json "$get_result"; then
        echo "Warning: Invalid JSON response from get-web-acl"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Invalid JSON response after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    local latest_token
    latest_token=$(extract_json_value "$get_result" '.WebACL.LockToken' 2>/dev/
null) || true

    if [ -z "$latest_token" ]; then
        echo "Warning: Could not extract lock token for update"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to extract lock token after $MAX_RETRIES
 attempts"
        fi
        sleep 2
        continue
    fi

    local update_result
    update_result=$(aws wafv2 update-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope "CLOUDFRONT" \
        --id "$WEB_ACL_ID" \
        --lock-token "$latest_token" \
        --default-action Allow={} \
        --rules '[{
            "Name": "UserAgentRule",
            "Priority": 0,
            "Statement": {
                "ByteMatchStatement": {
                    "SearchString": "MyAgent",
                    "FieldToMatch": {
                        "SingleHeader": {
                            "Name": "user-agent"
                        }
                    },
                    "TextTransformations": [
                        {
```

```
                            "Priority": 0,
                            "Type": "NONE"
                        }
                    ],
                    "PositionalConstraint": "EXACTLY"
                }
            },
            "Action": {
                "Count": {}
            },
            "VisibilityConfig": {
                "SampledRequestsEnabled": true,
                "CloudWatchMetricsEnabled": true,
                "MetricName": "UserAgentRuleMetric"
            }
        },
        {
            "Name": "AWS-AWSManagedRulesCommonRuleSet",
            "Priority": 1,
            "Statement": {
                "ManagedRuleGroupStatement": {
                    "VendorName": "AWS",
                    "Name": "AWSManagedRulesCommonRuleSet",
                    "ExcludedRules": []
                }
            },
            "OverrideAction": {
                "Count": {}
            },
            "VisibilityConfig": {
                "SampledRequestsEnabled": true,
                "CloudWatchMetricsEnabled": true,
                "MetricName": "AWS-AWSManagedRulesCommonRuleSet"
            }
        }]' \
        --visibility-config
 "SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=
$METRIC_NAME" \
        --region us-east-1 2>&1) || true

    if echo "$update_result" | grep -qi "WAFOptimisticLockException"; then
        echo "Optimistic lock exception encountered. Will retry with new lock
 token."
        if [ "$i" -eq "$MAX_RETRIES" ]; then
```

```
            handle_error "Failed to add AWS Managed Rules after $MAX_RETRIES
 attempts"
        fi
        sleep 2
        continue
    elif echo "$update_result" | grep -qi "error"; then
        handle_error "Failed to add AWS Managed Rules: $update_result"
    else
        echo "AWS Managed Rules added successfully."
        break
    fi
done

# Step 4: List CloudFront distributions
echo ""
echo "=================================================="
echo "STEP 4: Listing CloudFront Distributions"
echo "=================================================="

local cf_result
cf_result=$(aws cloudfront list-distributions --query "DistributionList.Items[*].
{Id:Id,DomainName:DomainName}" --output table 2>&1) || cf_result=""

if echo "$cf_result" | grep -qi "error"; then
    echo "Warning: Failed to list CloudFront distributions: $cf_result"
    echo "Continuing without CloudFront association."
    DISTRIBUTION_ID=""
else
    echo "$cf_result"

    # Auto-select first CloudFront distribution if available
    echo ""
    echo "=================================================="
    echo "STEP 5: Associate Web ACL with CloudFront Distribution"
    echo "=================================================="

    local first_dist
    first_dist=$(aws cloudfront list-distributions --query
 "DistributionList.Items[0].Id" --output text 2>&1) || first_dist=""

    if [ -n "$first_dist" ] && [ "$first_dist" != "None" ] && ! echo
 "$first_dist" | grep -qi "error"; then
        DISTRIBUTION_ID="$first_dist"
        echo "Using CloudFront distribution: $DISTRIBUTION_ID"
```

```
        local account_id
        account_id=$(aws sts get-caller-identity --query Account --output text
 2>/dev/null) || account_id=""

        if [ -z "$account_id" ]; then
            echo "Warning: Could not retrieve AWS account ID for association"
            DISTRIBUTION_ID=""
        else
            local associate_result
            associate_result=$(aws wafv2 associate-web-acl \
                --web-acl-arn "$WEB_ACL_ARN" \
                --resource-arn "arn:aws:cloudfront::${account_id}:distribution/
${DISTRIBUTION_ID}" \
                --region us-east-1 2>&1) || true

            if echo "$associate_result" | grep -qi "error"; then
                echo "Warning: Failed to associate Web ACL with CloudFront
 distribution: $associate_result"
                echo "Continuing without CloudFront association."
                DISTRIBUTION_ID=""
            else
                echo "Web ACL associated with CloudFront distribution
 successfully."
            fi
        fi
    else
        echo "No CloudFront distributions available. Skipping association."
        DISTRIBUTION_ID=""
    fi
fi

# Display summary of created resources
echo ""
echo "================================================"
echo "RESOURCE SUMMARY"
echo "================================================"
echo "Web ACL Name: $WEB_ACL_NAME"
echo "Web ACL ID: $WEB_ACL_ID"
echo "Web ACL ARN: $WEB_ACL_ARN"
if [ -n "${DISTRIBUTION_ID:-}" ]; then
    echo "Associated CloudFront Distribution: $DISTRIBUTION_ID"
fi
echo ""
```

```
# Auto-confirm cleanup
echo "============================================="
echo "CLEANUP CONFIRMATION"
echo "============================================="
echo "Proceeding with automatic cleanup of all created resources..."

echo ""
echo "============================================="
echo "Tutorial completed!"
echo "============================================="
echo "Log file: $LOG_FILE"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AssociateWebAcl
  - CreateWebAcl
  - DeleteWebAcl
  - DisassociateWebAcl
  - GetCallerIdentity
  - GetWebAcl
  - ListDistributions
  - UpdateWebAcl

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Spostamento di segreti codificati per proteggere l'archiviazione segreta

L'esempio di codice seguente mostra come:

- Creazione di ruoli IAM
- Creazione di un segreto in Secrets Manager
- Aggiorna il codice dell'applicazione
- Aggiorna il segreto
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# Script to move hardcoded secrets to AWS Secrets Manager
# This script demonstrates how to create IAM roles, store a secret in AWS Secrets
 Manager,
# and set up appropriate permissions

set -euo pipefail

# Set up logging
LOG_FILE="secrets_manager_tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting AWS Secrets Manager tutorial script at $(date)"
echo "============================================================"

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2

    if echo "$output" | grep -qi "error"; then
        echo "ERROR: Command failed: $cmd"
        echo "$output"
        cleanup_resources
        exit 1
    fi
}

# Function to generate a random identifier using secure method
generate_random_id() {
```

```
    python3 -c "import secrets; print('sm' + secrets.token_hex(4))"
}

# Function to safely clean up resources
cleanup_resources() {
    echo ""
    echo "========================================="
    echo "RESOURCES CREATED"
    echo "========================================="

    if [ -n "${SECRET_NAME:-}" ]; then
        echo "Secret: $SECRET_NAME"
    fi

    if [ -n "${RUNTIME_ROLE_NAME:-}" ]; then
        echo "IAM Role: $RUNTIME_ROLE_NAME"
    fi

    if [ -n "${ADMIN_ROLE_NAME:-}" ]; then
        echo "IAM Role: $ADMIN_ROLE_NAME"
    fi

    echo ""
    echo "========================================="
    echo "CLEANUP CONFIRMATION"
    echo "========================================="
    echo "Cleaning up all created resources..."

    # Delete secret if it exists
    if [ -n "${SECRET_NAME:-}" ]; then
        echo "Deleting secret: $SECRET_NAME"
        aws secretsmanager delete-secret --secret-id "$SECRET_NAME" --force-
delete-without-recovery 2>/dev/null || true
    fi

    # Detach policies and delete runtime role if it exists
    if [ -n "${RUNTIME_ROLE_NAME:-}" ]; then
        echo "Deleting inline policies from runtime role: $RUNTIME_ROLE_NAME"
        for policy in $(aws iam list-role-policies --role-name
 "$RUNTIME_ROLE_NAME" --query 'PolicyNames[]' --output text 2>/dev/null || true);
 do
            aws iam delete-role-policy --role-name "$RUNTIME_ROLE_NAME" --policy-
name "$policy" 2>/dev/null || true
        done
```

```
        echo "Deleting IAM role: $RUNTIME_ROLE_NAME"
        aws iam delete-role --role-name "$RUNTIME_ROLE_NAME" 2>/dev/null || true
    fi

    # Detach policies and delete admin role if it exists
    if [ -n "${ADMIN_ROLE_NAME:-}" ]; then
        echo "Detaching policy from role: $ADMIN_ROLE_NAME"
        aws iam detach-role-policy --role-name "$ADMIN_ROLE_NAME" --policy-arn
 "arn:aws:iam::aws:policy/SecretsManagerReadWrite" 2>/dev/null || true

        for policy in $(aws iam list-role-policies --role-name "$ADMIN_ROLE_NAME"
 --query 'PolicyNames[]' --output text 2>/dev/null || true); do
            aws iam delete-role-policy --role-name "$ADMIN_ROLE_NAME" --policy-
name "$policy" 2>/dev/null || true
        done

        echo "Deleting IAM role: $ADMIN_ROLE_NAME"
        aws iam delete-role --role-name "$ADMIN_ROLE_NAME" 2>/dev/null || true
    fi

    echo "Cleanup completed."
}

# Trap to ensure cleanup on script exit
trap 'echo "Script interrupted. Running cleanup..."; cleanup_resources' INT TERM
 EXIT

# Generate random identifiers for resources
ADMIN_ROLE_NAME="SecretsManagerAdmin-$(generate_random_id)"
RUNTIME_ROLE_NAME="RoleToRetrieveSecretAtRuntime-$(generate_random_id)"
SECRET_NAME="MyAPIKey-$(generate_random_id)"

echo "Using the following resource names:"
echo "Admin Role: $ADMIN_ROLE_NAME"
echo "Runtime Role: $RUNTIME_ROLE_NAME"
echo "Secret Name: $SECRET_NAME"
echo ""

# Step 1: Create IAM roles
echo "Creating IAM roles..."

# Create assume role policy document
ASSUME_ROLE_POLICY=$(cat <<'EOF'
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
)

# Create the SecretsManagerAdmin role
echo "Creating admin role: $ADMIN_ROLE_NAME"
ADMIN_ROLE_OUTPUT=$(aws iam create-role \
    --role-name "$ADMIN_ROLE_NAME" \
    --assume-role-policy-document "$ASSUME_ROLE_POLICY" 2>&1)

check_error "$ADMIN_ROLE_OUTPUT" "create-role for admin"
echo "$ADMIN_ROLE_OUTPUT"
aws iam tag-role --role-name "$ADMIN_ROLE_NAME" --tags Key=project,Value=doc-
smith Key=tutorial,Value=aws-secrets-manager-gs

# Attach the SecretsManagerReadWrite policy to the admin role
echo "Attaching SecretsManagerReadWrite policy to admin role"
ATTACH_POLICY_OUTPUT=$(aws iam attach-role-policy \
    --role-name "$ADMIN_ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/SecretsManagerReadWrite" 2>&1)

check_error "$ATTACH_POLICY_OUTPUT" "attach-role-policy for admin"
echo "Policy attached successfully"

# Create the RoleToRetrieveSecretAtRuntime role
echo "Creating runtime role: $RUNTIME_ROLE_NAME"
RUNTIME_ROLE_OUTPUT=$(aws iam create-role \
    --role-name "$RUNTIME_ROLE_NAME" \
    --assume-role-policy-document "$ASSUME_ROLE_POLICY" 2>&1)

check_error "$RUNTIME_ROLE_OUTPUT" "create-role for runtime"
echo "$RUNTIME_ROLE_OUTPUT"
aws iam tag-role --role-name "$RUNTIME_ROLE_NAME" --tags Key=project,Value=doc-
smith Key=tutorial,Value=aws-secrets-manager-gs
```

```
# Wait for roles to be fully created
echo "Waiting for IAM roles to be fully created..."
sleep 10

# Step 2: Create a secret in AWS Secrets Manager
echo "Creating secret in AWS Secrets Manager..."

# Generate secure secret value using environment variable or secure method
# WARNING: In production, use secure methods to inject secrets (AWS CodeBuild,
 parameter store, etc.)
if [ -z "${TUTORIAL_SECRET_VALUE:-}" ]; then
    SECRET_VALUE=$(python3 -c "import json;
 print(json.dumps({'ClientID':'my_client_id','ClientSecret':__import__('secrets').token_u
else
    SECRET_VALUE="$TUTORIAL_SECRET_VALUE"
fi

CREATE_SECRET_OUTPUT=$(aws secretsmanager create-secret \
    --name "$SECRET_NAME" \
    --description "API key for my application" \
    --secret-string "$SECRET_VALUE" \
    --add-replica-regions 'Region=us-east-1' \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=aws-secrets-manager-gs
 2>&1)

check_error "$CREATE_SECRET_OUTPUT" "create-secret"
echo "$CREATE_SECRET_OUTPUT"

# Get AWS account ID
echo "Getting AWS account ID..."
ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text 2>&1)
check_error "$ACCOUNT_ID" "get-caller-identity"
echo "Account ID: $ACCOUNT_ID"

# Get secret ARN for precise resource policy
echo "Getting secret ARN..."
SECRET_ARN=$(aws secretsmanager describe-secret \
    --secret-id "$SECRET_NAME" \
    --query 'ARN' \
    --output text 2>&1)
check_error "$SECRET_ARN" "describe-secret"

# Add resource policy to the secret with least privilege
```

```
echo "Adding resource policy to secret..."
RESOURCE_POLICY=$(cat <<EOF
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRuntimeRoleReadSecret",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::$ACCOUNT_ID:role/$RUNTIME_ROLE_NAME"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "$SECRET_ARN",
            "Condition": {
                "StringEquals": {
                    "secretsmanager:VersionStage": "AWSCURRENT"
                }
            }
        }
    ]
}
EOF
)

PUT_POLICY_OUTPUT=$(aws secretsmanager put-resource-policy \
    --secret-id "$SECRET_NAME" \
    --resource-policy "$RESOURCE_POLICY" \
    --block-public-policy 2>&1)

check_error "$PUT_POLICY_OUTPUT" "put-resource-policy"
echo "Resource policy added successfully"

# Enable rotation policy recommendation
echo "Enabling secret metadata tags for rotation tracking..."
aws secretsmanager tag-resource \
    --secret-id "$SECRET_NAME" \
    --tags Key=Purpose,Value=Tutorial Key=AutoRotation,Value=Recommended 2>/dev/
null || true

# Step 3: Demonstrate retrieving the secret
echo "Retrieving the secret value (for demonstration purposes)..."
GET_SECRET_OUTPUT=$(aws secretsmanager get-secret-value \
    --secret-id "$SECRET_NAME" 2>&1)
```

```
check_error "$GET_SECRET_OUTPUT" "get-secret-value"
echo "Secret retrieved successfully. Secret metadata:"
echo "$GET_SECRET_OUTPUT" | jq '{ARN: .ARN, Name: .Name,
 LastUpdatedDate: .LastUpdatedDate, VersionIdsToStages: .VersionIdsToStages}' 2>/
dev/null || echo "Secret metadata retrieved (jq not available)"

# Step 4: Update the secret with new values
echo "Updating the secret with new values..."
UPDATE_SECRET_VALUE=$(python3 -c "import json;
 print(json.dumps({'ClientID':'my_new_client_id','ClientSecret':__import__('secrets').tok

UPDATE_SECRET_OUTPUT=$(aws secretsmanager update-secret \
    --secret-id "$SECRET_NAME" \
    --secret-string "$UPDATE_SECRET_VALUE" 2>&1)

check_error "$UPDATE_SECRET_OUTPUT" "update-secret"
echo "Secret updated successfully"

# Step 5: Verify the updated secret
echo "Verifying the updated secret..."
VERIFY_SECRET_OUTPUT=$(aws secretsmanager get-secret-value \
    --secret-id "$SECRET_NAME" 2>&1)

check_error "$VERIFY_SECRET_OUTPUT" "get-secret-value for verification"
echo "Updated secret retrieved successfully. Secret metadata:"
echo "$VERIFY_SECRET_OUTPUT" | jq '{ARN: .ARN, Name: .Name,
 LastUpdatedDate: .LastUpdatedDate, VersionIdsToStages: .VersionIdsToStages}' 2>/
dev/null || echo "Secret metadata retrieved (jq not available)"

# Step 6: Display rotation recommendations
echo ""
echo "Rotation Configuration Recommendations:"
echo "========================================="
DESCRIBE_OUTPUT=$(aws secretsmanager describe-secret --secret-id "$SECRET_NAME"
 2>&1)
if echo "$DESCRIBE_OUTPUT" | grep -q "RotationRules"; then
    echo "Current rotation configuration:"
    echo "$DESCRIBE_OUTPUT" | jq '.RotationRules' 2>/dev/null || echo "Rotation
 rules available"
else
    echo "No automatic rotation configured. Consider enabling rotation with:"
    echo "aws secretsmanager rotate-secret --secret-id $SECRET_NAME --rotation-
lambda-arn arn:aws:lambda:REGION:ACCOUNT:function:FUNCTION_NAME --rotation-rules
 AutomaticallyAfterDays=30"
```

```
fi

echo ""
echo "============================================================"
echo "Tutorial completed successfully!"
echo ""
echo "Summary of what we did:"
echo "1. Created IAM roles for managing and retrieving secrets"
echo "2. Created a secret in AWS Secrets Manager with secure generation"
echo "3. Added a least-privilege resource policy to control access to the secret"
echo "4. Retrieved the secret value (simulating application access)"
echo "5. Updated the secret with cryptographically secure values"
echo "6. Verified the updated secret"
echo ""
echo "Security best practices applied:"
echo "- Used cryptographically secure random ID generation"
echo "- Applied least-privilege resource policies with version stages"
echo "- Tagged resources for rotation tracking"
echo "- Blocked public access to secrets"
echo "- Used ARN-specific permissions instead of wildcards"
echo ""
echo "Next steps you might want to consider:"
echo "- Enable automatic secret rotation with AWS Lambda"
echo "- Implement secret caching in your application"
echo "- Set up CloudTrail logging for secret access auditing"
echo "- Use AWS CodeGuru Reviewer to find hardcoded secrets in your code"
echo "- For multi-region applications, replicate your secrets across regions"
echo "- Configure VPC endpoints for private access to Secrets Manager"
echo ""

echo "Script completed at $(date)"
exit 0
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateRole
  - CreateSecret
  - DeleteRole
  - DeleteSecret
  - DetachRolePolicy

- GetCallerIdentity
- GetSecretValue
- PutResourcePolicy
- UpdateSecret

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esegui test di stress della CPU su istanze di macchine virtuali utilizzando l'iniezione di errori

L'esempio di codice seguente mostra come:

- Creazione di ruoli IAM
- Crea un allarme CloudWatch
- Crea un modello di esperimento
- Esegui l'esperimento
- Verifica i risultati
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS FIS CPU Stress Test Tutorial Script
# This script automates the steps in the AWS FIS CPU stress test tutorial
```

```
#    approach using epoch time calculations that work across all Linux
 distributions

# Set up logging
LOG_FILE="fis-tutorial-$(date +%Y%m%d-%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting AWS FIS CPU Stress Test Tutorial Script"
echo "Logging to $LOG_FILE"
echo "============================================="

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2

    if echo "$output" | grep -i "error" > /dev/null; then
        # Ignore specific expected errors
        if [[ "$cmd" == *"aws fis get-experiment"* ]] && [[ "$output" ==
 *"ConfigurationFailure"* ]]; then
            echo "Note: Experiment failed due to configuration issue. This is
 expected in some cases."
            return 0
        fi

        echo "ERROR: Command failed: $cmd"
        echo "Output: $output"
        cleanup_on_error
        exit 1
    fi
}

# Function to clean up resources on error
cleanup_on_error() {
    echo "Error encountered. Cleaning up resources..."

    if [ -n "$EXPERIMENT_ID" ]; then
        echo "Stopping experiment $EXPERIMENT_ID if running..."
        aws fis stop-experiment --id "$EXPERIMENT_ID" 2>/dev/null || true
    fi

    if [ -n "$TEMPLATE_ID" ]; then
        echo "Deleting experiment template $TEMPLATE_ID..."
```

```
        aws fis delete-experiment-template --id "$TEMPLATE_ID" || true
    fi

    if [ -n "$INSTANCE_ID" ]; then
        echo "Terminating EC2 instance $INSTANCE_ID..."
        aws ec2 terminate-instances --instance-ids "$INSTANCE_ID" || true
    fi

    if [ -n "$ALARM_NAME" ]; then
        echo "Deleting CloudWatch alarm $ALARM_NAME..."
        aws cloudwatch delete-alarms --alarm-names "$ALARM_NAME" || true
    fi

    if [ -n "$INSTANCE_PROFILE_NAME" ]; then
        echo "Removing role from instance profile..."
        aws iam remove-role-from-instance-profile --instance-profile-name
 "$INSTANCE_PROFILE_NAME" --role-name "$EC2_ROLE_NAME" || true

        echo "Deleting instance profile..."
        aws iam delete-instance-profile --instance-profile-name
 "$INSTANCE_PROFILE_NAME" || true
    fi

    if [ -n "$FIS_ROLE_NAME" ]; then
        echo "Deleting FIS role policy..."
        aws iam delete-role-policy --role-name "$FIS_ROLE_NAME" --policy-name
 "$FIS_POLICY_NAME" || true

        echo "Deleting FIS role..."
        aws iam delete-role --role-name "$FIS_ROLE_NAME" || true
    fi

    if [ -n "$EC2_ROLE_NAME" ]; then
        echo "Detaching policy from EC2 role..."
        aws iam detach-role-policy --role-name "$EC2_ROLE_NAME" --policy-arn
 "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" || true

        echo "Deleting EC2 role..."
        aws iam delete-role --role-name "$EC2_ROLE_NAME" || true
    fi

    echo "Cleanup completed."
}
```

```
# Generate unique identifiers for resources
TIMESTAMP=$(date +%Y%m%d%H%M%S)
FIS_ROLE_NAME="FISRole-${TIMESTAMP}"
FIS_POLICY_NAME="FISPolicy-${TIMESTAMP}"
EC2_ROLE_NAME="EC2SSMRole-${TIMESTAMP}"
INSTANCE_PROFILE_NAME="EC2SSMProfile-${TIMESTAMP}"
ALARM_NAME="FIS-CPU-Alarm-${TIMESTAMP}"

# Track created resources
CREATED_RESOURCES=()

echo "Step 1: Creating IAM role for AWS FIS"
# Create trust policy file for AWS FIS
cat > fis-trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "fis.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create IAM role for FIS
echo "Creating IAM role $FIS_ROLE_NAME for AWS FIS..."
FIS_ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$FIS_ROLE_NAME" \
  --assume-role-policy-document file://fis-trust-policy.json)
check_error "$FIS_ROLE_OUTPUT" "aws iam create-role"
aws iam tag-role --role-name "$FIS_ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-fault-injection-service-gs
CREATED_RESOURCES+=("IAM Role: $FIS_ROLE_NAME")

# Create policy document for SSM actions
cat > fis-ssm-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "ssm:SendCommand",
        "ssm:ListCommands",
        "ssm:ListCommandInvocations"
      ],
      "Resource": "*"
    }
  ]
}
EOF

# Attach policy to the role
echo "Attaching policy $FIS_POLICY_NAME to role $FIS_ROLE_NAME..."
FIS_POLICY_OUTPUT=$(aws iam put-role-policy \
  --role-name "$FIS_ROLE_NAME" \
  --policy-name "$FIS_POLICY_NAME" \
  --policy-document file://fis-ssm-policy.json)
check_error "$FIS_POLICY_OUTPUT" "aws iam put-role-policy"
CREATED_RESOURCES+=("IAM Policy: $FIS_POLICY_NAME attached to $FIS_ROLE_NAME")

echo "Step 2: Creating IAM role for EC2 instance with SSM permissions"
# Create trust policy file for EC2
cat > ec2-trust-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create IAM role for EC2
echo "Creating IAM role $EC2_ROLE_NAME for EC2 instance..."
EC2_ROLE_OUTPUT=$(aws iam create-role \
  --role-name "$EC2_ROLE_NAME" \
  --assume-role-policy-document file://ec2-trust-policy.json)
check_error "$EC2_ROLE_OUTPUT" "aws iam create-role"
```

```
aws iam tag-role --role-name "$EC2_ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=aws-fault-injection-service-gs
CREATED_RESOURCES+=("IAM Role: $EC2_ROLE_NAME")

# Attach SSM policy to the EC2 role
echo "Attaching AmazonSSMManagedInstanceCore policy to role $EC2_ROLE_NAME..."
EC2_POLICY_OUTPUT=$(aws iam attach-role-policy \
  --role-name "$EC2_ROLE_NAME" \
  --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore)
check_error "$EC2_POLICY_OUTPUT" "aws iam attach-role-policy"
CREATED_RESOURCES+=("IAM Policy: AmazonSSMManagedInstanceCore attached to
 $EC2_ROLE_NAME")

# Create instance profile
echo "Creating instance profile $INSTANCE_PROFILE_NAME..."
PROFILE_OUTPUT=$(aws iam create-instance-profile \
  --instance-profile-name "$INSTANCE_PROFILE_NAME")
check_error "$PROFILE_OUTPUT" "aws iam create-instance-profile"
CREATED_RESOURCES+=("IAM Instance Profile: $INSTANCE_PROFILE_NAME")

# Add role to instance profile
echo "Adding role $EC2_ROLE_NAME to instance profile $INSTANCE_PROFILE_NAME..."
ADD_ROLE_OUTPUT=$(aws iam add-role-to-instance-profile \
  --instance-profile-name "$INSTANCE_PROFILE_NAME" \
  --role-name "$EC2_ROLE_NAME")
check_error "$ADD_ROLE_OUTPUT" "aws iam add-role-to-instance-profile"

# Wait for role to propagate
echo "Waiting for IAM role to propagate..."
sleep 10

echo "Step 3: Launching EC2 instance"
# Get the latest Amazon Linux 2 AMI ID
echo "Finding latest Amazon Linux 2 AMI..."
AMI_ID=$(aws ec2 describe-images \
  --owners amazon \
  --filters "Name=name,Values=amzn2-ami-hvm-*-x86_64-gp2"
 "Name=state,Values=available" \
  --query "sort_by(Images, &CreationDate)[-1].ImageId" \
  --output text)
check_error "$AMI_ID" "aws ec2 describe-images"
echo "Using AMI: $AMI_ID"

# Launch EC2 instance
```

```
echo "Launching EC2 instance with AMI $AMI_ID..."
INSTANCE_OUTPUT=$(aws ec2 run-instances \
  --image-id "$AMI_ID" \
  --instance-type t2.micro \
  --iam-instance-profile Name="$INSTANCE_PROFILE_NAME" \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=FIS-Test-
Instance},{Key=project,Value=doc-smith},{Key=tutorial,Value=aws-fault-injection-
service-gs}]')
check_error "$INSTANCE_OUTPUT" "aws ec2 run-instances"

# Get instance ID
INSTANCE_ID=$(echo "$INSTANCE_OUTPUT" | grep -i "InstanceId" | head -1 | awk -
F'"' '{print $4}')
if [ -z "$INSTANCE_ID" ]; then
    echo "Failed to get instance ID"
    cleanup_on_error
    exit 1
fi
echo "Launched instance: $INSTANCE_ID"
CREATED_RESOURCES+=("EC2 Instance: $INSTANCE_ID")

# Enable detailed monitoring
echo "Enabling detailed monitoring for instance $INSTANCE_ID..."
MONITOR_OUTPUT=$(aws ec2 monitor-instances --instance-ids "$INSTANCE_ID")
check_error "$MONITOR_OUTPUT" "aws ec2 monitor-instances"

# Wait for instance to be running and status checks to pass
echo "Waiting for instance to be ready..."
aws ec2 wait instance-running --instance-ids "$INSTANCE_ID"
aws ec2 wait instance-status-ok --instance-ids "$INSTANCE_ID"
echo "Instance is ready"

echo "Step 4: Creating CloudWatch alarm for CPU utilization"
# Create CloudWatch alarm
echo "Creating CloudWatch alarm $ALARM_NAME..."
ALARM_OUTPUT=$(aws cloudwatch put-metric-alarm \
  --alarm-name "$ALARM_NAME" \
  --alarm-description "Alarm when CPU exceeds 50%" \
  --metric-name CPUUtilization \
  --namespace AWS/EC2 \
  --statistic Maximum \
  --period 60 \
  --threshold 50 \
  --comparison-operator GreaterThanOrEqualToThreshold \
```

```
  --dimensions "Name=InstanceId,Value=$INSTANCE_ID" \
  --evaluation-periods 1)
check_error "$ALARM_OUTPUT" "aws cloudwatch put-metric-alarm"
CREATED_RESOURCES+=("CloudWatch Alarm: $ALARM_NAME")

# Get the alarm ARN
echo "Getting CloudWatch alarm ARN..."
ALARM_ARN_OUTPUT=$(aws cloudwatch describe-alarms \
  --alarm-names "$ALARM_NAME")
check_error "$ALARM_ARN_OUTPUT" "aws cloudwatch describe-alarms"
ALARM_ARN=$(echo "$ALARM_ARN_OUTPUT" | grep -i "AlarmArn" | head -1 | awk -F'"'
 '{print $4}')
if [ -z "$ALARM_ARN" ]; then
    echo "Failed to get alarm ARN"
    cleanup_on_error
    exit 1
fi
echo "Alarm ARN: $ALARM_ARN"

# Wait for the alarm to initialize and reach OK state
echo "Waiting for CloudWatch alarm to initialize (60 seconds)..."
sleep 60

# Check alarm state
echo "Checking alarm state..."
ALARM_STATE_OUTPUT=$(aws cloudwatch describe-alarms \
  --alarm-names "$ALARM_NAME")
ALARM_STATE=$(echo "$ALARM_STATE_OUTPUT" | grep -i "StateValue" | head -1 | awk -
F'"' '{print $4}')
echo "Current alarm state: $ALARM_STATE"

# If alarm is not in OK state, wait longer or generate some baseline metrics
if [ "$ALARM_STATE" != "OK" ]; then
    echo "Alarm not in OK state. Waiting for alarm to stabilize (additional 60
 seconds)..."
    sleep 60

    # Check alarm state again
    ALARM_STATE_OUTPUT=$(aws cloudwatch describe-alarms \
      --alarm-names "$ALARM_NAME")
    ALARM_STATE=$(echo "$ALARM_STATE_OUTPUT" | grep -i "StateValue" | head -1 |
 awk -F'"' '{print $4}')
    echo "Updated alarm state: $ALARM_STATE"
```

```
    if [ "$ALARM_STATE" != "OK" ]; then
        echo "Warning: Alarm still not in OK state. Experiment may fail to
 start."
    fi
fi

echo "Step 5: Creating AWS FIS experiment template"
# Get the IAM role ARN
echo "Getting IAM role ARN for $FIS_ROLE_NAME..."
ROLE_ARN_OUTPUT=$(aws iam get-role \
  --role-name "$FIS_ROLE_NAME")
check_error "$ROLE_ARN_OUTPUT" "aws iam get-role"
ROLE_ARN=$(echo "$ROLE_ARN_OUTPUT" | grep -i "Arn" | head -1 | awk -F'"' '{print
 $4}')
if [ -z "$ROLE_ARN" ]; then
    echo "Failed to get role ARN"
    cleanup_on_error
    exit 1
fi
echo "Role ARN: $ROLE_ARN"

# Get account ID and region
ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
REGION=$(aws configure get region)
if [ -z "$REGION" ]; then
    REGION="us-east-1"  # Default to us-east-1 if region not set
fi
INSTANCE_ARN="arn:aws:ec2:${REGION}:${ACCOUNT_ID}:instance/${INSTANCE_ID}"
echo "Instance ARN: $INSTANCE_ARN"

# Create experiment template - Fixed JSON escaping issue
cat > experiment-template.json << EOF
{
  "description": "Test CPU stress predefined SSM document",
  "targets": {
    "testInstance": {
      "resourceType": "aws:ec2:instance",
      "resourceArns": ["$INSTANCE_ARN"],
      "selectionMode": "ALL"
    }
  },
  "actions": {
    "runCpuStress": {
      "actionId": "aws:ssm:send-command",
```

```
      "parameters": {
        "documentArn": "arn:aws:ssm:$REGION::document/AWSFIS-Run-CPU-Stress",
        "documentParameters": "{\"DurationSeconds\":\"120\"}",
        "duration": "PT5M"
      },
      "targets": {
        "Instances": "testInstance"
      }
    }
  },
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": "$ALARM_ARN"
    }
  ],
  "roleArn": "$ROLE_ARN",
  "tags": {
    "Name": "FIS-CPU-Stress-Experiment",
    "project": "doc-smith",
    "tutorial": "aws-fault-injection-service-gs"
  }
}
EOF

# Create experiment template
echo "Creating AWS FIS experiment template..."
TEMPLATE_OUTPUT=$(aws fis create-experiment-template --cli-input-json file://
experiment-template.json)
check_error "$TEMPLATE_OUTPUT" "aws fis create-experiment-template"
TEMPLATE_ID=$(echo "$TEMPLATE_OUTPUT" | grep -i "id" | head -1 | awk -F'"'
 '{print $4}')
if [ -z "$TEMPLATE_ID" ]; then
    echo "Failed to get template ID"
    cleanup_on_error
    exit 1
fi
echo "Experiment template created with ID: $TEMPLATE_ID"
CREATED_RESOURCES+=("FIS Experiment Template: $TEMPLATE_ID")

echo "Step 6: Starting the experiment"
# Start the experiment
echo "Starting AWS FIS experiment using template $TEMPLATE_ID..."
EXPERIMENT_OUTPUT=$(aws fis start-experiment \
```

```
  --experiment-template-id "$TEMPLATE_ID" \
  --tags '{"Name": "FIS-CPU-Stress-Run"}')
check_error "$EXPERIMENT_OUTPUT" "aws fis start-experiment"
EXPERIMENT_ID=$(echo "$EXPERIMENT_OUTPUT" | grep -i "id" | head -1 | awk -F'"'
 '{print $4}')
if [ -z "$EXPERIMENT_ID" ]; then
    echo "Failed to get experiment ID"
    cleanup_on_error
    exit 1
fi
echo "Experiment started with ID: $EXPERIMENT_ID"
CREATED_RESOURCES+=("FIS Experiment: $EXPERIMENT_ID")

echo "Step 7: Tracking experiment progress"
# Track experiment progress
echo "Tracking experiment progress..."
MAX_CHECKS=30
CHECK_COUNT=0
EXPERIMENT_STATE=""

while [ $CHECK_COUNT -lt $MAX_CHECKS ]; do
    EXPERIMENT_INFO=$(aws fis get-experiment --id "$EXPERIMENT_ID")
    # Don't check for errors here, as we expect some experiments to fail

    EXPERIMENT_STATE=$(echo "$EXPERIMENT_INFO" | grep -i "status" | head -1 | awk
 -F'"' '{print $4}')
    echo "Experiment state: $EXPERIMENT_STATE"

    if [ "$EXPERIMENT_STATE" == "completed" ] || [ "$EXPERIMENT_STATE" ==
 "stopped" ] || [ "$EXPERIMENT_STATE" == "failed" ]; then
        # Show the reason for the state
        REASON=$(echo "$EXPERIMENT_INFO" | grep -i "reason" | head -1 | awk -F'"'
 '{print $4}')
        if [ -n "$REASON" ]; then
            echo "Reason: $REASON"
        fi
        break
    fi

    echo "Waiting 10 seconds before checking again..."
    sleep 10
    CHECK_COUNT=$((CHECK_COUNT + 1))
done
```

```
if [ $CHECK_COUNT -eq $MAX_CHECKS ]; then
    echo "Experiment is taking longer than expected. You can check its status
 later using:"
    echo "aws fis get-experiment --id $EXPERIMENT_ID"
fi

echo "Step 8: Verifying experiment results"
# Check CloudWatch alarm state
echo "Checking CloudWatch alarm state..."
ALARM_STATE_OUTPUT=$(aws cloudwatch describe-alarms --alarm-names "$ALARM_NAME")
check_error "$ALARM_STATE_OUTPUT" "aws cloudwatch describe-alarms"
echo "$ALARM_STATE_OUTPUT"

# Get CPU utilization metrics
echo "Getting CPU utilization metrics..."
END_TIME=$(date -u +"%Y-%m-%dT%H:%M:%SZ")

# FIXED: Cross-platform compatible way to calculate time 10 minutes ago
# This approach uses epoch seconds and basic arithmetic which works on all Linux
 distributions
CURRENT_EPOCH=$(date +%s)
TEN_MINUTES_AGO_EPOCH=$((CURRENT_EPOCH - 600))
START_TIME=$(date -u -d "@$TEN_MINUTES_AGO_EPOCH" +"%Y-%m-%dT%H:%M:%SZ" 2>/dev/
null || date -u -r "$TEN_MINUTES_AGO_EPOCH" +"%Y-%m-%dT%H:%M:%SZ")

# Create metric query file
cat > metric-query.json << EOF
[
  {
    "Id": "cpu",
    "MetricStat": {
      "Metric": {
        "Namespace": "AWS/EC2",
        "MetricName": "CPUUtilization",
        "Dimensions": [
          {
            "Name": "InstanceId",
            "Value": "$INSTANCE_ID"
          }
        ]
      },
      "Period": 60,
      "Stat": "Maximum"
    }
```

```
  }
]
EOF

METRICS_OUTPUT=$(aws cloudwatch get-metric-data \
  --start-time "$START_TIME" \
  --end-time "$END_TIME" \
  --metric-data-queries file://metric-query.json)
check_error "$METRICS_OUTPUT" "aws cloudwatch get-metric-data"
echo "CPU Utilization Metrics:"
echo "$METRICS_OUTPUT"

# Display summary of created resources
echo ""
echo "==========================================="
echo "RESOURCES CREATED"
echo "==========================================="
for resource in "${CREATED_RESOURCES[@]}"; do
    echo "- $resource"
done
echo "==========================================="

# Prompt for cleanup
echo ""
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    echo "Starting cleanup process..."

    # Stop experiment if still running
    if [ "$EXPERIMENT_STATE" != "completed" ] && [ "$EXPERIMENT_STATE" !=
 "stopped" ] && [ "$EXPERIMENT_STATE" != "failed" ]; then
        echo "Stopping experiment $EXPERIMENT_ID..."
        STOP_OUTPUT=$(aws fis stop-experiment --id "$EXPERIMENT_ID")
        check_error "$STOP_OUTPUT" "aws fis stop-experiment"
        echo "Waiting for experiment to stop..."
        sleep 10
    fi

    # Delete experiment template
```

```
    echo "Deleting experiment template $TEMPLATE_ID..."
    DELETE_TEMPLATE_OUTPUT=$(aws fis delete-experiment-template --id
"$TEMPLATE_ID")
    check_error "$DELETE_TEMPLATE_OUTPUT" "aws fis delete-experiment-template"

    # Delete CloudWatch alarm
    echo "Deleting CloudWatch alarm $ALARM_NAME..."
    DELETE_ALARM_OUTPUT=$(aws cloudwatch delete-alarms --alarm-names
"$ALARM_NAME")
    check_error "$DELETE_ALARM_OUTPUT" "aws cloudwatch delete-alarms"

    # Terminate EC2 instance
    echo "Terminating EC2 instance $INSTANCE_ID..."
    TERMINATE_OUTPUT=$(aws ec2 terminate-instances --instance-ids "$INSTANCE_ID")
    check_error "$TERMINATE_OUTPUT" "aws ec2 terminate-instances"
    echo "Waiting for instance to terminate..."
    aws ec2 wait instance-terminated --instance-ids "$INSTANCE_ID"

    # Clean up IAM resources
    echo "Removing role from instance profile..."
    REMOVE_ROLE_OUTPUT=$(aws iam remove-role-from-instance-profile \
      --instance-profile-name "$INSTANCE_PROFILE_NAME" \
      --role-name "$EC2_ROLE_NAME")
    check_error "$REMOVE_ROLE_OUTPUT" "aws iam remove-role-from-instance-profile"

    echo "Deleting instance profile..."
    DELETE_PROFILE_OUTPUT=$(aws iam delete-instance-profile \
      --instance-profile-name "$INSTANCE_PROFILE_NAME")
    check_error "$DELETE_PROFILE_OUTPUT" "aws iam delete-instance-profile"

    echo "Deleting FIS role policy..."
    DELETE_POLICY_OUTPUT=$(aws iam delete-role-policy \
      --role-name "$FIS_ROLE_NAME" \
      --policy-name "$FIS_POLICY_NAME")
    check_error "$DELETE_POLICY_OUTPUT" "aws iam delete-role-policy"

    echo "Detaching policy from EC2 role..."
    DETACH_POLICY_OUTPUT=$(aws iam detach-role-policy \
      --role-name "$EC2_ROLE_NAME" \
      --policy-arn "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore")
    check_error "$DETACH_POLICY_OUTPUT" "aws iam detach-role-policy"

    echo "Deleting FIS role..."
    DELETE_FIS_ROLE_OUTPUT=$(aws iam delete-role \
```

```
      --role-name "$FIS_ROLE_NAME")
    check_error "$DELETE_FIS_ROLE_OUTPUT" "aws iam delete-role"

    echo "Deleting EC2 role..."
    DELETE_EC2_ROLE_OUTPUT=$(aws iam delete-role \
      --role-name "$EC2_ROLE_NAME")
    check_error "$DELETE_EC2_ROLE_OUTPUT" "aws iam delete-role"

    # Clean up temporary files
    echo "Cleaning up temporary files..."
    rm -f fis-trust-policy.json ec2-trust-policy.json fis-ssm-policy.json
 experiment-template.json metric-query.json

    echo "Cleanup completed successfully."
else
    echo "Cleanup skipped. Resources will remain in your AWS account."
    echo "You can manually clean up the resources listed above."
fi

echo ""
echo "Script execution completed."
echo "Log file: $LOG_FILE"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AddRoleToInstanceProfile
  - AttachRolePolicy
  - CreateExperimentTemplate
  - CreateInstanceProfile
  - CreateRole
  - DeleteAlarms
  - DeleteExperimentTemplate
  - DeleteInstanceProfile
  - DeleteRole
  - DeleteRolePolicy
  - DescribeAlarms
  - DescribeImages

- DetachRolePolicy
- GetCallerIdentity
- GetExperiment
- GetMetricData
- GetRole
- MonitorInstances
- PutMetricAlarm
- PutRolePolicy
- RemoveRoleFromInstanceProfile
- RunInstances
- StartExperiment
- StopExperiment
- TerminateInstances
- Attendi

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Configurazione della gestione dei sistemi

L'esempio di codice seguente mostra come:

- Creare autorizzazioni IAM per Systems Manager
- Creare un ruolo IAM per Systems Manager
- Configurazione di Systems Manager
- Verifica la configurazione
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS Systems Manager Setup Script
# This script sets up AWS Systems Manager for a single account and region
#
# Version 17 fixes:
# 1. Added cloudformation.amazonaws.com to the IAM role trust policy
# 2. Systems Manager Quick Setup uses CloudFormation for deployments, so the role
 must trust CloudFormation service

# Initialize log file
LOG_FILE="ssm_setup_$(date +%Y%m%d_%H%M%S).log"
UNIQUE_ID=$(openssl rand -hex 4)
echo "Starting AWS Systems Manager setup at $(date)" > "$LOG_FILE"

# Function to log commands and their outputs with immediate terminal display
log_cmd() {
    echo "$(date): Running command: $1" | tee -a "$LOG_FILE"
    local output
    output=$(eval "$1" 2>&1)
    local status=$?
    echo "$output" | tee -a "$LOG_FILE"
    return $status
}

# Function to check for errors in command output
check_error() {
    local cmd_output="$1"
    local cmd_status="$2"
    local error_msg="$3"
```

```
    if [[ $cmd_status -ne 0 || "$cmd_output" =~ [Ee][Rr][Rr][Oo][Rr] ]]; then
        echo "ERROR: $error_msg" | tee -a "$LOG_FILE"
        echo "Command output: $cmd_output" | tee -a "$LOG_FILE"
        cleanup_on_error
        exit 1
    fi
}

# Array to track created resources for cleanup
declare -a CREATED_RESOURCES

# Function to add a resource to the tracking array
track_resource() {
    local resource_type="$1"
    local resource_id="$2"
    CREATED_RESOURCES+=("$resource_type:$resource_id")
    echo "Tracked resource: $resource_type:$resource_id" | tee -a "$LOG_FILE"
}

# Function to clean up resources on error
cleanup_on_error() {
    echo "" | tee -a "$LOG_FILE"
    echo "========================================" | tee -a "$LOG_FILE"
    echo "ERROR OCCURRED - CLEANING UP RESOURCES" | tee -a "$LOG_FILE"
    echo "========================================" | tee -a "$LOG_FILE"
    echo "The following resources were created:" | tee -a "$LOG_FILE"

    # Display resources in reverse order
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        echo "${CREATED_RESOURCES[$i]}" | tee -a "$LOG_FILE"
    done

    echo "" | tee -a "$LOG_FILE"
    echo "Attempting to clean up resources..." | tee -a "$LOG_FILE"

    # Clean up resources in reverse order
    cleanup_resources
}

# Function to clean up all created resources
cleanup_resources() {
    # Process resources in reverse order (last created, first deleted)
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        IFS=':' read -r resource_type resource_id <<< "${CREATED_RESOURCES[$i]}"
```

```
        echo "Deleting $resource_type: $resource_id" | tee -a "$LOG_FILE"

        case "$resource_type" in
            "IAM_POLICY")
                # Delete the policy (detachment should have been handled when the
 role was deleted)
                log_cmd "aws iam delete-policy --policy-arn $resource_id" || true
                ;;
            "IAM_ROLE")
                # Detach all policies from the role first
                if [[ -n "$POLICY_ARN" ]]; then
                    log_cmd "aws iam detach-role-policy --role-name $resource_id
 --policy-arn $POLICY_ARN" || true
                fi

                # Delete the role
                log_cmd "aws iam delete-role --role-name $resource_id" || true
                ;;
            "SSM_CONFIG_MANAGER")
                log_cmd "aws ssm-quicksetup delete-configuration-manager --
manager-arn $resource_id" || true
                ;;
            *)
                echo "Unknown resource type: $resource_type, cannot delete
 automatically" | tee -a "$LOG_FILE"
                ;;
        esac
    done

    echo "Cleanup completed" | tee -a "$LOG_FILE"

    # Clean up temporary files
    rm -f ssm-onboarding-policy.json trust-policy.json ssm-config.json 2>/dev/
null || true
}

# Main script execution
echo "AWS Systems Manager Setup Script"
echo "================================"
echo "This script will set up AWS Systems Manager for a single account and
 region."
echo "It will create IAM policies and roles, then enable Systems Manager
 features."
```

```
echo ""

# Get the current AWS region
CURRENT_REGION=$(aws configure get region)
if [[ -z "$CURRENT_REGION" ]]; then
    echo "No AWS region configured. Please specify a region:"
    CURRENT_REGION="${AWS_DEFAULT_REGION:-us-west-2}"
    if [[ -z "$CURRENT_REGION" ]]; then
        echo "ERROR: A region must be specified" | tee -a "$LOG_FILE"
        exit 1
    fi
fi

echo "Using AWS region: $CURRENT_REGION" | tee -a "$LOG_FILE"

# Step 1: Create IAM policy for Systems Manager onboarding
echo "Step 1: Creating IAM policy for Systems Manager onboarding..."

# Create policy document
cat > ssm-onboarding-policy.json << 'EOF'
{
   "Version":"2012-10-17",
   "Statement": [
     {
       "Sid": "QuickSetupActions",
       "Effect": "Allow",
       "Action": [
         "ssm-quicksetup:*"
       ],
       "Resource": "*"
     },
     {
       "Sid": "SsmReadOnly",
       "Effect": "Allow",
       "Action": [
         "ssm:DescribeAutomationExecutions",
         "ssm:GetAutomationExecution",
         "ssm:ListAssociations",
         "ssm:DescribeAssociation",
         "ssm:ListDocuments",
         "ssm:ListResourceDataSync",
         "ssm:DescribePatchBaselines",
         "ssm:GetPatchBaseline",
         "ssm:DescribeMaintenanceWindows",
```

```
            "ssm:DescribeMaintenanceWindowTasks"
          ],
          "Resource": "*"
        },
        {
          "Sid": "SsmDocument",
          "Effect": "Allow",
          "Action": [
            "ssm:GetDocument",
            "ssm:DescribeDocument"
          ],
          "Resource": [
            "arn:aws:ssm:*:*:document/AWSQuickSetupType-*",
            "arn:aws:ssm:*:*:document/AWS-EnableExplorer"
          ]
        },
        {
          "Sid": "SsmEnableExplorer",
          "Effect": "Allow",
          "Action": "ssm:StartAutomationExecution",
          "Resource": "arn:aws:ssm:*:*:automation-definition/AWS-EnableExplorer:*"
        },
        {
          "Sid": "SsmExplorerRds",
          "Effect": "Allow",
          "Action": [
            "ssm:GetOpsSummary",
            "ssm:CreateResourceDataSync",
            "ssm:UpdateResourceDataSync"
          ],
          "Resource": "arn:aws:ssm:*:*:resource-data-sync/AWS-QuickSetup-*"
        },
        {
          "Sid": "OrgsReadOnly",
          "Effect": "Allow",
          "Action": [
            "organizations:DescribeAccount",
            "organizations:DescribeOrganization",
            "organizations:ListDelegatedAdministrators",
            "organizations:ListRoots",
            "organizations:ListParents",
            "organizations:ListOrganizationalUnitsForParent",
            "organizations:DescribeOrganizationalUnit",
            "organizations:ListAWSServiceAccessForOrganization"
```

```
        ],
        "Resource": "*"
      },
      {
        "Sid": "OrgsAdministration",
        "Effect": "Allow",
        "Action": [
          "organizations:EnableAWSServiceAccess",
          "organizations:RegisterDelegatedAdministrator",
          "organizations:DeregisterDelegatedAdministrator"
        ],
        "Resource": "*",
        "Condition": {
          "StringEquals": {
            "organizations:ServicePrincipal": [
              "ssm.amazonaws.com",
              "ssm-quicksetup.amazonaws.com",
              "member.org.stacksets.cloudformation.amazonaws.com",
              "resource-explorer-2.amazonaws.com"
            ]
          }
        }
      },
      {
        "Sid": "CfnReadOnly",
        "Effect": "Allow",
        "Action": [
          "cloudformation:ListStacks",
          "cloudformation:DescribeStacks",
          "cloudformation:ListStackSets",
          "cloudformation:DescribeOrganizationsAccess"
        ],
        "Resource": "*"
      },
      {
        "Sid": "OrgCfnAccess",
        "Effect": "Allow",
        "Action": [
          "cloudformation:ActivateOrganizationsAccess"
        ],
        "Resource": "*"
      },
      {
        "Sid": "CfnStackActions",
```

```
        "Effect": "Allow",
        "Action": [
          "cloudformation:CreateStack",
          "cloudformation:DeleteStack",
          "cloudformation:DescribeStackResources",
          "cloudformation:DescribeStackEvents",
          "cloudformation:GetTemplate",
          "cloudformation:RollbackStack",
          "cloudformation:TagResource",
          "cloudformation:UntagResource",
          "cloudformation:UpdateStack"
        ],
        "Resource": [
          "arn:aws:cloudformation:*:*:stack/StackSet-AWS-QuickSetup-*",
          "arn:aws:cloudformation:*:*:stack/AWS-QuickSetup-*",
          "arn:aws:cloudformation:*:*:type/resource/*"
        ]
      },
      {
        "Sid": "CfnStackSetActions",
        "Effect": "Allow",
        "Action": [
          "cloudformation:CreateStackInstances",
          "cloudformation:CreateStackSet",
          "cloudformation:DeleteStackInstances",
          "cloudformation:DeleteStackSet",
          "cloudformation:DescribeStackInstance",
          "cloudformation:DetectStackSetDrift",
          "cloudformation:ListStackInstanceResourceDrifts",
          "cloudformation:DescribeStackSet",
          "cloudformation:DescribeStackSetOperation",
          "cloudformation:ListStackInstances",
          "cloudformation:ListStackSetOperations",
          "cloudformation:ListStackSetOperationResults",
          "cloudformation:TagResource",
          "cloudformation:UntagResource",
          "cloudformation:UpdateStackSet"
        ],
        "Resource": [
          "arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-*",
          "arn:aws:cloudformation:*:*:type/resource/*",
          "arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-*:*"
        ]
      },
```

```
    {
      "Sid": "ValidationReadonlyActions",
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles",
        "iam:GetRole"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IamRolesMgmt",
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:GetRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:GetRolePolicy",
        "iam:ListRolePolicies"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ]
    },
    {
      "Sid": "IamPassRole",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "ssm.amazonaws.com",
            "ssm-quicksetup.amazonaws.com",
            "cloudformation.amazonaws.com"
          ]
        }
```

```
      }
    },
    {
      "Sid": "IamRolesPoliciesMgmt",
      "Effect": "Allow",
      "Action": [
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ],
      "Condition": {
        "ArnEquals": {
          "iam:PolicyARN": [
            "arn:aws:iam::aws:policy/
AWSSystemsManagerEnableExplorerExecutionPolicy",
            "arn:aws:iam::aws:policy/AWSQuickSetupSSMDeploymentRolePolicy"
          ]
        }
      }
    },
    {
      "Sid": "CfnStackSetsSLR",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/
stacksets.cloudformation.amazonaws.com/
AWSServiceRoleForCloudFormationStackSetsOrgAdmin",
        "arn:aws:iam::*:role/aws-service-role/ssm.amazonaws.com/
AWSServiceRoleForAmazonSSM",
        "arn:aws:iam::*:role/aws-service-role/
accountdiscovery.ssm.amazonaws.com/AWSServiceRoleForAmazonSSM_AccountDiscovery",
        "arn:aws:iam::*:role/aws-service-role/ssm-quicksetup.amazonaws.com/
AWSServiceRoleForSSMQuickSetup",
        "arn:aws:iam::*:role/aws-service-role/resource-explorer-2.amazonaws.com/
AWSServiceRoleForResourceExplorer"
      ]
    }
  ]
```

```
}
EOF

# Create the IAM policy
POLICY_OUTPUT=$(log_cmd "aws iam create-policy --policy-name SSMOnboardingPolicy-
$UNIQUE_ID --policy-document file://ssm-onboarding-policy.json --output json")
POLICY_STATUS=$?
check_error "$POLICY_OUTPUT" $POLICY_STATUS "Failed to create IAM policy"

# Extract the policy ARN
POLICY_ARN=$(echo "$POLICY_OUTPUT" | grep -o 'arn:aws:iam::[0-9]*:policy/
SSMOnboardingPolicy-[a-f0-9]*')
if [[ -z "$POLICY_ARN" ]]; then
    echo "ERROR: Failed to extract policy ARN" | tee -a "$LOG_FILE"
    exit 1
fi

# Track the created policy
track_resource "IAM_POLICY" "$POLICY_ARN"
aws iam tag-policy --policy-arn "$POLICY_ARN" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=aws-systems-manager-gs

echo "Created policy: $POLICY_ARN" | tee -a "$LOG_FILE"

# Step 2: Create and configure IAM role for Systems Manager
echo ""
echo "Step 2: Creating IAM role for Systems Manager..."

# Get current user name
USER_OUTPUT=$(log_cmd "aws sts get-caller-identity --output json")
USER_STATUS=$?
check_error "$USER_OUTPUT" $USER_STATUS "Failed to get caller identity"

# Extract account ID
ACCOUNT_ID=$(echo "$USER_OUTPUT" | grep -o '"Account": "[0-9]*"' | cut -d'"' -f4)
if [[ -z "$ACCOUNT_ID" ]]; then
    echo "ERROR: Failed to extract account ID" | tee -a "$LOG_FILE"
    exit 1
fi

# Generate a unique role name
ROLE_NAME="SSMTutorialRole-$(openssl rand -hex 4)"

# Create trust policy for the role - FIXED: Added cloudformation.amazonaws.com
```

```
cat > trust-policy.json << 'EOF'
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ssm.amazonaws.com",
                    "ssm-quicksetup.amazonaws.com",
                    "cloudformation.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNT_ID:root"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF

# Replace ACCOUNT_ID placeholder in trust policy
sed -i "s/ACCOUNT_ID/$ACCOUNT_ID/g" trust-policy.json

# Create the IAM role
ROLE_OUTPUT=$(log_cmd "aws iam create-role --role-name $ROLE_NAME --assume-role-
policy-document file://trust-policy.json --description 'Role for Systems Manager
 tutorial' --output json")
ROLE_STATUS=$?
check_error "$ROLE_OUTPUT" $ROLE_STATUS "Failed to create IAM role"

# Extract the role ARN
ROLE_ARN=$(echo "$ROLE_OUTPUT" | grep -o 'arn:aws:iam::[0-9]*:role/[^"]*')
if [[ -z "$ROLE_ARN" ]]; then
    echo "ERROR: Failed to extract role ARN" | tee -a "$LOG_FILE"
    cleanup_on_error
    exit 1
fi
```

```
# Track the created role
track_resource "IAM_ROLE" "$ROLE_NAME"
aws iam tag-role --role-name "$ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=aws-systems-manager-gs

echo "Created IAM role: $ROLE_NAME" | tee -a "$LOG_FILE"
echo "Role ARN: $ROLE_ARN" | tee -a "$LOG_FILE"

# Set identity variables for cleanup
IDENTITY_TYPE="role"
IDENTITY_NAME="$ROLE_NAME"

# Attach the policy to the role
ATTACH_OUTPUT=$(log_cmd "aws iam attach-role-policy --role-name $ROLE_NAME --
policy-arn $POLICY_ARN")
ATTACH_STATUS=$?
check_error "$ATTACH_OUTPUT" $ATTACH_STATUS "Failed to attach policy to role
 $ROLE_NAME"

echo "Policy attached to role: $ROLE_NAME" | tee -a "$LOG_FILE"

# Step 3: Create Systems Manager configuration using Host Management
echo ""
echo "Step 3: Creating Systems Manager configuration..."

# Generate a random identifier for the configuration name
CONFIG_NAME="SSMSetup-$(openssl rand -hex 4)"

# Create configuration file for Systems Manager setup using Host Management
# Added both required parameters for single account deployment based on
 CloudFormation documentation
cat > ssm-config.json << EOF
[
  {
    "Type": "AWSQuickSetupType-SSMHostMgmt",
    "LocalDeploymentAdministrationRoleArn": "$ROLE_ARN",
    "LocalDeploymentExecutionRoleName": "$ROLE_NAME",
    "Parameters": {
      "TargetAccounts": "$ACCOUNT_ID",
      "TargetRegions": "$CURRENT_REGION"
    }
  }
]
EOF
```

```
echo "Configuration file created:" | tee -a "$LOG_FILE"
cat ssm-config.json | tee -a "$LOG_FILE"

# Create the configuration manager
CONFIG_OUTPUT=$(log_cmd "aws ssm-quicksetup create-configuration-manager --name
 \"$CONFIG_NAME\" --configuration-definitions file://ssm-config.json --region
 $CURRENT_REGION")
CONFIG_STATUS=$?
check_error "$CONFIG_OUTPUT" $CONFIG_STATUS "Failed to create Systems Manager
 configuration"

# Extract the manager ARN
MANAGER_ARN=$(echo "$CONFIG_OUTPUT" | grep -o 'arn:aws:ssm-quicksetup:[^"]*')
if [[ -z "$MANAGER_ARN" ]]; then
    echo "ERROR: Failed to extract manager ARN" | tee -a "$LOG_FILE"
    exit 1
fi

# Track the created configuration manager
track_resource "SSM_CONFIG_MANAGER" "$MANAGER_ARN"

echo "Created Systems Manager configuration: $MANAGER_ARN" | tee -a "$LOG_FILE"

# Step 4: Verify the setup
echo ""
echo "Step 4: Verifying the setup..."

# Wait for the configuration to be fully deployed
echo "Waiting for the configuration to be deployed (this may take a few
 minutes)..."
sleep 30

# Check the configuration manager status
VERIFY_OUTPUT=$(log_cmd "aws ssm-quicksetup get-configuration-manager --manager-
arn $MANAGER_ARN --region $CURRENT_REGION")
VERIFY_STATUS=$?
check_error "$VERIFY_OUTPUT" $VERIFY_STATUS "Failed to verify configuration
 manager"

echo "Systems Manager setup completed successfully!" | tee -a "$LOG_FILE"

# List the created resources
echo ""
```

```
echo "============================================"
echo "CREATED RESOURCES"
echo "============================================"
for resource in "${CREATED_RESOURCES[@]+"${CREATED_RESOURCES[@]}"}"; do
    echo "$resource"
done

# Prompt for cleanup
echo ""
echo "============================================"
echo "CLEANUP CONFIRMATION"
echo "============================================"
echo "Do you want to clean up all created resources? (y/n): "
CLEANUP_CHOICE="y"

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    echo "Cleaning up resources..." | tee -a "$LOG_FILE"
    cleanup_resources
    echo "Cleanup completed." | tee -a "$LOG_FILE"
else
    echo "Resources will not be cleaned up. You can manually clean them up
 later." | tee -a "$LOG_FILE"
fi

echo ""
echo "Script execution completed. See $LOG_FILE for details."

# Clean up temporary files
rm -f ssm-onboarding-policy.json trust-policy.json ssm-config.json 2>/dev/null ||
 true
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateConfigurationManager
  - CreatePolicy
  - CreateRole
  - DeleteConfigurationManager
  - DeletePolicy

- DeleteRole
- DetachRolePolicy
- GetCallerIdentity
- GetConfigurationManager

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Sicurezza in IAM e AWS STS

La sicurezza del cloud ha AWS la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili a AWS Identity and Access Management (IAM), consulta AWS Services in Scope by Compliance Program AWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Inoltre, sei responsabile anche di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizzano AWS Identity and Access Management (IAM) e Servizio di token di sicurezza AWS (AWS STS). I seguenti argomenti mostrano come configurare IAM e AWS STS soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse IAM.

Indice

- AWS credenziali di sicurezza
- AWS linee guida per l'audit di sicurezza
- Protezione dei dati in AWS Identity and Access Management
- Registrazione e monitoraggio AWS Identity and Access Management
- Convalida della conformità per AWS Identity and Access Management
- Resilienza in AWS Identity and Access Management
- Sicurezza dell'infrastruttura in AWS Identity and Access Management
- Configurazione e analisi delle vulnerabilità in AWS Identity and Access Management

- AWS politiche gestite per AWS Identity and Access Management, IAM Access Analyzer e gestore degli accessi agli account
- Funzioni di sicurezza al di fuori di IAM

AWS credenziali di sicurezza

Quando interagisci con AWS, specifichi le tue credenziali di AWS sicurezza per verificare chi sei e se disponi dell'autorizzazione ad accedere alle risorse che stai richiedendo. AWS utilizza le credenziali di sicurezza per autenticare e autorizzare le richieste.

Ad esempio, se si desidera scaricare un file protetto da un bucket Amazon Simple Storage Service (Amazon S3), è necessario che le credenziali consentano tale accesso. Se le tue credenziali non mostrano che sei autorizzato a scaricare il file, AWS respinge la tua richiesta. Tuttavia, le tue credenziali di AWS sicurezza non sono necessarie per scaricare un file in un bucket Amazon S3 condiviso pubblicamente.

Esistono diversi tipi di utenti AWS, ognuno con le proprie credenziali di sicurezza:

- Proprietario dell'account (utente root): l'utente che ha creato l' Account AWS e ha accesso completo.
- Centro identità AWS IAM utenti: utenti gestiti in Centro identità AWS IAM.
- Responsabili federati: utenti di provider di identità esterni a cui è concesso l'accesso temporaneo AWS tramite la federazione. Per ulteriori informazioni sulle identità federate, consulta la pagina Fornitori di identità e federazione in AWS.
- Utenti IAM: singoli utenti creati all'interno del servizio AWS Identity and Access Management (IAM).

Gli utenti dispongono di credenziali di sicurezza a lungo termine o temporanee. L'utente root, l'utente IAM e le chiavi di accesso dispongono di credenziali di sicurezza a lungo termine che non scadono. Per proteggere le credenziali a lungo termine, è consigliabile disporre di procedure per gestire le chiavi di accesso, modificare le password e abilitare l'MFA.

Per semplificare la gestione delle credenziali degli utenti root tra gli account dei membri in AWS Organizations, è possibile proteggere centralmente le credenziali utente root dell'utilizzo Account AWS gestito. AWS OrganizationsGestire centralmente l'accesso root per gli account membriconsente di rimuovere e impedire centralmente il ripristino a lungo termine delle credenziali degli utenti root, prevenendo accessi root involontari su larga scala.

I ruoli IAM e utenti in Centro identità AWS IAM gli user principal AWS STS federati dispongono di credenziali di sicurezza temporanee. Le credenziali di sicurezza temporanee scadono dopo un periodo di tempo definito o quando l'utente termina la sessione. Le credenziali temporanee funzionano quasi esattamente come le credenziali a lungo termine, con le seguenti differenze:

- Le credenziali di sicurezza provvisorie sono a breve termine, come implica il nome. Possono essere configurate per durare ovunque per pochi minuti o diverse ore. Una volta scadute, le credenziali AWS non le riconosce più né consente alcun tipo di accesso alle richieste API effettuate con esse.
- Le credenziali di sicurezza temporanee non sono archiviate con l'utente, ma vengono generate dinamicamente e fornite all'utente quando richiesto. Quando (o anche prima) le credenziali di sicurezza temporanee scadono, l'utente può richiedere nuove credenziali, purché l'utente che le richiede abbia ancora le autorizzazioni per farlo.

Di conseguenza, le credenziali temporanee presentano i seguenti vantaggi rispetto alle credenziali a lungo termine:

- Non è necessario distribuire o incorporare credenziali di AWS sicurezza a lungo termine in un'applicazione.
- È possibile fornire l'accesso alle AWS risorse agli utenti senza dover definire un' AWS identità per loro. Le credenziali provvisorie sono la base dei ruoli e della federazione delle identità.
- Le credenziali di sicurezza temporanee hanno una durata limitata, perciò non è necessario aggiornarle o revocarle in modo esplicito quando non sono più necessarie. Dopo che le credenziali di sicurezza temporanee scadono, non possono essere riutilizzate. È possibile specificare quando scadono le credenziali, fino a un limite massimo.

Considerazioni relative alla sicurezza

Ti consigliamo di prendere in considerazione le informazioni seguenti nel momento in cui stabilisci le disposizioni di sicurezza per il tuo Account AWS:

- Quando crei un account Account AWS, creiamo l'utente root dell'account. Le credenziali dell'utente root (proprietario dell'account) consentono il pieno accesso a tutte le risorse nell'account. La prima operazione da eseguire con l'utente root è concedere a un altro utente le autorizzazioni amministrative Account AWS in modo da ridurre al minimo l'utilizzo dell'utente root.

- L'autenticazione a più fattori (MFA) offre un ulteriore livello di sicurezza per gli utenti che possono accedere al tuo Account AWS. Per una maggiore sicurezza, ti consigliamo di richiedere l'autenticazione MFA sulle Utente root dell'account AWS credenziali e su tutti gli utenti IAM. Per ulteriori informazioni, consulta AWS Multi-factor autenticazione in IAM.
- AWS richiede diversi tipi di credenziali di sicurezza, a seconda di come accedi AWS e del tipo di AWS utente che sei. Ad esempio, si utilizzano le credenziali di accesso Console di gestione AWS mentre si utilizzano i tasti di accesso per effettuare chiamate programmatiche a. AWSPer informazioni su come determinare il tipo di utente e la pagina di accesso, consulta Cos'è l' AWS accesso nella Guida per l'utente.Accedi ad AWS
- Non puoi utilizzare policy IAM per negare esplicitamente all'utente root l'accesso alle risorse. È possibile utilizzare solo una policy AWS Organizations di controllo del servizio (SCP) per limitare le autorizzazioni dell'utente root.
- Se dimentichi o perdi la password dell'utente root, dovrai accedere all'indirizzo e-mail associato al tuo account per reimpostarla.
- Se perdi le chiavi di accesso dell'utente root, devi essere in grado di accedere al tuo account come utente root per crearne di nuove.
- Non utilizzare l'utente root per le attività quotidiane. Utilizzalo per eseguire le attività che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono il tuo accesso come utente root, consulta la pagina Attività che richiedono credenziali dell'utente root.
- Le credenziali di sicurezza sono specifiche dell'account. Se hai accesso a più Account AWS, avrai credenziali separate per ogni account.
- Le politiche determinano le azioni che un utente, un ruolo o un membro di un gruppo di utenti può eseguire, su quali AWS risorse e in quali condizioni. Utilizzando le policy è possibile controllare in modo sicuro l'accesso Servizi AWS e le risorse presenti in. Account AWS Se devi modificare o revocare le autorizzazioni in risposta a un evento di sicurezza, puoi eliminare o modificare le policy anziché modificare direttamente l'identità.
- Assicurati di salvare in un luogo sicuro le credenziali di accesso per il tuo utente IAM per l'accesso di emergenza e tutte le chiavi di accesso che hai creato per l'accesso programmatico. Se perdi le chiavi di accesso, dovrai accedere al tuo account e crearne di nuove.
- Ti consigliamo vivamente di utilizzare le credenziali temporanee fornite dai ruoli IAM e dai principali federati anziché quelle a lungo termine fornite dagli utenti IAM e dalle chiavi di accesso.

Accesso programmatico con AWS credenziali di sicurezza

Si consiglia di utilizzare chiavi di accesso a breve termine, quando possibile, per effettuare chiamate programmatiche AWS o utilizzare l'or. AWS Command Line Interface AWS Strumenti per PowerShell Tuttavia, è possibile utilizzare anche chiavi di AWS accesso a lungo termine per questi scopi.

Quando crei una chiavi di accesso a lungo termine, crei anche l'ID chiave di accesso (ad esempio, `AKIAIOSFODNN7EXAMPLE`) e la chiave di accesso segreta (ad esempio, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). La chiave di accesso segreta può essere scaricata solo nel momento in cui viene creata. Se non si scarica la chiave di accesso segreta o se viene smarrita, è necessario crearne una nuova.

In molti scenari, non è necessaria una chiave di accesso a lungo termine a validità illimitata (come accade invece quando si creano le chiavi di accesso per un utente IAM). Al contrario, è possibile creare ruoli IAM e generare credenziali di sicurezza temporanee. Tali credenziali di sicurezza temporanee includono un ID chiave di accesso e una chiave di accesso segreta, ma includono anche un token di sicurezza che ne indica la scadenza. Dopo che scadono, non sono più valide. Per ulteriori informazioni, consulta Alternative alle chiavi di accesso a lungo termine

Gli ID delle chiavi di accesso che iniziano con `AKIA` sono chiavi di accesso a lungo termine per un utente IAM o un utente Account AWS root. Gli ID delle chiavi di accesso che iniziano con `ASIA` sono chiavi di accesso alle credenziali temporanee create utilizzando AWS STS le operazioni.

Gli utenti hanno bisogno di un accesso programmatico se vogliono interagire con persone AWS esterne a. Console di gestione AWS Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Consigliato) Utilizza le credenziali della console come credenziali temporanee per firmare richieste programmatiche a AWS SDK o AWS CLI API. AWS	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per saperne di più AWS CLI, consulta Login for AWS local development nella Guida per l'utente.

Quale utente necessita dell'accesso programmatico?	Per	Come
		AWS Command Line Interface • Per AWS gli SDK, consulta Login for AWS local development nella Guida di riferimento per AWS SDK and Tools.
Identità della forza lavoro (Utenti gestiti nel centro identità IAM)	Usa le credenziali temporane e per firmare le richieste programmatiche agli AWS SDK o alle AWS CLI API. AWS	Segui le istruzioni per l'interfa ccia che desideri utilizzare. • Per saperne di più AWS CLI, consulta Configura zione dell'uso Centro identità AWS IAM nella Guida AWS CLI per l'utente. AWS Command Line Interface • Per AWS SDK, strumenti e AWS API, consulta l'autenti cazione IAM Identity Center nella Guida di riferimento per AWS SDK e strumenti.
IAM	Usa le credenziali temporane e per firmare le richieste programmatiche agli SDK o alle API AWS CLI. AWS AWS	Seguendo le istruzioni in Uso delle credenziali temporanee con le AWS risorse nella IAM User Guide.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Non consigliato) Usa le credenziali a lungo termine per firmare richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per saperne di più AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente. AWS Command Line Interface • Per AWS SDK e strumenti, consulta Autenticare utilizzando credenziali a lungo termine nella Guida di riferimento per SDK e strumenti. AWS • Per le AWS API, consulta Gestire le chiavi di accesso per gli utenti IAM nella IAM User Guide.

Alternative alle chiavi di accesso a lungo termine

Per numerosi casi d'uso comuni, esistono delle alternative alle chiavi di accesso a lungo termine. Per migliorare la sicurezza del tuo account, considera quanto segue.

- Non incorporate le chiavi di accesso a lungo termine e le chiavi di accesso segrete nel codice dell'applicazione o in un archivio di codici: utilizzate AWS Secrets Manager invece una o un'altra soluzione per la gestione dei segreti, in modo da non dover codificare le chiavi in chiaro. L'applicazione o il client possono quindi recuperare i segreti quando necessario. Per ulteriori informazioni, vedi Che cos'è? AWS Secrets Managernella Guida AWS Secrets Manager per l'utente.

- Quando possibile, utilizza ruoli IAM per generare credenziali di sicurezza temporanee: usa sempre dei meccanismi per emettere credenziali di sicurezza temporanee anziché chiavi di accesso a lungo termine. Le credenziali di sicurezza temporanee sono più sicure perché non sono archiviate con l'utente ma vengono generate dinamicamente e fornite all'utente quando richiesto. Le credenziali di sicurezza temporanee hanno una durata limitata, quindi non è necessario gestirle o aggiornarle. I meccanismi che forniscono chiavi di accesso temporanee includono i ruoli IAM o l'autenticazione di un utente IAM Identity Center. Per le macchine che funzionano all'esterno dell' AWS utente, è possibile utilizzare AWS Identity and Access Management Roles Anywhere.
- Utilizza alternative alle chiavi di accesso a lungo termine per AWS Command Line Interface (AWS CLI) o la **`aws-shell`**: le alternative includono quanto segue.
  - Accedi per lo sviluppo AWS locale con le credenziali della console. È possibile utilizzare la AWS CLI versione 2 e il `aws login` comando per generare credenziali a breve termine per eseguire AWS CLI comandi utilizzando le credenziali della console. Per ulteriori informazioni, consulta Login for AWS local development nella Guida per l'AWS Command Line Interface utente.
  - AWS CloudShellè una shell preautenticata basata su browser che puoi avviare direttamente da. Console di gestione AWS Puoi eseguire AWS CLI comandi Servizi AWS tramite la tua shell preferita (Bash, Powershell o Z shell). Quando esegui questa operazione, non devi scaricare o installare strumenti a riga di comando. Per ulteriori informazioni, consulta Che cos'è AWS CloudShell? nella Guida per l'utente di AWS CloudShell .
- Non create chiavi di accesso a lungo termine per utenti umani che hanno bisogno di accedere alle applicazioni oppure Servizi AWS : IAM Identity Center può generare credenziali di accesso temporanee per consentire l'accesso agli utenti IdP esterni. Servizi AWS In questo modo si elimina la necessità di creare e gestire credenziali a lungo termine in IAM. In IAM Identity Center, crea un set di autorizzazioni di IAM Identity Center che conceda l'accesso agli utenti IdP esterni. Quindi assegna un gruppo da IAM Identity Center al set di autorizzazioni selezionato. Account AWS Per ulteriori informazioni, consulta What is Centro identità AWS IAM, Connect to your external identity provider e Permission sets nella Guida per l'Centro identità AWS IAM utente.
- Non archiviate le chiavi di accesso a lungo termine all'interno di un servizio di AWS elaborazione: assegnate invece un ruolo IAM alle risorse di calcolo. Ciò fornisce automaticamente le credenziali temporanee per concedere l'accesso. Ad esempio, quando crei un profilo di istanza collegato a un'istanza Amazon EC2, puoi assegnare un AWS ruolo all'istanza e renderlo disponibile per tutte le sue applicazioni. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza Amazon EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2.

AWS linee guida per l'audit di sicurezza

Controlla periodicamente la configurazione di sicurezza per accertarti che soddisfi i requisiti aziendali attuali. Grazie ai controlli puoi rimuovere ruoli, gruppi, policy e utenti IAM non necessari, inoltre puoi accertarti che gli utenti e il software dispongano solo delle autorizzazioni necessarie.

Di seguito sono riportate le linee guida per la revisione e il monitoraggio sistematici AWS delle risorse alla ricerca delle migliori pratiche di sicurezza.

> **Tip**
>
> Puoi monitorare l'uso di IAM in relazione alle best practice sulla sicurezza utilizzando AWS Security Hub CSPM. Security Hub CSPM utilizza i controlli di sicurezza per valutare le configurazioni delle risorse e gli standard di sicurezza per aiutarti a rispettare vari framework di conformità. Per ulteriori informazioni sull'utilizzo di Security Hub CSPM per valutare le risorse IAM, consulta i controlli di AWS Identity and Access Management nella Guida per l' AWS Security Hub CSPM utente.

Indice

- Quando è necessario eseguire un controllo di sicurezza
- Linee guida per l'audit
- Controlla le credenziali del tuo account AWS
- Verifica degli utenti IAM
- Verifica dei gruppi IAM
- Verifica dei ruoli IAM
- Verifica dei provider IAM per SAML e OpenID Connect (OIDC)
- Verifica le app per dispositivi mobili
- Suggerimenti per la verifica delle policy IAM

Quando è necessario eseguire un controllo di sicurezza

Controlla la configurazione di sicurezza nelle seguenti situazioni:

- Periodicamente. Come best practice per la sicurezza, segui la procedura descritta in questo documento a intervalli regolari.

- In caso di cambiamenti all'interno dell'organizzazione, ad esempio persone che lasciano l'azienda.
- Se hai smesso di utilizzare uno o più singoli AWS servizi, verifica di aver rimosso le autorizzazioni di cui gli utenti del tuo account non hanno più bisogno.
- Se hai aggiunto o rimosso software nei tuoi account, ad esempio applicazioni su istanze Amazon EC2, OpsWorks stack, modelli, ecc. CloudFormation
- Se sospetti che una persona non autorizzata possa aver eseguito l'accesso al tuo account.

Linee guida per l'audit

Quando verifichi la configurazione di sicurezza del tuo account, segui queste linee guida:

- Sii meticoloso. Esamina tutti gli elementi della configurazione di sicurezza, inclusi quelli che utilizzi raramente.
- Non dare nulla per scontato. Se non conosci a sufficienza alcuni elementi della tua configurazione di sicurezza (ad esempio il motivo della presenza di una policy specifica o dell'esistenza di un ruolo), analizza i requisiti aziendali per comprendere il potenziale rischio.
- Fai in modo che le cose siano semplici. Per facilitare i controlli (e la gestione), usa gruppi IAM, ruoli IAM, schemi di denominazione coerenti e policy semplici.

Controlla le credenziali del tuo account AWS

Segui questi passaggi quando controlli le credenziali del tuo AWS account:

1. Puoi rimuovere le eventuali chiavi di accesso associate a un utente root che non utilizzi. Ti consigliamo vivamente di non utilizzare le chiavi di accesso root per il lavoro quotidiano e di utilizzare invece utenti con credenziali temporanee, ad esempio. AWS utenti in Centro identità AWS IAM
2. Se per l'account sono necessarie chiavi di accesso, occorre aggiornarle all'occorrenza.

Verifica degli utenti IAM

Eseguire questa procedura quando si verificano gli utenti IAM esistenti:

1. Elenca gli utenti, quindi elimina gli utenti che non sono necessari.
2. Rimuovi gli utenti dai gruppi a cui non richiedono l'accesso.

3. Verifica le policy associate ai gruppi in cui si trova l'utente. Per informazioni, consulta Suggerimenti per la verifica delle policy IAM.
4. Elimina le credenziali di sicurezza di cui l'utente non ha bisogno o che potrebbero essere state esposte. Ad esempio, un utente IAM utilizzato per un'applicazione non necessita di una password (necessaria solo per accedere ai AWS siti Web). Analogamente, se un utente non utilizza chiavi di accesso, non c'è motivo per cui debba averne. Per ulteriori informazioni, consulta le pagine Gestione delle password per gli utenti IAM e Gestione delle chiavi di accesso per gli utenti IAM.

   È possibile generare e scaricare un report delle credenziali che riporta tutti gli utenti IAM presenti nell'account e lo stato delle loro diverse credenziali, tra cui password, chiavi di accesso e dispositivi MFA. Per le password e le chiavi di accesso, il report sulle credenziali mostra la data e l'orario dell'ultimo utilizzo della password o della chiave di accesso. Valuta la possibilità di rimuovere dal tuo account le credenziali che non sono state utilizzate di recente. (Non rimuovere l'utente designato per l'accesso di emergenza.) Per ulteriori informazioni, consulta Ottenere rapporti sulle credenziali per il tuo AWS account.
5. Aggiorna le password e le chiavi di accesso quando necessario per i casi d'uso che richiedono credenziali a lungo termine. Per ulteriori informazioni, consulta le pagine Gestione delle password per gli utenti IAM e Gestione delle chiavi di accesso per gli utenti IAM.
6. Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee. Se possibile, passa dagli utenti IAM ai principali federati, ad esempio utenti in Centro identità IAM. Mantieni il numero minimo di utenti IAM necessari alle tue applicazioni.

Verifica dei gruppi IAM

Eseguire questa procedura quando si verificano i gruppi &IAM:

1. Elenca i gruppi, quindi elimina i gruppi inutilizzati.
2. Verifica gli utenti di ciascun gruppo e rimuovi gli utenti che non appartengono al gruppo in esame.
3. Verifica le policy associate al gruppo. Per informazioni, consulta Suggerimenti per la verifica delle policy IAM.

Verifica dei ruoli IAM

Quando vengono verificati i ruoli IAM, completare le seguenti operazioni:

1. Elenca i ruoli, quindi elimina i ruoli inutilizzati.
2. Esamina la policy di attendibilità del ruolo. Accertati di sapere chi ricopre il ruolo di principal e di comprendere perché tale account o utente debba essere in grado di assumere tale ruolo.
3. Consulta la policy di accesso del ruolo per essere sicuro che conceda le autorizzazioni idonee a chiunque assumi tale ruolo. Consulta Suggerimenti per la verifica delle policy IAM.

Verifica dei provider IAM per SAML e OpenID Connect (OIDC)

Se hai creato un'entità IAM per stabilire l'attendibilità con un gestore dell'identità digitale (IdP) SAML oppure OIDC, attieniti alla seguente procedura:

1. Elimina i provider inutilizzati.
2. Scarica e rivedi i documenti AWS sui metadati per ogni IdP SAML e assicurati che i documenti riflettano le tue attuali esigenze aziendali.
3. Ottieni i documenti di metadati più recenti da SAML IdPs e aggiorna il provider in IAM.

Verifica le app per dispositivi mobili

Se hai creato un'app mobile che effettua richieste a AWS, procedi nel seguente modo:

1. Accertati che l'app mobile non contenga chiavi di accesso incorporate, anche se sono archiviate crittografate.
2. Ottieni credenziali temporanee per l'app utilizzando APIs Designed for that purpose.

> **Note**
>
> È consigliabile utilizzare Amazon Cognito per la gestione delle identità utente nell'applicazione. Questo servizio consente di autenticare gli utenti utilizzando Login with Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC). Per ulteriori informazioni, consulta pool di identità in Amazon Cognito nella Guida per gli sviluppatori di Amazon Cognito.

Suggerimenti per la verifica delle policy IAM

Le policy sono potenti e ingegnose, per cui è importante studiare e comprendere le autorizzazioni concesse da ogni policy. Quando esamini le policy, utilizza le seguenti linee guida:

- Collega le policy a gruppi o ruoli anziché ai singoli utenti. Se un singolo utente dispone di una policy, assicurati di comprendere perché l'utente necessita di tale policy.
- Accertati che utenti, gruppi e ruoli IAM dispongano delle autorizzazioni necessarie e che non dispongano di autorizzazioni aggiuntive.
- Utilizza il simulatore di policy IAM per testare le policy collegate a utenti o gruppi.
- Ricorda che le autorizzazioni di un utente sono il risultato di tutte le politiche applicabili, sia politiche basate sull'identità (su utenti, gruppi o ruoli) che politiche basate sulle risorse (su risorse come bucket Amazon S3, code Amazon SQS, argomenti e chiavi di Amazon SNS). AWS KMS È importante esaminare tutte le policy che si applicano a un utente e comprendere il relativo set completo di autorizzazioni concesso.
- È importante sapere che consentire a un utente di creare un utente, un gruppo, un ruolo o una policy IAM e collegare una policy all'entità principal significa effettivamente concedere a tale utente le autorizzazioni per tutte le risorse presenti nell'account. Gli utenti che possono creare policy e collegarle a un utente, gruppo o ruolo possono assegnare a se stessi qualunque autorizzazione. In generale, non concedere a utenti o ruoli che non ritieni attendibili autorizzazioni IAM per l'accesso completo alle risorse del tuo account. Quando esegui i controlli di sicurezza, accertati che le seguenti autorizzazioni IAM siano concesse a identità attendibili:
  - `iam:PutGroupPolicy`
  - `iam:PutRolePolicy`
  - `iam:PutUserPolicy`
  - `iam:CreatePolicy`
  - `iam:CreatePolicyVersion`
  - `iam:AttachGroupPolicy`
  - `iam:AttachRolePolicy`
  - `iam:AttachUserPolicy`
- Assicurati che le policy non concedano autorizzazioni per servizi che non utilizzi. Ad esempio, se utilizzi politiche gestite, assicurati che le politiche AWS gestite utilizzate nel tuo account AWS riguardino i servizi che utilizzi effettivamente. Per scoprire quali policy AWS gestite sono in uso nel

tuo account, utilizza l'`GetAccountAuthorizationDetails`API IAM (AWS CLI comando: `aws iam get-account-authorization-details`).

- Se la policy concede a un utente l'autorizzazione ad avviare un'istanza di Amazon EC2, potrebbe consentire anche l'operazione `iam:PassRole`, ma in questo caso dovrebbe elencare in modo esplicito i ruoli che l'utente può passare all'istanza di Amazon EC2.
- Esamina tutti i valori per l'elemento `Action` o `Resource` che includono `*`. Quando è possibile, concedi l'accesso `Allow` alle singole operazioni e risorse necessarie agli utenti. Tuttavia, quelle che seguono sono ragioni per cui potrebbe essere utile utilizzare `*` in una policy:
  - La policy è concepita per la concessione di autorizzazioni a livello amministrativo.
  - Per comodità, il carattere jolly viene utilizzato per un set di operazioni simili (ad esempio, `Describe*`): hai così a disposizione l'elenco completo delle operazioni a cui viene fatto riferimento in questo modo.
  - Il carattere jolly viene utilizzato per indicare una classe di risorse o il percorso di una risorsa (ad esempio, `arn:aws:iam::account-id:users/division_abc/*`); puoi concedere l'accesso a tutte le risorse in tale classe o percorso con la massima tranquillità.
  - Un'operazione di servizio non supporta autorizzazioni a livello di risorsa; l'unica scelta per una risorsa è `*`.
- Esamina i nomi delle policy per assicurarti che riflettano la funzione della policy stessa. Ad esempio, sebbene possa avere un nome che includa la dicitura "di sola lettura", la policy potrebbe effettivamente concedere le autorizzazioni di scrittura o modifica.

Per ulteriori informazioni sulla pianificazione dell'audit di sicurezza, consulta la pagina Best practice per sicurezza, identità e conformità nel Centro di architettura AWS .

Protezione dei dati in AWS Identity and Access Management

Il modello di responsabilità AWS condivisa (modello di) si applica alla protezione dei dati in AWS Identity and Access Management. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta Domande frequenti sulla privacy dei dati . Per informazioni sulla protezione dei dati in Europa, consulta il General Data Protection Regulation (GDPR) Center.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e di configurare i singoli utenti con Centro identità AWS IAM or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con IAM o altri utenti Servizi AWS utilizzando la console, l'API o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Crittografia dei dati in IAM e AWS STS

La crittografia dei dati in genere rientra in due categorie: crittografia dei dati a riposo e crittografia dei dati in transito.

Crittografia dei dati a riposo

I dati raccolti e archiviati da IAM vengono crittografati quando sono inattivi.

- IAM: i dati raccolti e archiviati all'interno di IAM includono indirizzi IP, metadati dell'account cliente e dati identificativi del cliente, incluse le password. I metadati dell'account cliente e i dati identificativi del cliente vengono crittografati quando sono inattivi utilizzando AES 256 e SHA 256 per gli hash.
- AWS STS— AWS STS non raccoglie i contenuti dei clienti ad eccezione dei registri di servizio che registrano le richieste riuscite, errate e difettose al servizio.

Crittografia dei dati in transito

I dati identificativi del cliente, incluse le password, vengono crittografati in transito utilizzando TLS 1.2 e 1.3. Tutti gli AWS STS endpoint supportano HTTPS per la crittografia dei dati in transito. Per un elenco degli AWS STS endpoint, consulta. AWS STS Regioni ed endpoint

Gestione delle chiavi in IAM e AWS STS

Non è possibile gestire le chiavi di crittografia utilizzando IAM o AWS STS. Per ulteriori informazioni sulle chiavi di crittografia, consulta Cos'è AWS KMS? nella Guida per gli AWS Key Management Service sviluppatori

Privacy del traffico di rete in IAM e AWS STS

Le richieste a IAM devono essere effettuate utilizzando il protocollo TLS (Transport Layer Security Protocol). È possibile proteggere le connessioni al AWS STS servizio utilizzando gli endpoint VPC. Per ulteriori informazioni, consulta Endpoint VPC di interfaccia.

Registrazione e monitoraggio AWS Identity and Access Management

Il monitoraggio è una parte importante del mantenimento dell'affidabilità, della disponibilità e delle prestazioni di AWS Identity and Access Management (IAM), Servizio di token di sicurezza AWS (AWS STS) e AWS delle altre soluzioni. AWS fornisce diversi strumenti per monitorare le AWS risorse e rispondere a potenziali incidenti:

- AWS CloudTrailacquisisce tutte le chiamate API per IAM e AWS STS come eventi, incluse le chiamate dalla console e le chiamate API. Per saperne di più sull'utilizzo CloudTrail con IAM and AWS STS, consultaRegistrazione di IAM e AWS STS Chiamate API con AWS CloudTrail. Per ulteriori informazioni in merito CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

- AWS Identity and Access Management e Access Analyzer ti aiuta a identificare le risorse della tua organizzazione e dei tuoi account, come i bucket Amazon S3 o i ruoli IAM, che sono condivise con un'entità esterna. In questo modo puoi identificare l'accesso non intenzionale alle risorse e ai dati, che rappresenta un rischio per la sicurezza. Per ulteriori informazioni, consulta Cos'è IAM Access Analyzer?
- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. È possibile raccogliere e tenere traccia dei parametri, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue EC2 istanze Amazon e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.
- Amazon CloudWatch Logs ti aiuta a monitorare, archiviare e accedere ai tuoi file di registro da EC2 istanze Amazon e altre fonti. CloudTrail CloudWatch I log possono monitorare le informazioni nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Amazon CloudWatch Logs User Guide.

Per ulteriori risorse e best practice per la sicurezza di IAM, consulta Migliori pratiche di sicurezza e casi d'uso in AWS Identity and Access Management.

Argomenti

- Registrazione di IAM e AWS STS Chiamate API con AWS CloudTrail
- Tieni traccia delle attività privilegiate in AWS CloudTrail

Registrazione di IAM e AWS STS Chiamate API con AWS CloudTrail

IAM e AWS STS sono integrati con AWS CloudTrail un servizio che fornisce un registro delle azioni intraprese da un utente o ruolo IAM. CloudTrail acquisisce tutte le chiamate API per IAM e AWS STS come eventi, incluse le chiamate dalla console e dalle chiamate API. Se crei un percorso, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3. Se non configuri un trail, è comunque possibile visualizzare gli eventi più recenti nella console di CloudTrail in Event history (Cronologia eventi). Puoi utilizzarlo CloudTrail per ottenere informazioni sulla richiesta effettuata a IAM o. AWS STS Ad esempio, puoi visualizzare l'indirizzo IP da cui è stata Effettuata la richiesta, l'autore della richiesta e il momento in cui è stata effettuata, nonché dettagli aggiuntivi.

Per saperne di più CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

Argomenti

- IAM e AWS STS informazioni in CloudTrail
- Registrazione di IAM e AWS STS Richieste API
- Registrazione delle richieste API su altri AWS services
- Registrazione di eventi di accesso dell'utente
- Registrazione di eventi di accesso per credenziali temporanee
- Esempi di eventi dell'API IAM nel registro CloudTrail
- Esempio AWS STS Eventi API nel registro CloudTrail
- Esempi di eventi di accesso nel registro CloudTrail
- Comportamento della policy di attendibilità del ruolo IAM

IAM e AWS STS informazioni in CloudTrail

CloudTrail è abilitato sull'utente al Account AWS momento della creazione dell'account. Quando si verifica un'attività in IAM o AWS STS, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo Account AWS. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo Account AWS, compresi gli eventi per IAM e AWS STS, crea un percorso. Un percorso consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le Regioni . Il percorso registra gli eventi provenienti da tutte le regioni della AWS partizione e invia i file di registro al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati degli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutti gli IAM e AWS STS le azioni vengono registrati CloudTrail e documentati nell'IAM API Reference e nell'Servizio di token di sicurezza AWS API Reference.

Registrazione di IAM e AWS STS Richieste API

CloudTrail registra tutte le richieste API autenticate in IAM e nelle operazioni AWS STS API. CloudTrail registra inoltre le richieste non autenticate relative alle AWS STS azioni `AssumeRoleWithSAML` e registra `AssumeRoleWithWebIdentity` le informazioni fornite dal provider di identità. Tuttavia, alcune AWS STS richieste non autenticate potrebbero non essere registrate perché non soddisfano l'aspettativa minima di essere sufficientemente valide da essere considerate attendibili come richiesta legittima. CloudTrail non registra AWS STS le richieste negate nell'account di destinazione per i presupposti di ruolo tra più account e le sessioni privilegiate avviate tramite l'API. `AssumeRoot`

Queste informazioni registrate possono essere utilizzate per mappare le chiamate effettuate da un principale federato OIDC o SAML con un ruolo assunto al chiamante federato esterno di origine. In caso di`AssumeRole`, è possibile mappare le chiamate al AWS servizio di origine o all'account dell'utente di origine. La `userIdentity` sezione dei dati JSON nella voce di CloudTrail registro contiene le informazioni necessarie per mappare la AssumeRole* richiesta con un responsabile di sessione specifico. Per ulteriori informazioni, vedere CloudTrail UserIdentity Element nella Guida per l'AWS CloudTrail utente.

AWS CloudTrail i log conterranno informazioni MFA quando l'utente IAM accede con MFA. Se l'utente IAM assume un ruolo IAM, CloudTrail `mfaAuthenticated: true` accederà anche `sessionContext` agli attributi per le azioni eseguite utilizzando il ruolo assunto. Tuttavia, la CloudTrail registrazione è separata da ciò che richiede IAM quando le chiamate API vengono effettuate con le credenziali del ruolo assunto. Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Ad esempio, le chiamate a IAM `CreateUser` e `DeleteRole` `ListGroups` ad altre operazioni API vengono tutte registrate. CloudTrail

Esempi di questo tipo di voce di log sono riportati più avanti in questo argomento.

Registrazione delle richieste API su altri AWS services

Le richieste autenticate ad altre operazioni dell'API del AWS servizio vengono registrate CloudTrail e queste voci di registro contengono informazioni su chi ha generato la richiesta.

Ad esempio, supponiamo di effettuare una richiesta per elencare le istanze Amazon EC2 o creare un gruppo di implementazione AWS CodeDeploy . I dettagli sulla persona o sul servizio che ha effettuato la richiesta sono contenuti nella voce di log per quella richiesta. Queste informazioni consentono di determinare se la richiesta è stata effettuata da un utente IAM Utente root dell'account AWS, da un ruolo o da un altro AWS servizio.

Per maggiori dettagli sulle informazioni sull'identità dell'utente nelle voci di CloudTrail registro, consulta UserIdentity Element nella Guida per l'AWS CloudTrail utente.

Registrazione di eventi di accesso dell'utente

CloudTrail registra gli eventi di accesso a strumenti di sviluppo locali come SDK, forum di AWS discussione AWS CLI e. Console di gestione AWS Marketplace AWS CloudTrailregistra i tentativi di accesso riusciti e non riusciti per gli utenti IAM, i principali federati SAML e OIDC e gli user principal federati. AWS STS

Per visualizzare CloudTrail gli eventi di esempio relativi agli accessi degli utenti root riusciti e non riusciti, consulta Example event records for root users nella Guida per l'utente. AWS CloudTrail

Come best practice di sicurezza, AWS non registra il testo del nome utente IAM immesso quando l'errore di accesso è causato da un nome utente errato. Il nome utente viene mascherato dal valore `HIDDEN_DUE_TO_SECURITY_REASONS`. Per un esempio di questo caso, consultare Esempio di evento di accesso non riuscito a causa di un nome utente non corretto più avanti in questo argomento. Il testo del nome utente viene oscurato perché tali errori potrebbero essere causati da errori utente. La registrazione di questi errori potrebbe esporre informazioni potenzialmente sensibili. Ad esempio:

- L'utente ha immesso accidentalmente la password nella casella del nome utente.
- Scegli il link per la pagina di accesso di uno Account AWS, ma poi digiti il numero di account per un altro. Account AWS
- L'utente ha dimenticato a quale account stava eseguendo l'accesso e ha accidentalmente digitato il nome dell'account e-mail personale, l'identificatore di accesso bancario o un altro ID privato.

Registrazione di eventi di accesso per credenziali temporanee

Quando un principale richiede credenziali temporanee, il tipo di principale determina il modo in cui CloudTrail registra l'evento. Questo può essere complicato quando un principale assume un ruolo in un altro account. Esistono più chiamate API per eseguire operazioni correlate a operazioni

multiaccount al ruolo. Innanzitutto, il principale chiama un' AWS STS API per recuperare le credenziali temporanee. Tale operazione viene registrata nell'account chiamante e nell'account in cui viene AWS STS eseguita l'operazione. Quindi l'entità principale utilizza il ruolo per eseguire altre chiamate API nell'account del ruolo assunto.

È possibile utilizzare la chiave di condizione `sts:SourceIdentity` nella policy di attendibilità del ruolo per richiedere agli utenti di specificare un'identità quando assumono un ruolo. Ad esempio, è possibile richiedere che gli utenti IAM specifichino il proprio nome utente come identità di origine. In questo modo è possibile determinare quale utente ha eseguito un'operazione specifica in AWS. Per ulteriori informazioni, consulta `sts:SourceIdentity`. Puoi utilizzare `sts:RoleSessionName` anche per richiedere agli utenti di specificare un nome di sessione quando assumono un ruolo. Questo può aiutarti a distinguere le sessioni di ruolo per un ruolo utilizzato da diversi responsabili durante la revisione dei registri. AWS CloudTrail

La tabella seguente mostra come vengono CloudTrail registrate le diverse informazioni sull'identità utente per ciascuna delle AWS STS API che generano credenziali temporanee.

Tipo di entità principale	API STS	Identità utente nel CloudTrail registro per l'account del chiamante	Identità utente nel CloudTrail registro per l'account del ruolo assunto	Identità dell'utente nel CloudTrail registro per le successive chiamate API del ruolo
Utente root dell'account AWS credenziali	GetSessionToken	Identità utente root	L'account proprietario del ruolo è uguale all'account chiamante	Identità utente root
Utente root dell'account AWS credenziali	AssumeRoot	Sessione dell'utente root	Numero di account e ID principale (se utente)	Sessione dell'utente root
Utente IAM	GetSessionToken	Identità utente IAM	L'account proprietario del ruolo è uguale	Identità utente IAM

Tipo di entità principale	API STS	Identità utente nel CloudTrail registro per l'account del chiamante	Identità utente nel CloudTrail registro per l'account del ruolo assunto	Identità dell'utente nel CloudTrail registro per le successive chiamate API del ruolo
			all'account chiamante	
Utente IAM	GetFederationToken	Identità utente IAM	L'account proprietario del ruolo è uguale all'account chiamante	Identità utente IAM
Utente IAM	AssumeRole	Identità utente IAM	Numero di conto e ID principale (se utente) o principale AWS del servizio	Solo identità ruolo (nessun utente)
Utente autenticato esternamente	AssumeRoleWithSAML	n/a	Identità utente SAML	Solo identità ruolo (nessun utente)
Utente autenticato esternamente	AssumeRoleWithWebIdentity	n/a	OIDC/Web identità dell'utente	Solo identità ruolo (nessun utente)

CloudTrail considera un'azione di sola lettura se non ha alcun effetto mutante su una risorsa. Quando si registra un evento di sola CloudTrail lettura, oscura le informazioni nel registro. `responseElements` Quando CloudTrail registra un evento che non è di sola lettura, nella voce di registro `responseElements` viene visualizzato il testo completo. Per le AWS STS API e `AssumeRole` `AssumeRoleWithSAML``AssumeRoleWithWebIdentity`, anche se sono registrate in sola lettura, CloudTrail includerà la versione completa `responseElements` tranne `secretAccessKey` nel registro per queste API.

La tabella seguente mostra in che modo CloudTrail i registri `responseElements` e `readOnly` le informazioni per ciascuna delle AWS STS API generano credenziali temporanee.

API STS	Informazioni sugli elementi della risposta	Read-only
AssumeRole	Incluso	true
AssumeRoleWithSAML	Incluso	true
AssumeRoleWithWebIdentity	Incluso	true
AssumeRoot	Incluso	false
GetFederationToken	Incluso	false
GetSessionToken	Incluso	false

Esempi di eventi dell'API IAM nel registro CloudTrail

CloudTrail i file di registro contengono eventi formattati utilizzando JSON. Un evento API rappresenta una singola richiesta API e include informazioni sul principale, l'operazione richiesta, gli eventuali parametri e la data e l'ora dell'operazione.

Esempio di evento API IAM nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata per l'`GetUserPolicy`azione IAM.

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:Role-Session-Name",
        "arn": "arn:aws:sts::111122223333:assumed-role/Role-Name/Role-Session-Name",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
```

```
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2024-09-09T17:50:16Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-09-09T17:51:44Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "GetUserPolicy",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.101",
    "userAgent": "aws-cli/1.16.96 Python/2.7.8 Linux/10 botocore/1.12.86",
    "requestParameters": {
        "userName": "ExampleIAMUserName",
        "policyName": "ExamplePoliccyName"
    },
    "responseElements": null,
    "requestID": "9EXAMPLE-0c68-11e4-a24e-d5e16EXAMPLE",
    "eventID": "cEXAMPLE-127e-4632-980d-505a4EXAMPLE",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "iam.amazonaws.com"
    }
}
```


Da queste informazioni sull'evento puoi determinare che la richiesta è stata effettuata per ottenere una policy utente denominata `ReadOnlyAccess-JaneDoe-201407151307` per l'utente `JaneDoe`, come specificato nell'elemento `requestParameters`. Puoi anche consultare che la richiesta è stata effettuata da un utente IAM denominato `JaneDoe` in data 15 luglio 2014 alle 21:40 UTC. In questo caso, la richiesta ha avuto origine in Console di gestione AWS, come si può vedere dall'`userAgent`elemento.

Esempio AWS STS Eventi API nel registro CloudTrail

CloudTrail i file di registro contengono eventi formattati utilizzando JSON. Un evento API rappresenta una singola richiesta API e include informazioni sul principale, l'operazione richiesta, gli eventuali parametri e la data e l'ora dell'operazione.

Esempio di account multiplo AWS STS Eventi API nei CloudTrail file di registro

L'utente IAM indicato `John` nell'account 777788889999 chiama l' AWS STS AssumeRoleazione per assumere il ruolo `EC2-dev` nell'account 111122223333. Quando si assume il ruolo, l'amministratore dell'account richiede agli utenti di impostare un'identità di origine uguale al proprio nome utente. L'utente passa il valore dell'identità di origine di `John`.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAQRSTUVWXYZEXAMPLE",
    "arn": "arn:aws:iam::777788889999:user/John",
    "accountId": "777788889999",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "John"
  },
  "eventTime": "2014-07-18T15:07:39Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRole",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "aws-cli/1.11.10 Python/2.7.8
 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67",
  "requestParameters": {
    "roleArn": "arn:aws:iam::111122223333:role/EC2-dev",
    "roleSessionName": "John-EC2-dev",
    "sourceIdentity": "John",
    "serialNumber": "arn:aws:iam::777788889999:mfa"
  },
  "responseElements": {
    "credentials": {
      "sessionToken": "<encoded session token blob>",
      "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
      "expiration": "Jul 18, 2023, 4:07:39 PM"
      },
    "assumedRoleUser": {
```

```
      "assumedRoleId": "AIDAQRSTUVWXYZEXAMPLE:John-EC2-dev",
      "arn": "arn:aws:sts::111122223333:assumed-role/EC2-dev/John-EC2-dev"
    },
    "sourceIdentity": "John",
    "sessionTokenSize": 1428,
    "sessionTokenUtilization": 34
  },
  "resources": [
    {
      "ARN": "arn:aws:iam::111122223333:role/EC2-dev",
      "accountId": "111122223333",
      "type": "AWS::IAM::Role"
    }
  ],
  "requestID": "4EXAMPLE-0e8d-11e4-96e4-e55c0EXAMPLE",
  "sharedEventID": "bEXAMPLE-efea-4a70-b951-19a88EXAMPLE",
  "eventID": "dEXAMPLE-ac7f-466c-a608-4ac8dEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Il secondo esempio mostra la voce di CloudTrail registro dell'account con ruolo assunto (111122223333) per la stessa richiesta.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AWSAccount",
    "principalId": "AIDAQRSTUVWXYZEXAMPLE",
    "accountId": "777788889999"
  },
  "eventTime": "2014-07-18T15:07:39Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRole",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "aws-cli/1.11.10 Python/2.7.8
 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67",
  "requestParameters": {
    "roleArn": "arn:aws:iam::111122223333:role/EC2-dev",
    "roleSessionName": "John-EC2-dev",
    "sourceIdentity": "John",
    "serialNumber": "arn:aws:iam::777788889999:mfa"
```

```
  },
  "responseElements": {
    "credentials": {
      "sessionToken": "<encoded session token blob>",
      "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
      "expiration": "Jul 18, 2014, 4:07:39 PM"
    },
    "assumedRoleUser": {
      "assumedRoleId": "AIDAQRSTUVWXYZEXAMPLE:John-EC2-dev",
      "arn": "arn:aws:sts::111122223333:assumed-role/EC2-dev/John-EC2-dev"
      },
    "sourceIdentity": "John",
    "sessionTokenSize": 1428,
    "sessionTokenUtilization": 34
  },
  "requestID": "4EXAMPLE-0e8d-11e4-96e4-e55c0EXAMPLE",
  "sharedEventID": "bEXAMPLE-efea-4a70-b951-19a88EXAMPLE",
  "eventID": "dEXAMPLE-ac7f-466c-a608-4ac8dEXAMPLE"
}
```

Esempio AWS STS evento API di concatenamento dei ruoli nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata da John Doe nell'account 1111. John in precedenza utilizzava il suo utente `John` per assumere il ruolo `JohnRole1`. Per questa richiesta, utilizza le credenziali di tale ruolo per assumere il ruolo `JohnRole2`. Questo è noto come concatenazione del ruolo. L'identità di origine che ha impostato quando ha assunto il ruolo `John1` persiste nella richiesta per assumere `JohnRole2`. Se John prova a impostare un'identità di origine diversa quando assume il ruolo, la richiesta viene rifiutata. John passa due tag di sessione nella richiesta. Imposta questi due tag come transitivi. La richiesta eredita il tag `Department` come transitivo perché John lo ha impostato come transitivo quando ha assunto `JohnRole1`. Per ulteriori informazioni sull'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti. Per ulteriori informazioni sulle chiavi transitive nelle catene di ruoli, consulta Concatenamento dei ruoli con i tag di sessione.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIN5ATK5U7KEXAMPLE:JohnRole1",
        "arn": "arn:aws:sts::111111111111:assumed-role/John/JohnRole1",
        "accountId": "111111111111",
```

```
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2019-10-02T21:50:54Z"
            },
                "sessionIssuer": {
                "type": "Role",
               "principalId": "AROAIN5ATK5U7KEXAMPLE",
                "arn": "arn:aws:iam::111111111111:role/JohnRole1",
                "accountId": "111111111111",
                "userName": "John"
            },
            "sourceIdentity": "John"
        }
    },
    "eventTime": "2019-10-02T22:12:29Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "123.145.67.89",
    "userAgent": "aws-cli/1.16.248 Python/3.4.7
Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 botocore/1.12.239",
    "requestParameters": {
        "incomingTransitiveTags": {
            "Department": "Engineering"
        },
        "tags": [
            {
                "value": "johndoe@example.com",
                "key": "Email"
            },
            {
                "value": "12345",
                "key": "CostCenter"
            }
        ],
        "roleArn": "arn:aws:iam::111111111111:role/JohnRole2",
        "roleSessionName": "Role2WithTags",
        "sourceIdentity": "John",
        "transitiveTagKeys": [
            "Email",
            "CostCenter"
        ],
```

```
        "durationSeconds": 3600
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
            "expiration": "Oct 2, 2019, 11:12:29 PM",
            "sessionToken": "AgoJb3JpZ2luX2VjEB4aCXVzLXdlc3QtMSJHMEXAMPLETOKEN
+//rJb8Lo30mFc5MlhFCEbubZvEj0wHB/mDMwIgSEe9gk/Zjr09tZV7F1HDTMhmEXAMPLETOKEN/iEJ/
rkqngII9//////////
ARABGgw0MjgzMDc4NjM5NjYiDLZjZFKwP4qxQG5sFCryASO4UPz5qE97wPPH1eLMvs7CgSDBSWfonmRTCfokm2FN1+hWUdQ
+C+WKFZb701eiv9J5La2EXAMPLETOKEN/c7S5Iro1WUJ0q3Cxuo/8HUoSxVhQHM7zF7mWWLhXLEQ52ivL
+F6q5dpXu4aTFedpMfnJa8JtkWwG9x1Axj0Ypy2ok8v5unpQGWych1vwdvj6ez1Dm8Xg1+qIzXILiEXAMPLETOKEN/
vQGqu8H+nxp3kabcrtOvTFTvxX6vsc8OGwUfHhzAfYGEXAMPLETOKEN/
L6v1yMM3B1OwFOrQBno1HEjf1oNI8RnQiMNFdUOtwYj7HUZIOCZmjfN8PPHq77N7GJl9lzvIZKQA0Owcjg
+mc78zHCj8y0siY8C96paEXAMPLETOKEN/
E3cpksxWdgs91HRzJWScjN2+r2LTGjYhyPqcmFzzo2mCE7mBNEXAMPLETOKEN/oJy
+2o83YNW5tOiDmczgDzJZ4UKR84yGYOMfSnF4XcEJrDgAJ3OJFwmTcTQICAlSwLEXAMPLETOKEN"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROAIFR7WHDTSOYQYHFUE:Role2WithTags",
            "arn": "arn:aws:sts::111111111111:assumed-role/test-role/Role2WithTags"
        },
        "sourceIdentity": "John",
        "sessionTokenSize": 1652,
        "sessionTokenUtilization": 40
    },
    "requestID": "b96b0e4e-e561-11e9-8b3f-7b396EXAMPLE",
    "eventID": "1917948f-3042-46ec-98e2-62865EXAMPLE",
    "resources": [
        {
            "ARN": "arn:aws:iam::111111111111:role/JohnRole2",
            "accountId": "111111111111",
            "type": "AWS::IAM::Role"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111111111111"
}
```

Esempio AWS service AWS STS Evento API nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata da un AWS servizio che chiama l'API di un altro servizio utilizzando le autorizzazioni di un ruolo di servizio. Mostra la voce di CloudTrail registro per la richiesta effettuata nell'account 777788889999.


```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAQRSTUVWXYZEXAMPLE:devdsk",
    "arn": "arn:aws:sts::777788889999:assumed-role/AssumeNothing/devdsk",
    "accountId": "777788889999",
    "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2016-11-14T17:25:26Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAQRSTUVWXYZEXAMPLE",
        "arn": "arn:aws:iam::777788889999:role/AssumeNothing",
        "accountId": "777788889999",
        "userName": "AssumeNothing"
      }
    }
  },
  "eventTime": "2016-11-14T17:25:45Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "DeleteBucket",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "[aws-cli/1.11.10 Python/2.7.8
 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67]",
  "requestParameters": {
    "bucketName": "amzn-s3-demo-bucket"
  },
  "responseElements": null,
  "requestID": "EXAMPLE463D56D4C",
  "eventID": "dEXAMPLE-265a-41e0-9352-4401bEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "777788889999"
```

```
}
```

Esempio SAML AWS STS Evento API nel CloudTrail file di registro

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata per l' AWS STS AssumeRoleWithSAMLazione. La richiesta include gli attributi SAML `CostCenter` e `Project` che vengono passati tramite l'asserzione SAML come tag di sessione. Tali tag sono impostati come transitivi in modo da garantirne la persistenza negli scenari di concatenazione del ruolo. La richiesta include il parametro API opzionale`DurationSeconds`, rappresentato come `durationSeconds` nel CloudTrail registro, ed è impostata su `1800` secondi. La richiesta include anche l'attributo SAML `sourceIdentity`, che viene inviato nell'asserzione SAML. Se un utente utilizza le credenziali della sessione del ruolo risultante per assumere un altro ruolo, questa identità di origine persiste.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "SAMLUser",
        "principalId": "SampleUkh1i4+ExamplexL/jEvs=:SamlExample",
        "userName": "SamlExample",
        "identityProvider": "bdGOnTesti4+ExamplexL/jEvs="
    },
    "eventTime": "2023-08-28T18:30:58Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRoleWithSAML",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "aws-internal/3 aws-sdk-java/1.12.479
 Linux/5.10.186-157.751.amzn2int.x86_64 OpenJDK_64-Bit_Server_VM/17.0.7+11 java/17.0.7
 kotlin/1.3.72 vendor/Amazon.com_Inc. cfg/retry-mode/standard",
    "requestParameters": {
        "sAMLAssertionID": "_c0046cEXAMPLEb9d4b8eEXAMPLE2619aEXAMPLE",
        "roleSessionName": "MyAssignedRoleSessionName",
        "sourceIdentity": "MySAMLUser",
        "principalTags": {
            "CostCenter": "987654",
            "Project": "Unicorn",
            "Department": "Engineering"
        },
        "transitiveTagKeys": [
            "CostCenter",
            "Project"
        ],
```

```
        "roleArn": "arn:aws:iam::444455556666:role/SAMLTestRoleShibboleth",
        "principalArn": "arn:aws:iam::444455556666:saml-provider/Shibboleth",
        "durationSeconds": 1800
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": "<encoded session token blob>",
            "expiration": "Aug 28, 2023, 7:00:58 PM"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROAD35QRSTUVWEXAMPLE:MyAssignedRoleSessionName",
            "arn": "arn:aws:sts::444455556666:assumed-role/SAMLTestRoleShibboleth/
MyAssignedRoleSessionName"
        },
        "packedPolicySize": 34,
        "sessionTokenSize": 1428,
        "sessionTokenUtilization": 34,
        "subject": "SamlExample",
        "subjectType": "transient",
        "issuer": "https://server.example.com/idp/shibboleth",
        "audience": "https://signin.aws.amazon.com/saml",
        "nameQualifier": "bdGOnTesti4+ExamplexL/jEvs=",
        "sourceIdentity": "MySAMLUser"
    },
    "requestID": "6EXAMPLE-e595-11e5-b2c7-c974fEXAMPLE",
    "eventID": "dEXAMPLE-265a-41e0-9352-4401bEXAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "444455556666",
            "type": "AWS::IAM::Role",
            "ARN": "arn:aws:iam::444455556666:role/SAMLTestRoleShibboleth"
        },
        {
            "accountId": "444455556666",
            "type": "AWS::IAM::SAMLProvider",
            "ARN": "arn:aws:iam::444455556666:saml-provider/test-saml-provider"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "444455556666",
    "eventCategory": "Management",
```

```
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "sts.us-east-2.amazonaws.com"
    }
}
```

Esempio: OIDC AWS STS Evento API nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata per l' AWS STS AssumeRoleWithWebIdentityazione. La richiesta include gli attributi `CostCenter` e `Project` che vengono passati tramite il token di un gestore dell'identità digitale OpenID Connect (OIDC) come tag di sessione. Tali tag sono impostati come transitivi in modo da garantirne la persistenza negli scenari di concatenazione del ruolo. La richiesta include l'attributo `sourceIdentity` dal token del provider di identità. Se un utente utilizza le credenziali della sessione del ruolo risultante per assumere un altro ruolo, questa identità di origine persiste.

La voce di CloudTrail registro contiene anche un `additionalEventData` campo con un `identityProviderConnectionVerificationMethod` attributo. Questo attributo indica il metodo AWS utilizzato per verificare la connessione con il provider OIDC. Il valore dell'attributo sarà `IAMTrustStore` o `Thumbprint`. Il `IAMTrustStore` valore indica che la connessione con l'IdP OIDC è AWS stata verificata con successo utilizzando la nostra libreria di autorità di certificazione (CA) affidabili. Il `Thumbprint` valore indica che è stata AWS utilizzata un'impronta personale del certificato impostata nella configurazione IdP per verificare il certificato del server IdP OIDC.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "WebIdentityUser",
    "principalId": "arn:aws:iam::444455556666:oidc-provider/<issuer url of OIDC
 provider>:<id of application>:<id of user>",
    "userName": "<id of user>",
    "identityProvider": "arn:aws:iam::444455556666:oidc-provider/<issuer url of OIDC
 provider>"
  },
  "eventTime": "2024-07-09T15:41:37Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRoleWithWebIdentity",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
```

```
  "userAgent": "aws-cli/2.13.29 Python/3.11.6 Windows/10 exe/AMD64 prompt/off command/
sts.assume-role-with-web-identity",
  "requestParameters": {
    "roleArn": "arn:aws:iam::444455556666:role/FederatedWebIdentityRole",
    "roleSessionName": "<assigned role session name>",
    "sourceIdentity": "MyWebIdentityUser",
    "durationSeconds": 3600,
    "principalTags": {
      "CostCenter": "24680",
      "Project": "Pegasus"
    },
    "transitiveTagKeys": [
      "CostCenter",
      "Project"
    ]
  },
  "responseElements": {
    "credentials": {
      "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
      "sessionToken": "<encoded session token blob>",
      "expiration": "Jul 9, 2024, 4:41:37 PM"
    },
    "subjectFromWebIdentityToken": "<id of user>",
    "sourceIdentity": "MyWebIdentityUser",
    "assumedRoleUser": {
      "assumedRoleId": "AROA123456789EXAMPLE:<assigned role session name>",
      "arn": "arn:aws:sts::444455556666:assumed-role/FederatedWebIdentityRole/<assigned
 role session name>"
    },
    "provider": "arn:aws:iam::444455556666:oidc-provider/<issuer url of OIDC
 provider>",
    "audience": "<id of application>",
    "sessionTokenSize": 1536,
    "sessionTokenUtilization": 37
  },
  "additionalEventData": {
    "identityProviderConnectionVerificationMethod": "IAMTrustStore"
  },
  "requestID": "aEXAMPLE-0b26-40df-8973-c7012EXAMPLE",
  "eventID": "aEXAMPLE-ee29-4ac0-a0ed-3f5c5EXAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "444455556666",
```

```
      "type": "AWS::IAM::Role",
      "ARN": "arn:aws:iam::444455556666:role/FederatedWebIdentityRole"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "444455556666",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_128_GCM_SHA256",
    "clientProvidedHostHeader": "sts.us-east-2.amazonaws.com"
  }
}
```

Esempio AWS STS Evento API che utilizza l'endpoint globale nel file di registro CloudTrail

Per le richieste all'endpoint globale Servizio di token di sicurezza AWS (AWS STS) (`https://sts.amazonaws.com`), AWS STS include campi di AWS CloudTrail registro aggiuntivi: `endpointType` e. `awsServingRegion` Questi campi vengono visualizzati sotto l'`additionalEventDataRequestDetails`elemento per registrare il tipo di server Regione AWS e di endpoint chiamato. Il campo `endpointType` può avere un valore di `global` o `regional` per indicare il tipo di endpoint globale che ha servito la richiesta. Per ulteriori informazioni sulle modifiche AWS STS globali agli endpoint, vedere. AWS STS Regioni ed endpoint

> **Note**
>
> AWS CloudTrail i registri delle richieste effettuate all'endpoint AWS STS globale verranno inviati alla regione Stati Uniti orientali (Virginia settentrionale). CloudTrail i registri delle richieste servite dagli endpoint AWS STS regionali continueranno a essere registrati nelle rispettive aree geografiche. CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per una AWS STS richiesta effettuata all'endpoint globale (`https://sts.amazonaws.com`) proveniente dalla regione Europa (Stoccolma) - eu-north-1. Il valore del `endpointType` campo di `global` indica che la AWS STS richiesta è stata soddisfatta dall'endpoint globale nella regione Europa (Stoccolma).

```
{
    "eventVersion": "1.08",
```

```
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA123456789EXAMPLE:developer",
        "arn": "arn:aws:sts::777788889999:assumed-role/Admin/developer",
        "accountId": "777788889999",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123456789EXAMPLE",
                "arn": "arn:aws:iam::777788889999:role/Admin",
                "accountId": "777788889999",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2025-02-12T21:44:28Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2025-02-12T22:16:48Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/2.15.33 Python/3.11.8 Linux/5.10.233-204.894.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/sts.assume-role",
    "requestParameters": {
        "roleArn": "arn:aws:iam::777788889999:role/test-role",
        "roleSessionName": "test-global-assume-role"
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": "<encoded session token blob>",
            "expiration": "Feb 12, 2025, 11:16:48 PM"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROA987654321EXAMPLE:test-global-assume-role",
            "arn": "arn:aws:sts::777788889999:assumed-role/test-role/test-global-
assume-role"
        },
        "sessionTokenSize": 1096,
```

```
        "sessionTokenUtilization": 26
    },
    "additionalEventData": {
        "RequestDetails": {
            "awsServingRegion": "eu-north-1",
            "endpointType": "global"
        }
    },
    "requestID": "EXAMPLE7-2497-457a-9586-f21feEXAMPLE",
    "eventID": "EXAMPLEc-3d26-4c3a-9c94-722a9EXAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "777788889999",
            "type": "AWS::IAM::Role",
            "ARN": "arn:aws:iam::777788889999:role/test-role"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "777788889999",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "sts-global.eu-north-1.amazonaws.com"
    }
}
```

Per fare un confronto, l'esempio seguente mostra una voce di CloudTrail registro per una AWS STS richiesta effettuata all'endpoint regionale (`https://sts.us-west-2.amazonaws.com`) che è stata servita dall'endpoint regionale nella regione Europa (Stoccolma) - eu-north-1. Il valore del `endpointType` campo di `regional` indica che la AWS STS richiesta è stata soddisfatta dall'endpoint globale nella regione Europa (Stoccolma).

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA123456789EXAMPLE:developer",
        "arn": "arn:aws:sts::777788889999:assumed-role/Admin/developer",
        "accountId": "777788889999",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
```

```
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123456789EXAMPLE",
                "arn": "arn:aws:iam::777788889999:role/Admin",
                "accountId": "777788889999",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2025-02-12T21:44:28Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2025-02-12T22:16:30Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "eu-north-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/2.15.33 Python/3.11.8 Linux/5.10.233-204.894.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/sts.assume-role",
    "requestParameters": {
        "roleArn": "arn:aws:iam::777788889999:role/test-role",
        "roleSessionName": "test-global-assume-role"
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": "<encoded session token blob>",
            "expiration": "Feb 12, 2025, 11:16:30 PM"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROA987654321EXAMPLE:test-global-assume-role",
            "arn": "arn:aws:sts::777788889999:assumed-role/test-role/test-global-
assume-role"
        },
        "sessionTokenSize": 1096,
        "sessionTokenUtilization": 26
    },
    "additionalEventData": {
        "RequestDetails": {
            "endpointType": "regional",
            "awsServingRegion": "eu-north-1"
```

```
        }
    },
    "requestID": "EXAMPLEd-2116-4cd7-bd72-9f72fEXAMPLE",
    "eventID": "EXAMPLEd-219a-48ed-bc54-00e3cEXAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "777788889999",
            "type": "AWS::IAM::Role",
            "ARN": "arn:aws:iam::777788889999:role/test-role"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "777788889999",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "sts.eu-north-1.amazonaws.com"
    }
}
```

Esempi di eventi di accesso nel registro CloudTrail

CloudTrail i file di registro contengono eventi formattati utilizzando JSON. Un evento di accesso rappresenta una singola richiesta di accesso e include informazioni sul principale di accesso, la regione e la data e l'ora dell'operazione.

Esempio di evento di successo dell'accesso nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro relativa a un evento di accesso riuscito.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn":"arn:aws:iam::111122223333:user/John",
    "accountId": "111122223333",
    "userName": "John"
  },
  "eventTime": "2014-07-16T15:49:27Z",
```

```
  "eventSource": "signin.amazonaws.com",
  "eventName": "ConsoleLogin",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.110",
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101
 Firefox/24.0",
  "requestParameters": null,
  "responseElements": {
    "ConsoleLogin": "Success"
  },
  "additionalEventData": {
    "MobileVersion": "No",
    "LoginTo": "https://console.aws.amazon.com/s3/  ",
    "MFAUsed": "No"
  },
  "eventID": "3fcfb182-98f8-4744-bd45-10a395ab61cb"
}
```

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta di codice di autorizzazione riuscita.

```
{
"eventVersion": "1.11",
"userIdentity": {
"type": "AssumedRole",
"principalId": "AROATJHQDX737YZP72NTF:thesjain-Isengard",
"arn": "arn:aws:sts::225989345271:assumed-role/Admin/thesjain-Isengard",
"accountId": "225989345271",
"sessionContext": {
"sessionIssuer": {
"type": "Role",
"principalId": "AROATJHQDX737YZP72NTF",
"arn": "arn:aws:iam::225989345271:role/Admin",
"accountId": "225989345271",
"userName": "Admin"
},
"attributes": {
"creationDate": "2025-11-17T22:50:14Z",
"mfaAuthenticated": "false"
}
}
},
```

```
"eventTime": "2025-11-17T22:51:32Z",
"eventSource": "signin.amazonaws.com",
"eventName": "AuthorizeOAuth2Access",
"awsRegion": "us-east-1",
"sourceIPAddress": "52.94.133.136",
"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36",
"requestParameters": {
"scope": "openid",
"redirect_uri": "http://127.0.0.1:53037/oauth/callback",
"code_challenge_method": "SHA-256",
"client_id": "arn:aws:signin:::devtools/same-device"
},
"responseElements": null,
"additionalEventData": {
"success": "true",
"x-amzn-vpce-id": ""
},
"requestID": "e2854c76-1cba-4360-9fd1-5037b591466b",
"eventID": "59e1720d-3deb-44ff-933d-6828be2a860a",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "225989345271",
"eventCategory": "Management",
"tlsDetails": {
"tlsVersion": "TLSv1.3",
"cipherSuite": "TLS_AES_128_GCM_SHA256",
"clientProvidedHostHeader": "us-east-1.signin.aws.amazon.com"
}
}
```


L'esempio seguente mostra una voce di CloudTrail registro per una richiesta di creazione di token OAuth2 riuscita.


```
{
"eventVersion": "1.11",
"userIdentity": {
"type": "AssumedRole",
"principalId": "AROATJHQDX737YZP72NTF:jacobjoj-Isengard",
```

```
"arn": "arn:aws:sts::225989345271:assumed-role/Admin/jacobjoj-Isengard",
"accountId": "225989345271",
"sessionContext": {
"sessionIssuer": {
"type": "Role",
"principalId": "AROATJHQDX737YZP72NTF",
"arn": "arn:aws:iam::225989345271:role/Admin",
"accountId": "225989345271",
"userName": "Admin"
},
"attributes": {
"creationDate": "2025-11-18T20:38:10Z",
"mfaAuthenticated": "false"
}
}
},
"eventTime": "2025-11-18T20:38:44Z",
"eventSource": "signin.amazonaws.com",
"eventName": "CreateOAuth2Token",
"awsRegion": "us-east-1",
"sourceIPAddress": "15.248.6.6",
"userAgent": "aws-cli/2.32.0 md/awscrt#0.28.4 ua/2.1 os/macos#24.6.0 md/arch#arm64
 lang/python#3.13.9 md/pyimpl#CPython m/b,AA,Z,E cfg/retry-mode#standard md/
installer#exe sid/35033f4ca1bd md/prompt#off md/command#login",
"requestParameters": {
"client_id": "arn:aws:signin:::devtools/same-device"
},
"responseElements": null,
"additionalEventData": {
"success": "true",
"x-amzn-vpce-id": ""
},
"requestID": "94562943-c85b-4dc1-bf72-43b0fd42d6de",
"eventID": "0b338fac-6a10-4740-b34d-1bb6923e799e",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "225989345271",
"eventCategory": "Management",
"tlsDetails": {
"tlsVersion": "TLSv1.3",
"cipherSuite": "TLS_AES_128_GCM_SHA256",
"clientProvidedHostHeader": "us-east-1.signin.aws.amazon.com"
}
```

```
}
```

Per ulteriori dettagli sulle informazioni contenute nei file di CloudTrail registro, vedere CloudTrail Event Reference nella Guida per l'AWS CloudTrail utente.

Esempio di evento di errore di accesso nel CloudTrail file di registro

L'esempio seguente mostra una voce di CloudTrail registro per un evento di accesso non riuscito.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn":"arn:aws:iam::111122223333:user/JaneDoe",
    "accountId": "111122223333",
    "userName": "JaneDoe"
  },
  "eventTime": "2014-07-08T17:35:27Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "ConsoleLogin",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.100",
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101
 Firefox/24.0",
  "errorMessage": "Failed authentication",
  "requestParameters": null,
  "responseElements": {
    "ConsoleLogin": "Failure"
  },
  "additionalEventData": {
    "MobileVersion": "No",
    "LoginTo": "https://console.aws.amazon.com/sns",
    "MFAUsed": "No"
  },
  "eventID": "11ea990b-4678-4bcd-8fbe-62509088b7cf"
}
```

Da queste informazioni puoi determinare che il tentativo di accesso è stato effettuato da un utente IAM denominato `JaneDoe`, come riportato nell'elemento `userIdentity`. Puoi anche consultare che il tentativo di accesso non è riuscito, come indicato nell'elemento `responseElements`. Puoi

verificare che `JaneDoe` ha provato ad accedere alla console Amazon SNS alle 17:35 UTC in data 8 luglio 2014.

Esempio di evento di accesso non riuscito a causa di un nome utente non corretto

L'esempio seguente mostra una voce di CloudTrail registro per un evento di accesso non riuscito causato dall'immissione di un nome utente errato da parte dell'utente. AWS maschera il `userName` testo con `HIDDEN_DUE_TO_SECURITY_REASONS` per evitare l'esposizione di informazioni potenzialmente sensibili.


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "accountId": "123456789012",
    "accessKeyId": "",
    "userName": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "eventTime": "2015-03-31T22:20:42Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "ConsoleLogin",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101
 Firefox/24.0",
  "errorMessage": "No username found in supplied account",
  "requestParameters": null,
  "responseElements": {
    "ConsoleLogin": "Failure"
  },
  "additionalEventData": {
    "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
    "MobileVersion": "No",
    "MFAUsed": "No"
  },
  "eventID": "a7654656-0417-45c6-9386-ea8231385051",
  "eventType": "AwsConsoleSignin",
  "recipientAccountId": "123456789012"
}
```

Comportamento della policy di attendibilità del ruolo IAM

Il 21 settembre 2022, AWS ha apportato modifiche al comportamento della politica di fiducia dei ruoli di IAM per richiedere autorizzazioni esplicite in una politica di fiducia dei ruoli quando un ruolo si assume se stesso. I ruoli IAM nell'elenco dei comportamenti consentiti precedenti hanno un EventData campo aggiuntivo presente come esplicito per gli eventiTrustGrant . `AssumeRole` Il valore di `explicitTrustGrant` è false quando un ruolo nella precedente lista consentita si presume di utilizzare il comportamento precedente. Quando un ruolo nella precedente lista consentita assume se stesso ma il comportamento della policy di attendibilità dei ruoli è stato aggiornato per consentire esplicitamente al ruolo di assumere se stesso, il valore di `explicitTrustGrant` è true.

Solo un numero molto limitato di ruoli IAM è presente nell'elenco dei ruoli consentiti per il comportamento legacy e questo campo è presente nei CloudTrail log di questi ruoli solo quando si assumono autonomamente. Nella maggior parte dei casi, non è necessario che un ruolo IAM si assuma automaticamente. AWS consiglia di aggiornare i processi, il codice o le configurazioni per rimuovere questo comportamento o di aggiornare le politiche di fiducia dei ruoli per consentire esplicitamente questo comportamento. Per ulteriori informazioni, consulta Annuncio di un aggiornamento del comportamento delle policy di attendibilità dei ruoli di IAM.

Tieni traccia delle attività privilegiate in AWS CloudTrail

L'account di AWS Organizations gestione o un account amministratore delegato per IAM può eseguire alcune attività degli utenti root sugli account dei membri utilizzando l'accesso root a breve termine. Le sessioni con privilegi a breve termine forniscono credenziali temporanee utilizzabili per intraprendere azioni con privilegi su un account membro dell'organizzazione. È possibile utilizzare i passaggi seguenti per identificare le azioni intraprese dall'account di gestione o da un amministratore delegato durante la sessione `sts:AssumeRoot`.

> **Note**
>
> L'endpoint globale non è supportato per. `sts:AssumeRoot` CloudTrail registra `ConsoleLogin` gli eventi nella regione specificata per l'endpoint.

Per tenere traccia delle azioni eseguite da una sessione privilegiata nei log CloudTrail

1. Trova l'`AssumeRoot`evento nei tuoi CloudTrail registri. Questo evento viene generato quando l'account di gestione o l'amministratore delegato di IAM riceve una serie di credenziali a breve termine da `sts:AssumeRoot`.

Nell'esempio seguente, l' CloudTrail evento for AssumeRoot viene registrato nel `eventName` campo.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:JohnRole1",
        "arn": "arn:aws:sts::111111111111:assumed-role/John/JohnRole1",
        "accountId": "111111111111",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111111111111:role/John",
                "accountId": "111111111111",
                "userName": "Admin2"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-10-25T20:45:28Z",
                "mfaAuthenticated": "false"
            },
            "assumedRoot": "true"
        }
    },
    "eventTime": "2024-10-25T20:52:11Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRoot",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "requestParameters": {
        "targetPrincipal": "222222222222",
        "taskPolicyArn": {
            "arn": "arn:aws:iam::aws:policy/root-task/S3UnlockBucketPolicy"
        }
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
```

```
            "expiration": "Oct 25, 2024, 9:07:11 PM"
        }
    }
}
```

Per la procedura di accesso ai CloudTrail registri, consulta Acquisizione e visualizzazione dei file di CloudTrail registro nella Guida per l'AWS CloudTrail utente.

2. Nel registro degli CloudTrail eventi, individua il file `targetPrincipal` che specifica le azioni eseguite sull'account membro e `accessKeyId` che è unico per la `AssumeRoot` sessione.

   Nell'esempio seguente, `targetPrincipal` è 222222222222 e il è EXAMPLE`accessKeyId`. ASIAIOSFODNN7

```
"eventTime": "2024-10-25T20:52:11Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRoot",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "requestParameters": {
        "targetPrincipal": "222222222222",
        "taskPolicyArn": {
            "arn": "arn:aws:iam::aws:policy/root-task/S3UnlockBucketPolicy"
        }
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
            "expiration": "Oct 25, 2024, 9:07:11 PM"
    }
}
```

3. Nei CloudTrail log del principale di destinazione, cercate l'ID della chiave di accesso che corrisponde al `accessKeyId` valore dell'evento. `AssumeRoot` Utilizza i valori del campo `eventName` per determinare le attività con privilegi eseguite durante la sessione `AssumeRoot`. È possibile che vengano eseguite più attività con privilegi in una singola sessione. La durata massima della sessione per `AssumeRoot` è 900 secondi (15 minuti).

   Nell'esempio seguente, l'account di gestione o l'amministratore delegato ha eliminato la policy basata sulle risorse per un bucket Amazon S3.

```
{
    "eventVersion": "1.10",
    "userIdentity": {
        "type": "Root",
        "principalId": "222222222222",
        "arn": "arn:aws:iam::222222222222:root",
        "accountId": "222222222222",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "creationDate": "2024-10-25T20:52:11Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-10-25T20:53:47Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "DeleteBucketPolicy",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "requestParameters": {
        "bucketName": "resource-policy-John",
        "Host": "resource-policy-John.s3.amazonaws.com",
        "policy": ""
    },
    "responseElements": null,
    "requestID": "1234567890abcdef0",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "readOnly": false,
    "resources": [
        {
            "accountId": "222222222222",
            "type": "AWS::S3::Bucket",
            "ARN": "arn:aws:s3:::resource-policy-John"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "222222222222",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
```

```
        "clientProvidedHostHeader": "resource-policy-John.s3.amazonaws.com"
    }
}
```

Convalida della conformità per AWS Identity and Access Management

Third-party gli auditor valutano la sicurezza e la conformità di AWS Identity and Access Management (IAM) nell'ambito di diversi programmi di AWS conformità. tra cui SOC, PCI, FedRAMP, ISO e altri.

La nostra nuova esperienza AWS di registrazione non è progettata per carichi di lavoro regolamentati. Se stai utilizzando la nostra nuova esperienza di AWS registrazione, ma desideri utilizzarla AWS per carichi di lavoro regolamentati, puoi iscriverti a AWS (avanzato) o attivare funzionalità avanzate per il tuo ambiente. AWS

Per sapere se un Servizio AWS rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program e scegli il programma di conformità che ti interessa. Per informazioni generali, vedere Programmi di AWS conformità Programmi di

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, vedere Scaricamento dei report in AWS Artifact .

La responsabilità dell'utente in materia di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità dell'utente in materia di conformità durante l'utilizzo Servizi AWS, consulta AWS la Documentazione sulla sicurezza.

Resilienza in AWS Identity and Access Management

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni hanno più zone di disponibilità fisicamente separate e isolate, collegate tramite reti a bassa latenza, ad alto throughput e altamente ridondanti. Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

AWS Identity and Access Management (IAM) e Servizio di token di sicurezza AWS (AWS STS) sono servizi autosufficienti, basati sulla regione, disponibili a livello globale.

IAM è fondamentale. Servizio AWS Ogni operazione eseguita in AWS deve essere autenticata e autorizzata da IAM. IAM verifica ogni richiesta in base alle identità e alle policy archiviate in IAM

per determinare se accettare o negare la richiesta. IAM è stato progettato con un piano di controllo e un piano dati separati in modo che il servizio si autentichi anche in caso di errori imprevisti. Le risorse IAM utilizzate nelle autorizzazioni, come ad esempio ruoli e policy, vengono archiviate nel piano di controllo. I clienti IAM possono modificare la configurazione di queste risorse utilizzando operazioni IAM come `DeletePolicy` e `AttachRolePolicy`. Tali richieste di modifica della configurazione pervengono al piano di controllo. Esiste un unico piano di controllo IAM per tutte le attività commerciali Regioni AWS, che si trova nella regione degli Stati Uniti orientali (Virginia settentrionale). Il sistema IAM propaga quindi le modifiche di configurazione ai piani dati IAM in ogni Regione AWS abilitata. Il piano dati IAM è essenzialmente una replica in sola lettura dei dati di configurazione del piano di controllo IAM. Ciascuno di essi Regione AWS dispone di un'istanza completamente indipendente del piano dati IAM, che esegue l'autenticazione e l'autorizzazione per le richieste provenienti dalla stessa regione. In ogni regione, il piano dati IAM è distribuito su almeno tre zone di disponibilità e ha una capacità sufficiente per tollerare la perdita di una zona di disponibilità senza conseguenze per il cliente. Sia il piano di controllo che il piano dati di IAM sono stati progettati per l'assenza di tempi di inattività pianificati, con tutti gli aggiornamenti software e le operazioni di dimensionamento eseguite in modo impercettibile per i clienti.

Nelle regioni abilitate per impostazione predefinita, le richieste all'endpoint AWS STS globale vengono inviate automaticamente nella stessa regione da cui proviene la richiesta. Nelle regioni opt-in, le richieste all'endpoint AWS STS globale vengono servite da un'unica regione Regione AWS, gli Stati Uniti orientali (Virginia settentrionale). È possibile utilizzare un AWS STS endpoint regionale per ridurre la latenza o fornire ridondanza aggiuntiva per le applicazioni. Per ulteriori informazioni, consulta Manage (Gestione) AWS STS in un Regione AWS.

Alcuni eventi possono interrompere le comunicazioni attraverso la rete. Regioni AWS Tuttavia, anche quando non puoi comunicare con l'endpoint IAM globale, AWS STS puoi comunque autenticare i principi IAM e IAM può autorizzare le tue richieste. I dettagli specifici di un evento che interrompe la comunicazione determineranno la tua capacità di accedere ai servizi. AWS Nella maggior parte delle situazioni, puoi continuare a utilizzare le credenziali IAM nel tuo AWS ambiente. Le seguenti condizioni possono applicarsi a un evento che interrompe la comunicazione.

Chiavi di accesso per gli utenti IAM

> Puoi autenticarti a tempo indeterminato in una regione con le chiavi di accesso per gli utenti IAM a lungo termine. Quando utilizzi AWS Command Line Interface and APIs, puoi fornire chiavi di AWS accesso in modo AWS da verificare la tua identità nelle richieste programmatiche.

> **Important**
>
> Come best practice, consigliamo che i tuoi utenti eseguano l'accesso con le credenziali temporanee al posto delle chiavi di accesso a lungo termine.

Credenziali temporanee

È possibile richiedere nuove credenziali temporanee con l'endpoint di servizio AWS STS regionale per almeno 24 ore. Le seguenti operazioni API generano credenziali temporanee.

- AssumeRole
- AssumeRoleWithWebIdentity
- AssumeRoleWithSAML
- GetFederationToken
- GetSessionToken

Principali e autorizzazioni

- Potresti non essere in grado di aggiungere, modificare o rimuovere i principali o le autorizzazioni in IAM.
- Le tue credenziali potrebbero non riflettere le modifiche alle autorizzazioni che hai applicato in IAM di recente. Per ulteriori informazioni, consulta Le modifiche che apporto non sono sempre immediatamente visibili.

Console di gestione AWS

- Potresti essere in grado di utilizzare un endpoint di accesso regionale per accedere alla Console di gestione AWS come utente IAM. Gli endpoint di accesso regionali hanno il seguente formato URL.

  `https://{Account ID}.signin.aws.amazon.com/console?region={Region}`

  Esempio: /console https://111122223333.signin.aws.amazon.com? region=us-west-2
- Potresti non essere in grado di completare l'autenticazione a più fattori (MFA) Universal 2nd Factor (U2F).

Best practice per la resilienza di IAM

AWS ha integrato la resilienza nelle zone di disponibilità. Regioni AWS Se osservi le seguenti best practice IAM nei sistemi che interagiscono con il tuo ambiente, puoi trarre vantaggio da tale resilienza.

1. Utilizza un endpoint di servizio AWS STS regionale anziché l'endpoint globale predefinito.
2. Verifica la configurazione del tuo ambiente alla ricerca di risorse vitali che creano o modificano abitualmente risorse IAM e prepara una soluzione di fallback che utilizzi le risorse IAM esistenti.

Sicurezza dell'infrastruttura in AWS Identity and Access Management

In quanto servizio gestito, AWS Identity and Access Management è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere a IAM attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

IAM è accessibile a livello di programmazione utilizzando le API HTTPS che consentono di inviare richieste HTTPS direttamente al servizio. L'API Query restituisce informazioni riservate, incluse le credenziali di sicurezza. Pertanto, è necessario utilizzare HTTPS con tutte le richieste API. Quando utilizzi le API HTTPS, devi includere il codice per firmare in modo digitale le richieste utilizzando le tue credenziali.

È possibile richiamare queste operazioni API da qualsiasi posizione di rete, ma IAM non supporta le policy di accesso basate sulle risorse che possono includere limitazioni sull'indirizzo IP di origine. Puoi anche utilizzare le policy IAM per controllare l'accesso da endpoint Amazon Virtual Private

Cloud (Amazon VPC) specifici o specifici. VPCs In effetti, questo isola l'accesso alla rete a una determinata risorsa IAM solo dal VPC specifico all'interno AWS della rete.

Configurazione e analisi delle vulnerabilità in AWS Identity and Access Management

AWS gestisce le attività di sicurezza di base come il sistema operativo guest (OS) e l'applicazione di patch al database, la configurazione del firewall e il disaster recovery. Queste procedure sono state riviste e certificate dalle terze parti appropriate. Per ulteriori dettagli, consulta le seguenti risorse :

- Modello di responsabilità condivisa
- Amazon Web Services: panoramica dei processi di sicurezza (whitepaper)

Le seguenti risorse affrontano anche l'analisi della configurazione e delle vulnerabilità in AWS Identity and Access Management (IAM):

- Convalida della conformità per AWS Identity and Access Management
- Migliori pratiche di sicurezza e casi d'uso in AWS Identity and Access Management

AWS politiche gestite per AWS Identity and Access Management, IAM Access Analyzer e gestore degli accessi agli account

Una policy AWS gestita è una policy autonoma creata e amministrata da. AWS AWS le policy gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere autorizzazioni con privilegi minimi per i tuoi casi d'uso specifici perché sono disponibili per tutti i clienti. AWS Si consiglia pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i propri casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una policy AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui la policy è associata. AWS è più probabile che aggiorni una

policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWS nella Guida per l'utente di IAM.

IAMReadOnlyAccess

Utilizza la policy gestita `IAMReadOnlyAccess` per consentire l'accesso in sola lettura alle risorse IAM. Questa policy concede l'autorizzazione per ottenere ed elencare tutte le risorse IAM. Consente di visualizzare dettagli e i report sulle attività per utenti, gruppi, ruoli, policy, provider di identità e dispositivi MFA. Non include la possibilità di creare o eliminare le risorse o di accedere alle risorse di Sistema di analisi degli accessi IAM. Visualizza la policy per l'elenco completo di servizi e operazioni supportati dalla policy.

IAMUserChangePassword

Utilizza la policy gestita `IAMUserChangePassword` per consentire agli IAM utenti di modificare le loro password.

Configura le Impostazioni dell'account IAM e la Policy sulle password per consentire agli utenti IAM di modificare la password dell'account IAM. Quando consenti questa operazione, IAM collega la policy a ciascun utente.

Per visualizzare la policy JSON, consulta IAMUserChangePassword la AWS Managed Policy Reference Guide.

IAMAccessAnalyzerFullAccess

Utilizza la policy `IAMAccessAnalyzerFullAccess` AWS gestita per consentire agli amministratori di accedere a IAM Access Analyzer.

Raggruppamenti di autorizzazioni

Questa policy è raggruppata in istruzioni in base al set di autorizzazioni fornite.

- Sistema di analisi degli accessi IAM: concede le autorizzazioni amministrative complete a tutte le risorse in Sistema di analisi degli accessi IAM.

- Crea ruolo collegato ai servizi: consente all'amministratore di creare un ruolo collegato ai servizi che consente a Sistema di analisi degli accessi IAM di analizzare le risorse in altri servizi per tuo conto. Questa autorizzazione consente di creare il ruolo collegato ai servizi solo per l'utilizzo da parte di Sistema di analisi degli accessi IAM.
- AWS Organizations: consente agli amministratori di utilizzare Sistema di analisi degli accessi IAM per un'organizzazione in AWS Organizations. Dopo aver abilitato l'accesso affidabile per IAM Access Analyzer in AWS Organizations, i membri dell'account di gestione possono visualizzare i risultati all'interno della propria organizzazione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "access-analyzer:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "access-analyzer.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:DescribeOrganizationalUnit",
        "organizations:ListAccounts",
        "organizations:ListAccountsForParent",
        "organizations:ListAWSServiceAccessForOrganization",
```

```
        "organizations:ListChildren",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListParents",
        "organizations:ListRoots"
      ],
      "Resource": "*"
    }
  ]
}
```

IAMAccessAnalyzerReadOnlyAccess

Utilizza la policy `IAMAccessAnalyzerReadOnlyAccess` AWS gestita per consentire l'accesso in sola lettura a IAM Access Analyzer.

Per consentire anche l'accesso in sola lettura a IAM Access Analyzer for AWS Organizations, crea una policy gestita dal cliente che consenta le azioni Descrivi ed Elenca della policy gestita. IAMAccessAnalyzerFullAccess AWS

Service-level autorizzazioni

Questa policy fornisce accesso in sola lettura a Sistema di analisi degli accessi IAM. In questa policy non sono incluse altre autorizzazioni di servizio.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMAccessAnalyzerReadOnlyAccess",
      "Effect": "Allow",
      "Action": [
        "access-analyzer:CheckAccessNotGranted",
        "access-analyzer:CheckNoNewAccess",
        "access-analyzer:Get*",
        "access-analyzer:List*",
        "access-analyzer:ValidatePolicy"
      ],
```

```
      "Resource": "*"
    }
  ]
}
```

AccessAnalyzerServiceRolePolicy

Non puoi collegarti AccessAnalyzerServiceRolePolicy alle tue entità IAM. Questa policy è collegata a un ruolo collegato ai servizi che consente a Sistema di analisi degli accessi IAM di eseguire operazioni per tuo conto. Per ulteriori informazioni, consulta Utilizzo dei ruoli collegati ai servizi per AWS Identity and Access Management e Access Analyzer.

Raggruppamenti di autorizzazioni

Questa policy consente l'accesso a Sistema di analisi degli accessi IAM per analizzare i metadati delle risorse da più Servizi AWS.

- Amazon DynamoDB: concede le autorizzazioni per visualizzare flussi e tabelle DynamoDB.
- Amazon Elastic Compute Cloud: consente le autorizzazioni per descrivere indirizzi IP, istantanee e VPC.
- Amazon Elastic Container Registry: concede le autorizzazioni per descrivere i repository di immagini, recuperare le impostazioni degli account e recuperare le policy dei repository e dei registri.
- Amazon Elastic File System: consente le autorizzazioni per visualizzare la descrizione di un file system Amazon EFS e visualizzare la policy a livello di risorse per un file system Amazon EFS.
- AWS Identity and Access Management— Consente le autorizzazioni per recuperare informazioni su un ruolo specificato ed elencare i ruoli IAM con un prefisso di percorso specificato. Concede le autorizzazioni per recuperare informazioni su utenti, gruppi IAM, profili di accesso, chiavi di accesso e dati dell'ultimo accesso al servizio.
- AWS Key Management Service— Consente le autorizzazioni per visualizzare informazioni dettagliate su una chiave KMS e sulle relative policy e concessioni chiave.
- AWS Lambda— Consente le autorizzazioni per visualizzare informazioni su alias, funzioni, livelli e alias Lambda.
- AWS Organizations— Consente AWS Organizations e consente la creazione di un analizzatore all'interno dell' AWS organizzazione come zona di fiducia.

- Amazon Relational Database Service: consente le autorizzazioni per visualizzare informazioni dettagliate sugli snapshot del database Amazon RDS e sugli snapshot dei cluster di database Amazon RDS.
- Amazon Simple Storage Service: consente le autorizzazioni per visualizzare informazioni dettagliate sui punti di accesso Amazon S3, sui bucket, sui punti di accesso dei bucket della directory Amazon S3 e sui bucket della directory.
- AWS Secrets Manager— Consente le autorizzazioni per visualizzare informazioni dettagliate sui segreti e sulle policy delle risorse allegate ai segreti.
- Amazon Simple Notification Service: consente le autorizzazioni per visualizzare informazioni dettagliate su un argomento.
- Amazon Simple Queue Service: consente le autorizzazioni per visualizzare informazioni dettagliate sulle code specificate.

Per visualizzare la policy JSON, consulta la AWS Managed Policy AccessAnalyzerServiceRolePolicy Reference Guide.

IAMAuditRootUserCredentials

Utilizza la policy `IAMAuditRootUserCredentials` AWS gestita per definire l'ambito delle autorizzazioni quando esegui un'attività privilegiata su un account AWS Organizations membro per verificare lo stato delle credenziali dell'utente root di un account membro. È possibile elencare oppure ottenere informazioni sulle credenziali dei singoli utenti root, ad esempio:

- Se è presente una password per l'utente root
- Se l'utente root ha una chiave di accesso e quando è stata utilizzata l'ultima volta
- Se l'utente root ha certificati di firma associati
- I dispositivi MFA associati all'utente root
- L'elenco dello stato delle credenziali consolidate dell'utente root

Non è possibile collegare `IAMAuditRootUserCredentials`alle entità IAM. Questa politica è associata AssumeRoot all'esecuzione di attività privilegiate su un account membro dell'organizzazione. Per ulteriori informazioni, consulta Gestire centralmente l'accesso root per gli account membri.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "DenyAllOtherActionsOnAnyResource",
         "NotAction": [
            "iam:ListAccessKeys",
            "iam:ListSigningCertificates",
            "iam:GetLoginProfile",
            "iam:ListMFADevices",
            "iam:GetAccountSummary",
            "iam:GetUser",
            "iam:GetAccessKeyLastUsed"
         ],
         "Effect": "Deny",
         "Resource": "*"
      },
      {
         "Sid": "DenyAuditingCredentialsOnNonRootUserResource",
         "Action": [
            "iam:ListAccessKeys",
            "iam:ListSigningCertificates",
            "iam:GetLoginProfile",
            "iam:ListMFADevices" ,
            "iam:GetUser",
            "iam:GetAccessKeyLastUsed"
         ],
         "Effect": "Deny",
         "NotResource": "arn:aws:iam::*:root"
      }
   ]
}
```

Raggruppamenti di autorizzazioni

Questa policy è raggruppata in istruzioni in base al set di autorizzazioni fornite.

- DenyAllOtherActionsOnAnyResource— Nega l'accesso alle credenziali per tutte le risorse.

- DenyAuditingCredentialsOnNonRootUserResource— Nega l'accesso alle credenziali per tutte le risorse utente non root.

IAMCreateRootUserPassword

Utilizza la policy `IAMCreateRootUserPassword` AWS gestita per limitare le autorizzazioni quando esegui un'attività privilegiata su un account AWS Organizations membro per consentire il recupero della password per un account membro senza credenziali utente root.

Non è possibile collegare `IAMCreateRootUserPassword`dalle entità IAM. Questa policy è associata AssumeRoot all'esecuzione di attività privilegiate su un account membro dell'organizzazione. Per ulteriori informazioni, consulta Gestire centralmente l'accesso root per gli account membri.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "DenyAllOtherActionsOnAnyResource",
         "NotAction": [
            "iam:CreateLoginProfile",
            "iam:GetLoginProfile"
         ],
         "Effect": "Deny",
         "Resource": "*"
      },
      {
         "Sid": "DenyCreatingPasswordOnNonRootUserResource",
         "Action": [
            "iam:CreateLoginProfile",
            "iam:GetLoginProfile"
         ],
         "Effect": "Deny",
         "NotResource": "arn:aws:iam::*:root"
      }
   ]
}
```

Raggruppamenti di autorizzazioni

Questa policy è raggruppata in istruzioni in base al set di autorizzazioni fornite.

- DenyAllOtherActionsOnAnyResource— Nega l'accesso per ottenere o creare una password per tutte le risorse.
- DenyCreatingPasswordOnNonRootUserResource— Nega l'accesso per ottenere o creare una password per tutte le risorse utente non root.

IAMDeleteRootUserCredentials

Utilizza la policy `IAMDeleteRootUserCredentials` AWS gestita per limitare le autorizzazioni quando esegui un'attività privilegiata su un account AWS Organizations membro per rimuovere le credenziali dell'utente root, tra cui password, chiavi di accesso, certificati di firma e disattivazione dell'MFA. Le autorizzazioni con privilegi sono necessarie per questa azione con privilegi in modo da poter visualizzare le informazioni sulle ultime credenziali utilizzate, verificare le ultime informazioni utilizzate per l'utente root dell'account membro ed elencare le autorizzazioni per tutte le credenziali dell'utente root da eliminare.

Non è possibile collegare `IAMDeleteRootUserCredentials`alle entità IAM. Questa politica è associata AssumeRoot all'esecuzione di attività privilegiate su un account membro dell'organizzazione. Per ulteriori informazioni, consulta Gestire centralmente l'accesso root per gli account membri.

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "DenyAllOtherActionsOnAnyResource",
   "Effect": "Deny",
   "NotAction": [
    "iam:DeleteAccessKey",
    "iam:DeleteSigningCertificate",
    "iam:DeleteLoginProfile",
    "iam:DeactivateMFADevice",
    "iam:ListAccessKeys",
    "iam:ListSigningCertificates",
```

```
    "iam:GetLoginProfile",
    "iam:ListMFADevices",
    "iam:GetUser",
    "iam:GetAccessKeyLastUsed"
   ],

   "Resource": "*"
  },
  {
   "Sid": "DenyDeletingRootUserCredentialsOnNonRootUserResource",
   "Effect": "Deny",
   "Action": [
    "iam:DeleteAccessKey",
    "iam:DeleteSigningCertificate",
    "iam:DeleteLoginProfile",
    "iam:DeactivateMFADevice",
    "iam:ListAccessKeys",
    "iam:ListSigningCertificates",
    "iam:GetLoginProfile",
    "iam:ListMFADevices",
    "iam:GetUser",
    "iam:GetAccessKeyLastUsed"
   ],

   "NotResource": "arn:aws:iam::*:root"
  }
 ]
}
```

Raggruppamenti di autorizzazioni

Questa policy è raggruppata in istruzioni in base al set di autorizzazioni fornite.

- DenyAllOtherActionsOnAnyResource— Nega l'accesso per ottenere o eliminare le credenziali per tutte le risorse.
- DenyDeletingRootUserCredentialsOnNonRootUserResource— Nega l'accesso per ottenere o eliminare le credenziali per tutte le risorse utente non root.

S3UnlockBucketPolicy

Utilizza la policy `S3UnlockBucketPolicy` AWS gestita per limitare le autorizzazioni quando esegui un'attività privilegiata su un account AWS Organizations membro per rimuovere una policy del bucket non configurata correttamente che impedisce a tutti i principali di accedere a un bucket Amazon S3.

Non è possibile collegare `S3UnlockBucketPolicy`alle entità IAM. Questa policy è associata all'esecuzione di attività privilegiate su AssumeRoot un account membro della tua organizzazione. Per ulteriori informazioni, consulta Gestire centralmente l'accesso root per gli account membri.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "DenyAllOtherActionsOnAnyResource",
         "NotAction": [
            "s3:DeleteBucketPolicy",
            "s3:PutBucketPolicy",
            "s3:GetBucketPolicy",
            "s3:ListAllMyBuckets"
         ],
         "Effect": "Deny",
         "Resource": "*"
      },
      {
         "Sid": "DenyManagingBucketPolicyForNonRootCallers",
         "Action": [
            "s3:DeleteBucketPolicy",
            "s3:PutBucketPolicy",
            "s3:GetBucketPolicy",
            "s3:ListAllMyBuckets"
         ],
         "Effect": "Deny",
         "Resource": "*",
         "Condition" : {
            "ArnNotLike" : {
               "aws:PrincipalArn" : "arn:aws:iam::*:root"
            }
         }
```

```
        }
    ]
}
```

Raggruppamenti di autorizzazioni

Questa policy è raggruppata in istruzioni in base al set di autorizzazioni fornite.

- DenyAllOtherActionsOnAnyResource— Nega l'accesso alle politiche dei bucket per tutte le risorse.
- DenyManagingBucketPolicyForNonRootCallers— Nega l'accesso alle policy dei bucket per tutte le risorse utente non root.

SQSUnlockQueuePolicy

Utilizza la policy `SQSUnlockQueuePolicy` AWS gestita per limitare le autorizzazioni quando esegui un'attività privilegiata su un account AWS Organizations membro per eliminare una policy basata sulle risorse di Amazon Simple Queue Service che impedisce a tutti i principali di accedere a una coda Amazon SQS.

Non è possibile collegare `SQSUnlockQueuePolicy`alle entità IAM. Questa policy è allegata all'esecuzione di attività privilegiate su un account membro della tua AssumeRoot organizzazione. Per ulteriori informazioni, consulta Gestire centralmente l'accesso root per gli account membri.

Raggruppamenti di autorizzazioni

Questa policy è raggruppata in istruzioni in base al set di autorizzazioni fornite.

- DenyAllOtherActionsOnAnyResource— Nega l'accesso alle azioni di Amazon SQS per tutte le risorse.
- DenyGettingQueueAttributesOnNonOwnQueue— Nega l'accesso agli attributi delle code di Amazon SQS per le code di proprietà di un altro account.
- DenyActionsForNonRootUser— Nega l'accesso alle azioni di Amazon SQS per tutte le risorse utente non root.

AccountAccessManagerServiceRolePolicy

Non è possibile collegare `AccountAccessManagerServiceRolePolicy`alle entità IAM. Questa policy è associata a un ruolo collegato al servizio che consente all'account access manager di eseguire azioni per tuo conto. Per ulteriori informazioni, consulta the section called "Sicurezza".

Service-level autorizzazioni

Questa politica consente all'account access manager di recuperare informazioni sugli account dell'organizzazione per mantenere un'iscrizione accurata agli account per la gestione degli accessi. L'Account Access Manager utilizza queste autorizzazioni per rilevare quando gli account lasciano l'organizzazione, verificare se l'accesso ai servizi attendibili è abilitato e confermare lo stato dell'account prima di gestire le autorizzazioni.

- AWS Organizations— Consente le autorizzazioni per elencare gli account dell'organizzazione, verificare se l'accesso ai servizi attendibili è abilitato e descrivere lo stato dell'account.

Per visualizzare la policy JSON, consulta AccountAccessManagerServiceRolePolicy la AWS Managed Policy Reference Guide.

Aggiornamenti di IAM, IAM Access Analyzer e account access manager a AWS policy gestite

Visualizza i dettagli sugli aggiornamenti di IAM e sulle policy AWS gestite da quando il servizio ha iniziato a tracciare queste modifiche. Per ricevere gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivi il feed RSS su IAM e nelle pagine della cronologia di Sistema di analisi degli accessi IAM.

Modifica	Descrizione	Data
AccountAccessManagerServiceRolePolicy — È stata aggiunta una politica gestita	L'Account Access Manager ha aggiunto una policy relativa ai ruoli collegati al servizio che consente al servizio di recuperare informazioni sugli account dell'organizzazione.	31 luglio 2026

Modifica	Descrizione	Data
IAMUserChangePassword: autorizzazioni aggiunte	IAM ha aggiunto autorizzazioni a `IAMUserChangePassword` per autorizzare gli utenti specificati all'interno di un percorso.	28 maggio 2025
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Strumento di analisi degli accessi IAM ha aggiunto `iam:GetAccountAuthorizationDetails` alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	12 maggio 2025
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Strumento di analisi degli accessi IAM ha aggiunto il supporto per i punti di accesso dei bucket della directory Amazon S3 alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	31 marzo 2025
IAMDeleteRootUserCredentials: autorizzazioni rimosse	IAM ha rimosso l'autorizzazione `iam:DeleteVirtualMFADevice` dalla policy gestita.	7 gennaio 2025

Modifica	Descrizione	Data
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM supporta l'autorizzazione per recuperare informazioni sulle impostazioni dell'account Amazon ECR e sulle policy di registro alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	10 dicembre 2024
IAMAuditRootUserCredentials — È stata aggiunta una politica gestita	IAM ha aggiunto le policy gestite per la gestione centralizzata dell'accesso root per gli account membri per definire le attività con privilegi che è possibile eseguire sugli account membri AWS Organizations .	14 novembre 2024
IAMCreateRootUserPassword— È stata aggiunta una politica gestita	IAM ha aggiunto le policy gestite per la gestione centralizzata dell'accesso root per gli account membri per definire le attività con privilegi che è possibile eseguire sugli account membri AWS Organizations .	14 novembre 2024

Modifica	Descrizione	Data
IAMDeleteRootUserCredentials— È stata aggiunta una politica gestita	IAM ha aggiunto le policy gestite per la gestione centralizzata dell'accesso root per gli account membri per definire le attività con privilegi che è possibile eseguire sugli account membri AWS Organizations .	14 novembre 2024
S3UnlockBucketPolicy— È stata aggiunta una politica gestita	IAM ha aggiunto le policy gestite per la gestione centralizzata dell'accesso root per gli account membri per definire le attività con privilegi che è possibile eseguire sugli account membri AWS Organizations .	14 novembre 2024
SQSUnlockQueuePolicy— È stata aggiunta una politica gestita	IAM ha aggiunto le policy gestite per la gestione centralizzata dell'accesso root per gli account membri per definire le attività con privilegi che è possibile eseguire sugli account membri AWS Organizations .	14 novembre 2024

Modifica	Descrizione	Data
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto il supporto per l'autorizzazione per recuperare informazioni sui tag di ruoli e utenti IAM alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	29 ottobre 2024
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto il supporto per l'autorizzazione per recuperare informazioni sulle policy di ruoli e utenti IAM alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	30 maggio 2024
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto il supporto per l'autorizzazione per recuperare lo stato corrente dell'accesso pubblico al blocco per gli snapshot di Amazon EC2 alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	23 gennaio 2024

Modifica	Descrizione	Data
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto il supporto per i flussi e le tabelle DynamoDB alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	11 gennaio 2024
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto il supporto per i bucket di directory Amazon S3 alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	1 dicembre 2023
IAMAccessAnalyzerReadOnlyAccess: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto le autorizzazioni per consentirti di verificare se gli aggiornamenti alle tue policy garantiscono un accesso aggiuntivo. Questa autorizzazione è richiesta da Sistema di analisi degli accessi IAM per eseguire i controlli delle policy sulla policy.	26 novembre 2023

Modifica	Descrizione	Data
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto le operazioni IAM alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` per supportare le seguenti operazioni: • Elencare le entità per una policy • Generare dettagli sull'ultimo accesso al servizio • Elencare le informazioni sulla chiave di accesso	26 novembre 2023
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto il supporto per i seguenti tipi di risorse alle autorizzazioni a livello di servizio di`AccessAnalyzerServiceRolePolicy` : • Snapshot del volume Amazon EBS • Repository di Amazon ECR • File system di Amazon EFS • Snapshot del database Amazon RDS • Snapshot del cluster database Amazon RDS • Argomenti di Amazon SNS	25 ottobre 2022

Modifica	Descrizione	Data
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto l'operazione `lambda:GetFunctionUrlConfig` alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	6 aprile 2022
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto nuove operazioni di Amazon S3 per analizzare i metadati associati ai punti di accesso multi-regione.	2 settembre 2021
IAMAccessAnalyzerReadOnlyAccess: autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto una nuova operazione per concedere le autorizzazioni `ValidatePolicy` per consentire all'utente di utilizzare i controlli delle policy per la convalida. Questa autorizzazione è richiesta da Sistema di analisi degli accessi AWS IAM per eseguire i controlli delle policy sulla policy.	16 marzo 2021
Sistema di analisi degli accessi IAM ha iniziato il monitoraggio delle modifiche	IAM Access Analyzer ha iniziato a tracciare le modifiche per le sue policy AWS gestite.	1° marzo 2021

Funzioni di sicurezza al di fuori di IAM

Utilizzi IAM per controllare l'accesso alle attività eseguite utilizzando gli Console di gestione AWSstrumenti della AWS riga di comando o le operazioni dell'API di servizio utilizzando AWS SDKs. In alcuni prodotti AWS sono disponibili altre modalità per la protezione delle risorse. In via esemplificativa, di seguito sono elencati alcuni esempi.

Amazon EC2

In Amazon Elastic Compute Cloud si accede a un'istanza con una coppia di chiavi (per le istanze di Linux) o utilizzando un nome utente e una password (per le istanze di Microsoft Windows).

Per ulteriori informazioni, consulta la seguente documentazione:

- Guida introduttiva alle istanze Amazon EC2 Linux nella Amazon EC2 User Guide
- Guida introduttiva alle istanze di Amazon EC2 Windows nella Amazon EC2 User Guide

Amazon RDS

Per accedere al motore di database in Amazon Relational Database Service è necessario utilizzare nome utente e password associati al database.

Per ulteriori informazioni, consulta Nozioni di base su Amazon RDS nella Guida per l'utente di Amazon RDS.

Amazon EC2 e Amazon RDS

In Amazon EC2 e Amazon RDS si utilizzano gruppi di sicurezza per controllare il traffico verso un'istanza o un database.

Per ulteriori informazioni, consulta la seguente documentazione:

- Gruppi EC2 di sicurezza Amazon per istanze Linux nella Amazon EC2 User Guide
- Gruppi EC2 di sicurezza Amazon per istanze Windows nella Amazon EC2 User Guide
- Gruppi di sicurezza di Amazon RDS nella Guida per l'utente di Amazon RDS

WorkSpaces

In Amazon WorkSpaces, gli utenti accedono a un desktop con un nome utente e una password.

Per ulteriori informazioni, consulta la sezione Getting Started with WorkSpaces nella Amazon WorkSpaces Administration Guide.

Questi metodi di controllo degli accessi non sono parte di IAM. IAM ti consente di controllare il modo in cui questi AWS prodotti vengono amministrati: creazione o chiusura di un'istanza EC2 Amazon, configurazione di WorkSpaces nuovi desktop e così via. Ovvero, IAM consente di controllare i processi eseguiti tramite l'esecuzione di richieste ad Amazon Web Services nonché l'accesso alla Console di gestione AWS. Tuttavia, IAM non ti aiuta a gestire la sicurezza per attività come l'accesso a un sistema operativo (Amazon EC2), un database (Amazon RDS) o un desktop (Amazon WorkSpaces).

Quando lavori con un AWS prodotto specifico, assicurati di leggere la documentazione per conoscere le opzioni di sicurezza per tutte le risorse che appartengono a quel prodotto.

Utilizzo AWS Identity and Access Management Access Analyzer

AWS Identity and Access Management Access Analyzer offre le seguenti funzionalità:

- Gli analizzatori degli accessi esterni di Sistema di analisi degli accessi IAM consente di identificare le risorse nell'organizzazione e negli account che sono condivise con un'entità esterna.
- Gli analizzatori degli accessi interni di Strumento di analisi degli accessi IAM aiutano a identificare l'accesso interno alle risorse selezionate nella tua organizzazione e nei tuoi account.
- Gli analizzatori di accessi inutilizzati di Sistema di analisi degli accessi IAM aiutano a identificare gli accessi inutilizzati nell'organizzazione e negli account.
- IAM Access Analyzer convalida le politiche IAM rispetto alla grammatica delle politiche e AWS alle best practice.
- I controlli delle policy personalizzati di Sistema di analisi degli accessi IAM a convalidare le policy IAM rispetto agli standard di sicurezza specificati.
- IAM Access Analyzer genera policy IAM basate sull'attività di accesso nei tuoi log. AWS CloudTrail

Identificazione delle risorse condivise con un'entità esterna

Sistema di analisi degli accessi IAM consente di identificare le risorse nell'organizzazione e negli account, ad esempio bucket Amazon S3 o ruoli IAM, condivise con un'entità esterna. In questo modo puoi identificare l'accesso non intenzionale alle risorse e ai dati, che rappresenta un rischio per la sicurezza. IAM Access Analyzer identifica le risorse condivise con i principali esterni utilizzando un ragionamento basato sulla logica per analizzare le politiche basate sulle risorse nel tuo ambiente. AWS Per ogni istanza di una risorsa condivisa al di fuori dell'account, Sistema di analisi degli accessi IAM genera un risultato. I risultati comprendono informazioni sull'accesso e sull'entità esterna a cui è concesso. Puoi rivedere i risultati per determinare se l'accesso è intenzionale e sicuro o se è involontario e rappresenta un rischio per la sicurezza. Oltre a facilitare l'identificazione delle risorse condivise con un'entità esterna, puoi utilizzare i risultati di Sistema di analisi degli accessi IAM per visualizzare in anteprima il modo in cui le policy influiscono sull'accesso multi-account e pubblico sulla risorsa prima di implementare le autorizzazioni delle risorse. I risultati sono organizzati in una dashboard visiva riassuntiva. La dashboard evidenzia la suddivisione tra risultati relativi all'accesso multi-account e pubblico e differenzia i risultati per tipo di risorsa. Per ulteriori informazioni sulle

dashboard, consulta Visualizzare il pannello di controllo dei risultati di Sistema di analisi degli accessi IAM.

> **Note**
>
> Un'entità esterna può essere un altro AWS account, un utente root, un utente o ruolo IAM, un utente federato, un utente anonimo o un'altra entità che puoi usare per creare un filtro. For more information, see Elementi della policy JSON di AWS : entità principale.

Per istruzioni su come iniziare, consultaCrea un sistema di analisi degli accessi esterni per Sistema di analisi degli accessi IAM. Dopo aver creato un analizzatore, puoi Gestire un sistema di analisi degli accessi esterni per Sistema di analisi degli accessi IAM farlo.

Quando abiliti Sistema di analisi degli accessi IAM, crei un analizzatore per l'intera organizzazione o per il tuo account. L'organizzazione o l'account scelto è noto come zona di attendibilità per l'analizzatore. L'analizzatore monitora tutte le risorse supportate all'interno della zona di attendibilità. È considerato attendibile qualsiasi accesso alle risorse da parte delle entità principali che si trovano all'interno della zona di attendibilità. Una volta abilitato, Sistema di analisi degli accessi IAM analizza le policy applicate a tutte le risorse supportate nella zona di attendibilità. Dopo la prima analisi, Sistema di analisi degli accessi IAM analizza queste policy periodicamente. Se aggiungi una nuova policy o ne modifichi una esistente, Strumento di analisi degli accessi IAM analizza la policy nuova o aggiornata entro circa 30 minuti.

Durante l'analisi delle policy, se Sistema di analisi degli accessi IAM ne identifica una che concede l'accesso a un principale esterno che non rientra nella zona di attendibilità, viene generato un risultato. Ogni risultato include i dettagli sulla risorsa, sull'entità esterna che ha accesso e sulle autorizzazioni concesse in modo da poter intraprendere le azioni appropriate. Puoi visualizzare i dettagli inclusi nel risultato per determinare se l'accesso alla risorsa è intenzionale o un potenziale rischio da risolvere. Quando aggiungi una policy a una risorsa o aggiorni una policy esistente, per prima cosa Sistema di analisi degli accessi IAM la analizza. Sistema di analisi degli accessi IAM inoltre analizza periodicamente tutte le policy basate sulle risorse.

In rare occasioni e in determinate condizioni, Sistema di analisi degli accessi IAM non riceve alcuna notifica relativamente a una policy aggiunta o aggiornata, il che può causare dei ritardi nei risultati generati. Sistema di analisi degli accessi IAM può richiedere fino a 6 ore per generare o risolvere i risultati se crei o elimini un punto di accesso multi-regione associato a un bucket Amazon S3 o se aggiorni la policy per il punto di accesso multi-regione. Inoltre, se si verifica un problema di consegna

relativo alla consegna dei AWS CloudTrail log o alle modifiche alle restrizioni RCP (Resource Control Policy), la modifica della policy non attiva una nuova scansione della risorsa riportata nel risultato. In questo caso, Sistema di analisi degli accessi IAM analizza la policy nuova o aggiornata durante la scansione periodica successiva, che avviene entro 24 ore. Se desideri confermare che una modifica apportata a una policy risolve un problema di accesso segnalato in un esito, puoi eseguire nuovamente la scansione della risorsa segnalata in un risultato utilizzando il collegamento Nuova scansione nella pagina dei dettagli Risultati o utilizzando l'operazione `StartResourceScan` dell'API dello Strumento di analisi degli accessi IAM. Per ulteriori informazioni, consulta Risolvere i risultati di Sistema di analisi degli accessi IAM.

> **Important**
>
> Per l'accesso esterno, Strumento di analisi degli accessi IAM analizza solo le policy applicate alle risorse nella stessa Regione AWS in cui è abilitato. Per monitorare tutte le risorse nel tuo AWS ambiente, devi creare un analizzatore di accesso esterno per abilitare IAM Access Analyzer in ogni regione in cui stai utilizzando le risorse supportate. AWS
> Per gli accessi inutilizzati, i risultati relativi all'analizzatore non cambiano in base alla regione. Non è necessario creare un analizzatore degli accessi non utilizzati in ogni Regione in cui disponi di risorse.

Strumento di analisi degli accessi IAM analizza i seguenti tipi di risorse per l'accesso esterno:

- Bucket Amazon Simple Storage Service
- Bucket di directory di Amazon Simple Storage Service
- AWS Identity and Access Management ruoli
- AWS Key Management Service chiavi
- AWS Lambda funzioni e livelli
- Code Amazon Simple Queue Service
- AWS Secrets Manager segreti
- Argomenti su Amazon Simple Notification Service
- Volumi e snapshot di Amazon Elastic Block Store
- Amazon Relational Database Service
- Snapshot di cluster di database di Amazon Relational Database Service
- Repository di Amazon Elastic Container Registry

- File system di Amazon Elastic File System
- Flussi Amazon DynamoDB
- Tabelle Amazon DynamoDB

Identificazione dell'accesso interno alle risorse aziendali critiche

Per le risorse aziendali critiche selezionate, Strumento di analisi degli accessi IAM aiuta a identificare quali principali all'interno dell'organizzazione o dell'account hanno accesso. Questa analisi supporta l'implementazione del principio del privilegio minimo garantendo che le risorse specificate siano accessibili solo ai principali designati all'interno dell'organizzazione.

Per istruzioni su come iniziare, consulta. Creazione di un analizzatore degli accessi interni di Sistema di analisi degli accessi IAM Dopo aver creato un analizzatore, puoi Gestione di un analizzatore degli accessi interni di Sistema di analisi degli accessi IAM farlo.

L'analisi degli accessi interni consente di:

- Determinare quali utenti o ruoli IAM all'interno dell'account o dell'organizzazione possono accedere alle risorse specificate
- Comprendere i percorsi di accesso tra i principali e le risorse all'interno del tuo ambiente AWS
- Verificare che i controlli di accesso funzionino come previsto
- Rivedere gli esiti per determinare se l'accesso è intenzionale e sicuro o se è involontario e rappresenta un rischio per la sicurezza
- Identificare e correggere gli accessi non intenzionali all'interno dell'organizzazione

Strumento di analisi degli accessi IAM analizza i seguenti tipi di risorse per l'accesso interno:

- Bucket Amazon Simple Storage Service
- Bucket di directory di Amazon Simple Storage Service
- Amazon Relational Database Service
- Snapshot di cluster di database di Amazon Relational Database Service
- Flussi Amazon DynamoDB
- Tabelle Amazon DynamoDB

Identificazione dell'accesso inutilizzato concesso a utenti e ruoli IAM

IAM Access Analyzer ti aiuta a identificare ed esaminare gli accessi non utilizzati nella tua AWS organizzazione e nei tuoi account. Sistema di analisi degli accessi IAM monitora continuamente tutti i ruoli e gli utenti IAM dell'organizzazione e degli account AWS e genera risultati per gli accessi inutilizzati. I risultati evidenziano ruoli inutilizzati, chiavi di accesso inutilizzate per gli utenti IAM e password inutilizzate per gli utenti IAM. Per i ruoli e gli utenti IAM attivi, i risultati forniscono visibilità su servizi e operazioni inutilizzati.

Per istruzioni su come creare un analizzatore di accessi inutilizzato e gestirne i risultati, consulta. Gestire un sistema di analisi degli accessi inutilizzati per Sistema di analisi degli accessi IAM

IAM Access Analyzer esamina le informazioni di ultimo accesso per tutti i ruoli AWS dell'organizzazione e gli account per aiutarti a identificare gli accessi non utilizzati. Le ultime informazioni a cui si è effettuato l'accesso per le operazioni IAM aiutano a identificare le azioni inutilizzate per i ruoli all'interno degli Account AWS. Per ulteriori informazioni, consulta Perfeziona le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso.

I risultati degli analizzatori degli accessi esterni, interni e non utilizzati sono organizzati in un pannello di controllo con un riepilogo visivo. La dashboard evidenzia le AWS risorse Account AWS che contengono il maggior numero di risultati e fornisce una suddivisione dei risultati per tipo. Per ulteriori informazioni sulle pagine del pannello di controllo, consulta Visualizzare il pannello di controllo dei risultati di Sistema di analisi degli accessi IAM.

Convalida delle politiche rispetto a AWS best practice

È possibile convalidare le policy in rapporto alla sintassi della policy IAM e alle best practice AWS utilizzando i controlli delle policy di base forniti dalla convalida delle policy di Sistema di analisi degli accessi IAM. Puoi creare o modificare una policy utilizzando l'editor di policy AWS CLI, AWS API o JSON nella console IAM. È possibile visualizzare i risultati del controllo della convalida delle policy che includono avvisi di sicurezza, errori, avvisi generali e suggerimenti per la policy. Questi risultati forniscono consigli pratici che ti aiutano a creare policy funzionali e conformi alle best practice. AWS Per ulteriori informazioni sulla convalida delle policy tramite l'apposito procedimento, consulta Convalidare le policy con Sistema di analisi degli accessi IAM. Per un elenco dei controlli delle policy eseguiti da IAM Access Analyzer, consulta. Controlli per la convalida delle policy

Convalida delle policy rispetto agli standard di sicurezza specificati

È possibile convalidare le policy rispetto agli standard di sicurezza specificati utilizzando i controlli delle policy personalizzati di Sistema di analisi degli accessi IAM. Puoi creare o modificare una policy utilizzando l'editor di policy AWS CLI, AWS API o JSON nella console IAM. Tramite la console, puoi verificare se la policy aggiornata concede un nuovo accesso rispetto alla versione esistente. AWS CLI Tramite un' AWS API, puoi anche verificare che azioni IAM specifiche che ritieni critiche non siano consentite da una policy. Questi controlli evidenziano un'istruzione di policy che concede nuovi accessi. È possibile aggiornare l'istruzione di policy ed eseguire nuovamente i controlli finché la policy non sarà conforme allo standard di sicurezza. Per ulteriori informazioni sulla convalida delle policy tramite i controlli delle policy personalizzati, consulta Convalidare le policy utilizzando i controlli delle policy personalizzate di Sistema di analisi degli accessi IAM.

Generazione di policy

IAM Access Analyzer analizza AWS CloudTrail i tuoi log per identificare le azioni e i servizi che sono stati utilizzati da un'entità IAM (utente o ruolo) entro l'intervallo di date specificato. Viene quindi generata una policy IAM basata su tale attività di accesso. È possibile utilizzare la policy generata per perfezionare le autorizzazioni di un'entità collegandola a un utente o ruolo IAM. Per ulteriori informazioni sulla generazione di policy tramite Sistema di analisi degli accessi IAM, consulta Generazione di policy per Sistema di analisi degli accessi IAM.

Prezzi per Sistema di analisi degli accessi IAM

Sistema di analisi degli accessi IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese da ogni analizzatore.

- Verrà addebitato il rispettivo costo per ogni analizzatore degli accessi creato.
- La creazione di analizzatori degli accessi inutilizzati in più regioni comporterà un addebito per ogni analizzatore.
- Service-linked i ruoli non vengono analizzati per individuare le attività di accesso non utilizzate e non sono inclusi nel numero totale di ruoli IAM analizzati.

Strumento di analisi degli accessi IAM addebita l'analisi degli accessi interni in base al numero di risorse monitorate al mese per analizzatore degli accessi interni.

Sistema di analisi degli accessi IAM addebita i costi per i controlli delle policy personalizzati in base al numero di richieste API ricevute per verificare la presenza di nuovi accessi.

Per un elenco completo dei prezzi e degli addebiti per IAM Access Analyzer, consulta i prezzi di IAM Access Analyzer.

Per vedere la tua fattura, vai sul Pannello di controllo di gestione dei costi e della fatturazione nella console Gestione costi e fatturazione AWS. La fattura contiene collegamenti per passare ai report di utilizzo, che consentono di visualizzare i dettagli della fattura. Per saperne di più sulla Account AWS fatturazione, consulta la Guida per l'utente AWS Billing

Se hai domande sulla AWS fatturazione, sugli account e sugli eventi, contatta. Supporto

Risultati di Sistema di analisi degli accessi IAM

IAM Access Analyzer genera risultati per l'accesso esterno, l'accesso interno e l'accesso non utilizzato nella tua organizzazione Account AWS .

Per gli accessi esterni, Sistema di analisi degli accessi IAM genera un risultato per ogni istanza di una policy basata sulle risorse che concede l'accesso a una risorsa nella zona di attendibilità a un principale esterno a tale zona. Quando crei un analizzatore di accessi esterno, scegli un'organizzazione o Account AWS un'analisi. Qualsiasi entità principale nell'organizzazione o nell'account scelto per l'analizzatore viene considerata attendibile. Poiché le entità principali nella stessa organizzazione o account sono attendibili, le risorse e le entità principali all'interno dell'organizzazione o dell'account rappresentano la zona di attendibilità per l'analizzatore. Qualsiasi condivisione all'interno della zona di attendibilità è considerata sicura, quindi Sistema di analisi degli accessi IAM non genera alcun risultato. Ad esempio, se si seleziona un'organizzazione come zona di attendibilità per un analizzatore, tutte le risorse e le entità principali dell'organizzazione si trovano all'interno della zona di attendibilità. Se concedi le autorizzazioni per un bucket Amazon S3 in uno degli account membri dell'organizzazione a un principale in un altro account membro dell'organizzazione, Sistema di analisi degli accessi IAM non genera un risultato. Invece, se concedi l'autorizzazione a un principale in un account che non è membro dell'organizzazione, Sistema di analisi degli accessi IAM genera un risultato.

Per gli accessi interni, Sistema di analisi degli accessi IAM genera esiti quando esiste un possibile percorso di accesso tra un ruolo o un utente IAM all'interno della tua organizzazione e le risorse specificate. Analogamente all'analisi degli accessi esterni, l'ambito scelto (organizzazione o account) determina ciò che viene considerato interno. Se si seleziona un'organizzazione come ambito, Strumento di analisi degli accessi IAM genererà esiti per i percorsi di accesso tra i principali e le

risorse all'interno dell'organizzazione. Se si seleziona un account, verranno generati gli esiti per i percorsi di accesso all'interno di quell'account specifico. Strumento di analisi degli accessi IAM utilizza il ragionamento automatico per valutare tutte le policy IAM e monitorare chi ha accesso alle tue risorse.

La combinazione degli esiti relativi agli accessi esterni e interni con la stessa zona di attendibilità fornisce un'analisi completa di tutti i possibili accessi a una particolare risorsa, sia dall'interno sia dall'esterno del limite di attendibilità definito.

Per gli accessi non utilizzati, Sistema di analisi degli accessi IAM genera esiti per gli accessi non utilizzati concessi nell'organizzazione e negli account AWS . Quando crei un analizzatore di accessi inutilizzato, IAM Access Analyzer monitora continuamente tutti i ruoli e gli utenti IAM nell' AWS organizzazione e negli account e genera risultati per gli accessi non utilizzati. Sistema di analisi degli accessi IAM genera i seguenti tipi di risultati per gli accessi inutilizzati:

- Ruoli inutilizzati: ruoli senza attività di accesso all'interno della finestra di utilizzo specificata.
- Chiavi e password di accesso utente IAM non utilizzate: credenziali appartenenti a utenti IAM che non sono state utilizzate per accedere a te nella finestra di utilizzo specificata. Account AWS
- Autorizzazioni inutilizzate: autorizzazioni a livello di servizio e a livello di operazione che non sono state utilizzate da un ruolo all'interno della finestra di utilizzo specificata. Sistema di analisi degli accessi IAM utilizza policy basate sull'identità associate ai ruoli per determinare i servizi e le operazioni a cui tali ruoli possono accedere. Sistema di analisi degli accessi IAM supporta la revisione delle autorizzazioni inutilizzate per tutte le autorizzazioni a livello di servizio. Per un elenco completo delle autorizzazioni a livello di operazione supportate per i risultati di accesso inutilizzati, consulta Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM.

> **Note**
>
> Sistema di analisi degli accessi IAM offre gli esiti sugli accessi esterni a titolo gratuito. Sono previsti addebiti per gli esiti relativi agli accessi non utilizzati in base al numero di ruoli e utenti IAM analizzati per analizzatore al mese. Sono previsti anche costi per i risultati degli accessi interni in base al numero di AWS risorse monitorate per analizzatore al mese. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

Argomenti

- Come funzionano i risultati di Sistema di analisi degli accessi IAM
- Nozioni di base su AWS Identity and Access Management Access Analyzer
- Visualizzare il pannello di controllo dei risultati di Sistema di analisi degli accessi IAM
- Rivedere i risultati di Sistema di analisi degli accessi IAM
- Filtrare i risultati di Sistema di analisi degli accessi IAM
- Archiviare i risultati di Sistema di analisi degli accessi IAM
- Risolvere i risultati di Sistema di analisi degli accessi IAM
- Esiti di errore di Sistema di analisi degli accessi IAM
- Tipi di risorse di Sistema di analisi degli accessi IAM supportate per gli accessi esterni e interni
- Amministratore delegato per Sistema di analisi degli accessi IAM.
- Regole di archiviazione
- Monitoraggio AWS Identity and Access Management Access Analyzer con Amazon EventBridge
- Integra IAM Access Analyzer con AWS Security Hub CSPM
- Registrazione delle chiamate API IAM Access Analyzer con AWS CloudTrail
- Chiavi di filtro di Sistema di analisi degli accessi IAM
- Utilizzo di ruoli collegati ai servizi per AWS Identity and Access Management Access Analyzer

Come funzionano i risultati di Sistema di analisi degli accessi IAM

Questo argomento descrive i concetti e i termini utilizzati in IAM Access Analyzer per aiutarti a familiarizzare con il modo in cui IAM Access Analyzer monitora l'accesso alle tue AWS risorse.

Risultati dell'accesso esterno

I risultati degli accessi esterni vengono generati una sola volta per ogni istanza di risorsa condivisa al di fuori della zona di attendibilità. Ogni volta che una policy basata sulle risorse viene modificata, Sistema di analisi degli accessi IAM la rianalizza. Se la policy aggiornata condivide una risorsa già identificata in un risultato, ma con autorizzazioni o condizioni diverse, viene generato un nuovo risultato per quell'istanza della condivisione delle risorse. Anche le modifiche a una policy di controllo delle risorse che influiscono sulla Limitazione della policy di controllo delle risorse (RCP) generano un nuovo esito. Strumento di analisi degli accessi IAM valuta anche le configurazioni di controllo degli accessi stabilite da policy dichiarative. Se l'accesso nel primo risultato viene rimosso, il risultato viene aggiornato nello stato Risolto.

Lo stato di tutti i risultati rimane Attivo fino a quando non vengono archiviati o non si rimuove l'accesso che ha generato il risultato. Quando si rimuove l'accesso, lo stato del risultato viene aggiornato in Risolto.

> **Note**
>
> Dopo la modifica di una policy, perché Sistema di analisi degli accessi IAM possa analizzare la risorsa e aggiornare il risultato degli accessi esterni, potrebbero essere necessari fino a 30 minuti. Le modifiche a una policy di controllo delle risorse (RCP) non attivano una nuova scansione della risorsa segnalata nel risultato. In questo caso, Sistema di analisi degli accessi IAM analizza la policy nuova o aggiornata durante la scansione periodica successiva che avviene entro 24 ore.

Come il Sistema di analisi degli accessi IAM genera risultati per gli accessi esterni

AWS Identity and Access Management Access Analyzer utilizza una tecnologia chiamata Zelkova per analizzare le politiche IAM e identificare l'accesso esterno alle risorse.

Zelkova traduce le policy IAM in istruzioni logiche equivalenti e gestisce una suite di risolutori logici generici e specializzati (teorie dei moduli di soddisfacibilità) per il problema. Sistema di analisi degli accessi IAM applica Zelkova ripetutamente a una policy con query sempre più specifiche per caratterizzare classi di comportamenti consentite dalla policy, in base al contenuto della policy stessa. Per ulteriori informazioni sulle teorie dei moduli di soddisfacibilità, consulta Teorie dei moduli di soddisfacibilità.

Per i sistemi di analisi degli accessi esterni, Sistema di analisi degli accessi IAM non esamina i log di accesso per determinare se un'entità esterna accede a una risorsa all'interno della zona di attendibilità. Genera invece un risultato quando una policy basata sulle risorse consente l'accesso a una risorsa, indipendentemente se l'entità esterna ha eseguito l'accesso alla risorsa.

Sistema di analisi degli accessi IAM, inoltre, non considera lo stato di alcun account esterno al momento della sua determinazione. In altre parole, se indica che l'account 111122223333 può accedere al bucket Amazon S3, non ha alcuna informazione sugli utenti, i ruoli, le policy di controllo dei servizi o altre configurazioni pertinenti in tale account. Questo è per la privacy del cliente, in quanto Sistema di analisi degli accessi IAM non conosce il proprietario dell'altro account. Questo vale anche per la sicurezza, in quanto è importante conoscere i potenziali accessi esterni anche se al momento non ci sono principali in grado di utilizzarli.

Sistema di analisi degli accessi IAM considera solo alcune chiavi di condizione IAM che gli utenti esterni non possono influenzare direttamente o che hanno un impatto sull'autorizzazione. Per esempi di chiavi di condizione considerate da Sistema di analisi degli accessi IAM, consulta Chiavi di filtro di Sistema di analisi degli accessi IAM.

Attualmente IAM Access Analyzer non riporta i risultati dei Servizio AWS principali o degli account di servizio interni. Nei rari casi in cui il sistema di analisi degli accessi non è in grado di determinare completamente se un'istruzione della policy concede l'accesso a un'entità esterna, si sbaglia nel dichiarare un risultato falso positivo. Ciò si verifica perché Sistema di analisi degli accessi IAM è progettato per fornire una visione completa della condivisione delle risorse nell'account e per ridurre al minimo i falsi negativi.

Esiti relativi agli accessi interni

Per utilizzare l'analisi degli accessi interni, è necessario innanzitutto configurare l'analizzatore selezionando le risorse specifiche che si desidera monitorare. Una volta configurati, gli esiti relativi agli accessi interni vengono generati quando un principale (utente o ruolo IAM) all'interno dell'organizzazione o dell'account ha accesso alle risorse selezionate. Viene generato un nuovo esito quando l'analizzatore esegue nuovamente la scansione delle risorse specificate e identifica un principale che ha accesso alle risorse. Se la policy aggiornata autorizza un principale già identificato in un esito ma con autorizzazioni o condizioni diverse, viene generato un nuovo esito per l'istanza del principale e della risorsa. Questa policy aggiornata potrebbe essere una policy basata sulle risorse, una policy basata sull'identità, una policy di controllo dei servizi (SCP) o una policy di controllo delle risorse (RCP). Strumento di analisi degli accessi IAM valuta anche le configurazioni di controllo degli accessi stabilite da policy dichiarative.

> **Note**
>
> I risultati di accesso interni sono disponibili solo utilizzando l'azione API ListFindings V2.

Come Sistema di analisi degli accessi IAM genera esiti relativi agli accessi esterni

Per analizzare gli accessi interni, è necessario creare un analizzatore separato per gli esiti relativi agli accessi interni per le proprie risorse, anche se è già stato creato un analizzatore per generare esiti relativi agli accessi esterni o agli accessi non utilizzati.

Dopo aver creato l'analizzatore degli accessi interni, Strumento di analisi degli accessi IAM valuta tutte le policy basate sulle risorse, le policy basate sulle identità, le policy di controllo dei servizi

(SCP), le policy di controllo delle risorse (RCP) e i limiti delle autorizzazioni all'interno di un account o di un'organizzazione specificati.

Creando un analizzatore dedicato agli accessi interni alle risorse selezionate, puoi identificare:

- Quando un principale della tua organizzazione o del tuo account può accedere alle risorse selezionate
- Il totale delle autorizzazioni effettive consentite per un principale in base all'intersezione di tutte le policy applicabili
- Percorsi di accesso complessi in cui un principale ottiene l'accesso in base alla combinazione di policy basate sull'identità e policy delle risorse

> **Note**
>
> Strumento di analisi degli accessi IAM non è in grado di generare esiti degli accesso interni per le organizzazioni che contengono più di 70.000 principali (utenti e ruoli IAM combinati).

Risultati degli accessi inutilizzati

Gli esiti relativi agli accessi non utilizzati vengono generati per le entità IAM (principali) all'interno dell'account o dell'organizzazione selezionati in base al numero di giorni specificato durante la creazione dell'analizzatore. Viene generato un nuovo risultato quando l'analizzatore esegue nuovamente la scansione delle entità se viene soddisfatta una delle seguenti condizioni:

- Un ruolo è inattivo per il numero specificato di giorni.
- Un'autorizzazione, una password utente o una chiave di accesso utente inutilizzate superano il numero di giorni specificato.

> **Note**
>
> I risultati di accesso non utilizzati sono disponibili solo utilizzando l'azione API ListFindings V2.

Come Sistema di analisi degli accessi IAM genera risultati per gli accessi inutilizzati

Per analizzare gli accessi non utilizzati, è necessario creare un analizzatore separato per gli esiti relativi agli accessi non utilizzati per i propri ruoli, anche se è già stato creato un analizzatore per generare esiti relativi agli accessi esterni o agli accessi interni per le proprie risorse.

Dopo aver creato il sistema di analisi degli accessi inutilizzati, Sistema di analisi degli accessi IAM esamina l'attività di accesso per identificare gli accessi inutilizzati. IAM Access Analyzer esamina le ultime informazioni ad accesso per tutti gli utenti IAM, i ruoli IAM inclusi i ruoli di servizio, le chiavi di accesso degli utenti e le password degli utenti nell'organizzazione e negli account AWS . Questo ti aiuta a identificare gli accessi non utilizzati.

> **Note**
>
> Un ruolo collegato al servizio è un tipo speciale di ruolo di servizio collegato Servizio AWS e di proprietà del servizio. Service-linked i ruoli non vengono analizzati da analizzatori di accesso inutilizzati.

Per gli utenti e i ruoli IAM attivi, Sistema di analisi degli accessi IAM utilizza le informazioni sull'ultimo accesso per le operazioni e i servizi IAM per identificare le autorizzazioni inutilizzate. Ciò consente di scalare il processo di revisione a livello di AWS organizzazione e account. Puoi utilizzare le informazioni sull'ultimo accesso per un'analisi più approfondita dei singoli ruoli. Ciò fornisce informazioni più dettagliate su quali autorizzazioni specifiche non vengono utilizzate.

Creando un analizzatore dedicato agli accessi non utilizzati, è possibile esaminare e identificare in modo completo gli accessi inutilizzati in tutto l' AWS ambiente, integrando i risultati generati dall'analizzatore di accesso esterno esistente.

Nozioni di base su AWS Identity and Access Management Access Analyzer

Utilizza le informazioni contenute in questo argomento per conoscere i requisiti necessari per utilizzare e gestire AWS Identity and Access Management Access Analyzer.

Se utilizzi IAM Access Analyzer all'interno di un' AWS Organizations organizzazione, puoi Aggiungere un amministratore delegato per Sistema di analisi degli accessi IAM. farlo. Per informazioni sui limiti di servizio, consultaQuote Sistema di analisi degli accessi AWS IAM.

Autorizzazioni necessarie per utilizzare Sistema di analisi degli accessi IAM

Per configurare correttamente e utilizzare Sistema di analisi degli accessi IAM, all'account che utilizzi devono essere concesse le autorizzazioni necessarie.

AWS politiche gestite per IAM Access Analyzer

AWS Identity and Access Management Access Analyzer fornisce policy AWS gestite per aiutarti a iniziare rapidamente.

- IAMAccessAnalyzerFullAccess- Consente l'accesso completo a IAM Access Analyzer per gli amministratori. Questa policy consente inoltre di creare i ruoli collegati ai servizi necessari per consentire a Sistema di analisi degli accessi IAM di analizzare le risorse nell'account o nell'organizzazione AWS .
- IAMAccessAnalyzerReadOnlyAccess- Consente l'accesso in sola lettura a IAM Access Analyzer. È necessario aggiungere ulteriori policy alle identità IAM (utenti, gruppi di utenti o ruoli) per consentire loro di visualizzare i risultati.

Risorse definite da Sistema di analisi degli accessi IAM

Per visualizzare le risorse definite da Sistema di analisi degli accessi IAM, consulta Tipi di risorsa definiti da Sistema di analisi degli accessi IAM in Service Authorization Reference.

Autorizzazioni di servizio necessarie per Sistema di analisi degli accessi IAM

Sistema di analisi degli accessi IAM utilizza un ruolo collegato ai servizi (SLR) chiamato `AWSServiceRoleForAccessAnalyzer`. Questa SLR garantisce al servizio l'accesso in sola lettura per analizzare le AWS risorse con politiche basate sulle risorse e analizzare gli accessi non utilizzati per conto dell'utente. Il servizio crea il ruolo nel tuo account nei seguenti casi:

- Crei un analizzatore degli accessi esterni con il tuo account come zona di attendibilità.
- Crei un analizzatore degli accessi inutilizzati con il tuo account come account selezionato.
- Crei un analizzatore degli accessi interni con il tuo account come zona di attendibilità.

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per AWS Identity and Access Management Access Analyzer.

> **Note**
>
> Sistema di analisi degli accessi IAM è un servizio regionale. Per gli accessi esterni e interni, devi abilitare Sistema di analisi degli accessi IAM in ogni Regione in modo indipendente. Per gli accessi inutilizzati, i risultati relativi all'analizzatore non cambiano in base alla regione. Non è necessario creare un analizzatore in ogni regione in cui sono disponibili risorse.

In alcuni casi, dopo aver creato un analizzatore in Sistema di analisi degli accessi IAM, il pannello di controllo o la pagina Risultati vengono caricati senza esiti o riepiloghi. Ciò potrebbe essere dovuto a un ritardo nella console per la compilazione dei risultati. Potrebbe essere necessario aggiornare manualmente il browser o ricontrollare più tardi per visualizzare i risultati o il riepilogo. Se ancora non viene visualizzato alcun risultato per l'analizzatore degli accessi esterni, non hai nel tuo account le risorse supportate a cui è possibile accedere da un'entità esterna. Se una policy che concede l'accesso a un'entità esterna viene applicata a una risorsa, Sistema di analisi degli accessi IAM genera un risultato.

> **Note**
>
> Per gli analizzatori degli accessi esterni, potrebbero essere necessari fino a 30 minuti dopo la modifica di una policy affinché Sistema di analisi degli accessi IAM possa analizzare la risorsa e generare un nuovo esito o aggiornarne uno esistente per l'accesso alla risorsa. Quando si crea un analizzatore degli accessi interni, potrebbero essere necessari diversi minuti oppure ore prima che gli esiti siano disponibili. Dopo la scansione iniziale, Strumento di analisi degli accessi IAM esegue automaticamente una nuova scansione di tutte le policy ogni 24 ore.
>
> Per tutti i tipi di gli analizzatori degli accessi, gli aggiornamenti degli esiti potrebbero non essere riportati immediatamente nel pannello di controllo.

Autorizzazioni Sistema di analisi degli accessi IAM necessarie per visualizzare la dashboard dei risultati

Per visualizzare la dashboard dei risultati di Sistema di analisi degli accessi IAM, all'account che utilizzi deve essere concesso l'accesso per eseguire le seguenti operazioni necessarie:

- `GetAnalyzer`
- `ListAnalyzers`

- `GetFindingsStatistics`

Per visualizzare le operazioni definite da Sistema di analisi degli accessi IAM, consulta Operazioni definite da Sistema di analisi degli accessi IAM in Service Authorization Reference.

Stato di Sistema di analisi degli accessi IAM

Per visualizzare lo stato degli analizzatori, scegli Analyzers (Analizzatori). Gli analizzatori creati per un'organizzazione o un account possono avere lo stato seguente:

Stato	Description
Attivo	Per gli analizzatori degli accessi interni ed esterni, l'analizzatore monitora attivamente le risorse all'interno della zona di attendibilità. L'analizzatore genera attivamente nuovi risultati e aggiorna quelli esistenti. Per gli analizzatori di accesso non utilizzati, l'analizzatore monitora attivamente gli accessi non utilizzati all'interno dell'organizzazion e selezionata o nel periodo di tracciamento specificato. Account AWS L'analizzatore genera attivamente nuovi risultati e aggiorna quelli esistenti.
Creazione in corso	La creazione dell'analizzatore è ancora in corso. Al termine, l'analizzatore diventa attivo.
Disabilitato	L'analizzatore è disabilitato a causa di un'azione intrapresa dall'amministratore. AWS Organizations ad esempio la rimozione dell'account dell'analizzatore come amministr atore delegato per Sistema di analisi degli accessi IAM. Quando l'analizzatore è in stato disabilitato, non genera nuovi risultati né aggiorna quelli esistenti.

Stato	Description
Non riuscito	Creazione dell'analizzatore non è riuscita a causa di un problema di configurazione. L'analizzatore non genererà alcun risultato. Eliminare l'analizzatore e crearne uno nuovo.

Crea un sistema di analisi degli accessi esterni per Sistema di analisi degli accessi IAM

Per abilitare un analizzatore degli accessi esterni in una regione, è necessario creare un analizzatore in tale regione. È necessario creare un analizzatore degli accessi esterni in ogni regione in cui desideri monitorare l'accesso alle risorse.

> **Note**
>
> Dopo la creazione o l'aggiornamento di un analizzatore, può essere necessario del tempo prima che gli esiti siano disponibili.

Crea un analizzatore di accessi esterno con Account AWS come zona di fiducia

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Scegliere Create analyzer (Crea analizzatore).
4. Nella sezione Analisi, scegli Analisi della risorsa - accessi esterni.
5. Nella sezione Dettagli dell'analizzatore, verifica che la regione visualizzata sia quella in cui desideri abilitare Sistema di analisi degli accessi IAM.
6. Inserire un nome per l'analizzatore.
7. Scegli Account attuale come zona di attendibilità per l'analizzatore.

> **Note**
>
> Se il tuo account non è l'account di AWS Organizations gestione o l'account di amministratore delegato, puoi creare un solo analizzatore con il tuo account come zona di fiducia.

8. Opzionale. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
9. Scegliere Create analyzer (Crea analizzatore).

Quando crei un analizzatore degli accessi esterni per abilitare Sistema di analisi degli accessi IAM, nell'account viene creato un ruolo collegato ai servizi denominato `AWSServiceRoleForAccessAnalyzer`.

Creare un sistema di analisi degli accessi esterni con l'organizzazione come zona di attendibilità

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Scegliere Create analyzer (Crea analizzatore).
4. Nella sezione Analisi, scegli Analisi della risorsa - accessi esterni.
5. Nella sezione Dettagli dell'analizzatore, verifica che la regione visualizzata sia quella in cui desideri abilitare Sistema di analisi degli accessi IAM.
6. Inserire un nome per l'analizzatore.
7. Scegli Organizzazione attuale come zona di attendibilità per l'analizzatore.
8. Opzionale. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
9. Seleziona Invia.

Quando si crea un analizzatore degli accessi esterni con l'organizzazione come zona di attendibilità, in ogni account dell'organizzazione viene creato un ruolo collegato al servizio denominato `AWSServiceRoleForAccessAnalyzer`.

Gestire un sistema di analisi degli accessi esterni per Sistema di analisi degli accessi IAM

Per abilitare un analizzatore degli accessi esterni in una regione, è necessario creare un analizzatore in tale regione. È necessario creare un analizzatore degli accessi esterni in ogni regione in cui desideri monitorare l'accesso alle risorse.

> **Note**
>
> Dopo la creazione o l'aggiornamento di un analizzatore, può essere necessario del tempo prima che gli esiti siano disponibili.

Aggiornare un sistema di analisi degli accessi esterni

Per aggiornare un sistema di analisi degli accessi esterni, utilizza la procedura seguente.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Nella sezione Analizzatori, scegli il nome dell'analizzatore degli accessi esterni da gestire.
4. Nella scheda Regole di archiviazione, puoi creare, modificare o eliminare le regole di archiviazione per il sistema di analisi. Per ulteriori informazioni, consulta Regole di archiviazione.
5. Nella scheda Tag, puoi gestire e creare tag per il sistema di analisi. Per ulteriori informazioni, consulta Tag per AWS Identity and Access Management risorse.

Eliminare un sistema di analisi degli accessi esterni

Per eliminare un sistema di analisi degli accessi esterni, utilizza la procedura seguente. Quando si elimina un sistema di analisi, le risorse non vengono più monitorate e non vengono generati nuovi risultati. Tutti i risultati generati dall'analizzatore vengono eliminati.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Nella sezione Analizzatori, scegli il nome dell'analizzatore degli accessi esterni da eliminare.
4. Scegli Elimina analizzatore.
5. Inserisci delete e scegli Elimina per confermare l'eliminazione del sistema di analisi.

Creazione di un analizzatore degli accessi interni di Sistema di analisi degli accessi IAM

Per abilitare un analizzatore degli accessi interni in una Regione, è necessario creare un analizzatore in tale Regione. È necessario creare un analizzatore degli accessi interni in ogni Regione in cui si desidera monitorare l'accesso alle risorse.

Sistema di analisi degli accessi IAM addebita l'analisi degli accessi interni in base al numero di risorse monitorate per analizzatore al mese. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi IAM.

> **Note**
>
> Dopo la creazione o l'aggiornamento di un analizzatore, può essere necessario del tempo prima che gli esiti siano disponibili.
> Strumento di analisi degli accessi IAM non è in grado di generare esiti degli accesso interni per le organizzazioni che contengono più di 70.000 principali (utenti e ruoli IAM combinati).
> È possibile creare un solo analizzatore degli accessi interni a livello di organizzazione in un'organizzazione AWS .

Crea un analizzatore di accesso interno con Account AWS come zona di fiducia

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Scegliere Create analyzer (Crea analizzatore).
4. Nella sezione Analisi, scegli Analisi della risorsa - accessi interni.
5. Nella sezione Dettagli dell'analizzatore, verifica che la regione visualizzata sia quella in cui desideri abilitare Sistema di analisi degli accessi IAM.
6. Inserire un nome per l'analizzatore.
7. Scegli Account attuale come zona di attendibilità per l'analizzatore.

   > **Note**
   >
   > Se il tuo account non è l'account di AWS Organizations gestione o l'account di amministratore delegato, puoi creare un solo analizzatore con il tuo account come zona di fiducia.

8. Nella sezione Risorse da analizzare, aggiungi le risorse che l'analizzatore deve monitorare.

   - Per aggiungere risorse per account, scegli Aggiungi > Aggiungi risorse dagli account selezionati.

     1. Scegli Tutti i tipi di risorse supportati oppure scegli Definisci tipi di risorse specifici e seleziona i tipi di risorse dall'elenco Tipo di risorsa.

        Gli analizzatori degli accessi interni supportano i seguenti tipi di risorse:

        - Bucket Amazon Simple Storage Service

- Bucket di directory di Amazon Simple Storage Service
- Amazon Relational Database Service
- Snapshot di cluster di database di Amazon Relational Database Service
- Flussi Amazon DynamoDB
- Tabelle Amazon DynamoDB

2. Scegli Aggiungi risorse.

- Per aggiungere risorse in base al nome della risorsa Amazon (ARN), scegli Aggiungi risorse > Aggiungi risorse incollando l'ARN della risorsa.

> **Note**
>
> ARNs devono corrispondere esattamente: i caratteri jolly non sono supportati. Per Amazon S3, sono supportati solo i bucket. ARNs Gli oggetti ARNs e i prefissi Amazon S3 non sono supportati.

1. Per ogni ARN di risorsa, inserisci l'ID del proprietario dell'account e l'ARN della risorsa separati da una virgola. Inserisci un ID proprietario dell'account e un ARN della risorsa per riga.
2. Scegli Aggiungi risorse.

- Per aggiungere risorse tramite un file CSV, scegli Aggiungi risorse > Aggiungi risorse caricando un file CSV.

  Puoi utilizzare Esploratore di risorse AWS per cercare risorse nei tuoi account ed esportare un file CSV. Dopodiché, puoi caricare il file CSV per configurare le risorse che l'analizzatore dovrà monitorare.

1. Seleziona Scegli file e seleziona il file CSV dal tuo computer.
2. Scegli Aggiungi risorse.

9. Opzionale. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
10. Scegliere Create analyzer (Crea analizzatore).

Quando crei un analizzatore degli accessi interni per abilitare Sistema di analisi degli accessi IAM, nell'account viene creato un ruolo collegato ai servizi denominato `AWSServiceRoleForAccessAnalyzer`.

Creazione di un analizzatore degli accessi esterni con l'organizzazione come zona di attendibilità

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Scegliere Create analyzer (Crea analizzatore).
4. Nella sezione Analisi, scegli Analisi della risorsa - accessi interni.
5. Nella sezione Dettagli dell'analizzatore, verifica che la regione visualizzata sia quella in cui desideri abilitare Sistema di analisi degli accessi IAM.
6. Inserire un nome per l'analizzatore.
7. Scegli Intera organizzazione come zona di attendibilità per l'analizzatore.
8. Nella sezione Risorse da analizzare, aggiungi le risorse che l'analizzatore deve monitorare.

   - Per aggiungere risorse per l'account, scegli Aggiungi risorse **>** Aggiungi risorse dagli account selezionati.

     1. Scegli Tutti i tipi di risorse supportati oppure scegli Definisci tipi di risorse specifici e seleziona i tipi di risorse dall'elenco Tipo di risorsa.

        Gli analizzatori degli accessi interni supportano i seguenti tipi di risorse:

        - Bucket Amazon Simple Storage Service
        - Bucket di directory di Amazon Simple Storage Service
        - Amazon Relational Database Service
        - Snapshot di cluster di database di Amazon Relational Database Service
        - Flussi Amazon DynamoDB
        - Tabelle Amazon DynamoDB

     2. Per selezionare gli account della tua organizzazione, scegli Seleziona dall'organizzazione. Nella sezione Seleziona account, scegli Gerarchia per selezionare gli account in base alla struttura organizzativa oppure Elenco per selezionare gli account da un elenco di tutti gli account dell'organizzazione.

        Per inserire manualmente gli account della tua organizzazione, scegli Inserisci AWS l'ID dell'account. Inserisci uno o più Account AWS IDs separati da virgole nel campo ID dell'AWS account.

     3. Scegli Aggiungi risorse.

- Per aggiungere risorse in base al nome della risorsa Amazon (ARN), scegli Aggiungi risorse > Aggiungi risorse incollando l'ARN della risorsa.

  > **Note**
  >
  > ARNs devono corrispondere esattamente: i caratteri jolly non sono supportati. Per Amazon S3, sono supportati solo i bucket. ARNs Gli oggetti ARNs e i prefissi Amazon S3 non sono supportati.

  1. Per ogni ARN di risorsa, inserisci l'ID del proprietario dell'account e l'ARN della risorsa separati da una virgola. Inserisci un ID proprietario dell'account e un ARN della risorsa per riga.
  2. Scegli Aggiungi risorse.
- Per aggiungere risorse tramite un file CSV, scegli Aggiungi risorse > Aggiungi risorse caricando un file CSV.

  Puoi utilizzare Esploratore di risorse AWS per cercare risorse nei tuoi account ed esportare un file CSV. Dopodiché, puoi caricare il file CSV per configurare le risorse che l'analizzatore dovrà monitorare.

  1. Seleziona Scegli file e seleziona il file CSV dal tuo computer.
  2. Scegli Aggiungi risorse.

9. Opzionale. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
10. Seleziona Invia.

Quando si crea un analizzatore degli accessi interni con l'organizzazione come zona di attendibilità, in ogni account dell'organizzazione viene creato un ruolo collegato al servizio denominato `AWSServiceRoleForAccessAnalyzer`.

Gestione di un analizzatore degli accessi interni di Sistema di analisi degli accessi IAM

Per abilitare un analizzatore degli accessi interni in una Regione, è necessario creare un analizzatore in tale Regione. È necessario creare un analizzatore degli accessi interni in ogni Regione in cui si desidera monitorare l'accesso alle risorse.

> **Note**
>
> Dopo la creazione o l'aggiornamento di un analizzatore, può essere necessario del tempo prima che gli esiti siano disponibili.

Aggiornamento di un analizzatore degli accessi interni

Utilizza la seguente procedura per aggiornare un analizzatore degli accessi interni.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Nella sezione Analizzatori, scegli il nome dell'analizzatore degli accessi interni da gestire.
4. Nella scheda Regole di archiviazione, puoi creare, modificare o eliminare le regole di archiviazione per il sistema di analisi. Per ulteriori informazioni, consulta Regole di archiviazione.
5. Nella scheda Tag, puoi gestire e creare tag per il sistema di analisi. Per ulteriori informazioni, consulta Tag per AWS Identity and Access Management risorse.
6. Nella scheda Risorse, scegli Modifica nella sezione Risorse da analizzare.
   a. Per aggiungere risorse per account, scegli Aggiungi risorse > Aggiungi risorse dagli account selezionati.
      1. Scegli Tutti i tipi di risorse supportati oppure scegli Definisci tipi di risorse specifici e seleziona i tipi di risorse dall'elenco Tipo di risorsa.

         Gli analizzatori degli accessi interni supportano i seguenti tipi di risorse:

         - Bucket Amazon Simple Storage Service
         - Bucket di directory di Amazon Simple Storage Service
         - Amazon Relational Database Service
         - Snapshot di cluster di database di Amazon Relational Database Service
         - Flussi Amazon DynamoDB
         - Tabelle Amazon DynamoDB
      2. Scegli Aggiungi risorse.
   b. Per aggiungere risorse in base al nome della risorsa Amazon (ARN), scegli Aggiungi risorse > Aggiungi risorse incollando l'ARN della risorsa.

> **Note**
>
> ARNs devono corrispondere esattamente: i caratteri jolly non sono supportati. Per Amazon S3, sono supportati solo i bucket. ARNs Gli oggetti ARNs e i prefissi Amazon S3 non sono supportati.

1. Per ogni ARN di risorsa, inserisci l'ID del proprietario dell'account e l'ARN della risorsa separati da una virgola. Inserisci un ID proprietario dell'account e un ARN della risorsa per riga.
2. Scegli Aggiungi risorse.

c. Per aggiungere risorse tramite un file CSV, scegli Aggiungi risorse > Aggiungi risorse caricando un file CSV.

   Puoi utilizzare Esploratore di risorse AWS per cercare risorse nei tuoi account ed esportare un file CSV. Dopodiché, puoi caricare il file CSV per configurare le risorse che l'analizzatore dovrà monitorare.

   1. Seleziona Scegli file e seleziona il file CSV dal tuo computer.
   2. Scegli Aggiungi risorse.

d. Per rimuovere risorse dall'analizzatore, seleziona la casella di controllo accanto alle risorse da rimuovere e scegli Rimuovi.

e. Scegli Save changes (Salva modifiche).

> **Note**
>
> Eventuali aggiornamenti all'analizzatore verranno valutati alla successiva nuova scansione automatica entro 24 ore.

Eliminazione di un analizzatore degli accessi interni

Utilizza la seguente procedura per eliminare un analizzatore degli accessi interni. Quando si elimina un sistema di analisi, le risorse non vengono più monitorate e non vengono generati nuovi risultati. Tutti i risultati generati dall'analizzatore vengono eliminati.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Nella sezione Analizzatori, scegli il nome dell'analizzatore degli accessi interni da eliminare.
4. Scegli Elimina analizzatore.
5. Inserisci delete e scegli Elimina per confermare l'eliminazione del sistema di analisi.

Crea un sistema di analisi degli accessi inutilizzati per Sistema di analisi degli accessi IAM

Creare un sistema di analisi degli accessi inutilizzati per l'account corrente

Utilizza la seguente procedura per creare un analizzatore degli accessi inutilizzati per un singolo Account AWS. Per gli accessi inutilizzati, i risultati relativi all'analizzatore non cambiano in base alla regione. Non è necessario creare un analizzatore in ogni regione in cui sono disponibili risorse.

Sistema di analisi degli accessi IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese da ogni analizzatore. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

> **Note**
>
> Dopo la creazione o l'aggiornamento di un analizzatore, può essere necessario del tempo prima che gli esiti siano disponibili.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Scegliere Create analyzer (Crea analizzatore).
4. Nella sezione Analisi, scegli Analisi del principale - accessi non utilizzati.
5. Inserire un nome per l'analizzatore.
6. In Periodo di monitoraggio, inserisci il numero di giorni per l'analisi. L'analizzatore valuterà solo le autorizzazioni per le entità IAM all'interno dell'account selezionato che sono esistite per l'intero periodo di tracciamento. Ad esempio, se imposti un periodo di tracciamento di 90 giorni, verranno analizzate solo le autorizzazioni che risalgono ad almeno 90 giorni prima e gli esiti

verranno generati se non si registra alcun utilizzo durante questo periodo. Puoi inserire un valore compreso tra 1 e 365 giorni.

7. Nella sezione Dettagli dell'analizzatore, verifica che la regione visualizzata sia quella in cui desideri abilitare Sistema di analisi degli accessi IAM.
8. Per Account selezionati, scegli Account corrente.

> **Note**
>
> Se il tuo account non è l'account di AWS Organizations gestione o l'account amministratore delegato, puoi creare un solo analizzatore con il tuo account come account selezionato.

9. Opzionale. Nella sezione Escludi gli utenti e i ruoli IAM con tag, puoi specificare coppie chiave-valore per utenti e ruoli IAM da escludere dall'analisi degli accessi non utilizzati. I risultati non verranno generati per gli utenti e i ruoli IAM esclusi che corrispondono alle coppie chiave-valore. Per Chiave tag, immetti un numero di caratteri compreso tra 1 e 128 che non abbia il prefisso `aws:`. Per Valore, puoi inserire un valore di lunghezza compresa tra 0 e 256 caratteri. Se non inserisci un valore, la regola viene applicata a tutti i principali con la chiave tag specificata. Scegli Aggiungi nuova esclusione per aggiungere altre coppie chiave-valore da escludere.
10. Opzionale. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
11. Scegliere Create analyzer (Crea analizzatore).

Quando crei un analizzatore degli accessi inutilizzati per abilitare Sistema di analisi degli accessi IAM, nell'account viene creato un ruolo collegato ai servizi denominato `AWSServiceRoleForAccessAnalyzer`.

Creare un sistema di analisi degli accessi inutilizzati con l'organizzazione corrente

Utilizzate la seguente procedura per creare un analizzatore di accessi inutilizzato per consentire a un'organizzazione di esaminare centralmente tutto ciò Account AWS che fa parte di un'organizzazione. Per l'analisi degli accessi inutilizzati, i risultati relativi all'analizzatore non cambiano in base alla regione. Non è necessario creare un analizzatore in ogni regione in cui sono disponibili risorse.

Sistema di analisi degli accessi IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese da ogni analizzatore. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

> **Note**
>
> Se un account membro viene rimosso dall'organizzazione, l'analizzatore degli accessi inutilizzati smetterà di generare nuovi risultati e di aggiornare i risultati esistenti per quell'account dopo 24 ore. I risultati associati all'account membro rimosso dall'organizzazione verranno rimossi definitivamente dopo 90 giorni.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Scegliere Create analyzer (Crea analizzatore).
4. Nella sezione Analisi, scegli Analisi del principale - accessi non utilizzati.
5. Inserire un nome per l'analizzatore.
6. In Periodo di monitoraggio, inserisci il numero di giorni per l'analisi. L'analizzatore valuterà solo le autorizzazioni per le entità IAM all'interno degli account dell'organizzazione selezionata che sono esistiti per l'intero periodo di tracciamento. Ad esempio, se imposti un periodo di tracciamento di 90 giorni, verranno analizzate solo le autorizzazioni che risalgono ad almeno 90 giorni prima e gli esiti verranno generati se non si registra alcun utilizzo durante questo periodo. Puoi inserire un valore compreso tra 1 e 365 giorni.
7. Nella sezione Dettagli dell'analizzatore, verifica che la regione visualizzata sia quella in cui desideri abilitare Sistema di analisi degli accessi IAM.
8. Per Account selezionati, scegli Organizzazione corrente.
9. Opzionale. Nella sezione Escludi Account AWS dall'analisi, puoi scegliere all' Account AWS interno della tua organizzazione di escludere dall'analisi degli accessi non utilizzati. Non verranno generati risultati per gli account esclusi.

    a. Per specificare un singolo account IDs da escludere, scegli Specificare Account AWS ID e inserisci l'account IDs separato da virgole nel campo Account AWS ID. Scegli Escludi. Gli account vengono quindi elencati nella tabella Account AWS da escludere.

    b. Per scegliere da un elenco di account dell'organizzazione da escludere, seleziona Scegli dall'organizzazione.

        i. Puoi cercare gli account per nome, e-mail e ID account nel campo Escludi account dall'organizzazione.

ii. Scegli Gerarchia per visualizzare gli account per unità organizzativa o scegli Elenco per visualizzare un elenco di tutti i singoli account dell'organizzazione.

iii. Scegli Escludi tutti gli account correnti per escludere tutti gli account in un'unità organizzativa o scegli Escludi per escludere singoli account.

Gli account vengono quindi elencati nella tabella Account AWS da escludere.

> **Note**
>
> Gli account esclusi non possono includere l'account proprietario del sistema di analisi dell'organizzazione. Quando vengono aggiunti nuovi account all'organizzazione, questi non vengono esclusi dall'analisi anche se in precedenza erano stati esclusi tutti gli account correnti all'interno di un'unità organizzativa. Per ulteriori informazioni sull'esclusione degli account dopo aver creato un sistema di analisi di accessi inutilizzati, consulta Gestire un sistema di analisi degli accessi inutilizzati per Sistema di analisi degli accessi IAM.

10. Opzionale. Nella sezione Escludi gli utenti e i ruoli IAM con tag, puoi specificare coppie chiave-valore per utenti e ruoli IAM da escludere dall'analisi degli accessi non utilizzati. I risultati non verranno generati per gli utenti e i ruoli IAM esclusi che corrispondono alle coppie chiave-valore. Per Chiave tag, immetti un numero di caratteri compreso tra 1 e 128 che non abbia il prefisso `aws:`. Per Valore, puoi inserire un valore di lunghezza compresa tra 0 e 256 caratteri. Se non inserisci un valore, la regola viene applicata a tutti i principali con la chiave tag specificata. Scegli Aggiungi nuova esclusione per aggiungere altre coppie chiave-valore da escludere.
11. Opzionale. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
12. Scegliere Create analyzer (Crea analizzatore).

Quando crei un analizzatore degli accessi inutilizzati per abilitare Sistema di analisi degli accessi IAM, nell'account viene creato un ruolo collegato ai servizi denominato `AWSServiceRoleForAccessAnalyzer`.

Gestire un sistema di analisi degli accessi inutilizzati per Sistema di analisi degli accessi IAM

Utilizzare le informazioni contenute in questo argomento per scoprire come aggiornare o eliminare un sistema di analisi degli accessi inutilizzati esistente.

> **Note**
>
> Dopo la creazione o l'aggiornamento di un analizzatore, può essere necessario del tempo prima che gli esiti siano disponibili.

Aggiornare un sistema di analisi degli accessi inutilizzati

Utilizza la seguente procedura per aggiornare un sistema di analisi degli accessi inutilizzati.

Sistema di analisi degli accessi IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese da ogni analizzatore. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
3. Nella sezione Analizzatori, scegli il nome dell'analizzatore degli accessi non utilizzati da gestire.
4. Nella scheda Esclusione, se l'analizzatore è stato creato per un'organizzazione come ambito di analisi, scegli Gestisci nella sezione Account AWS esclusi.

    a. Per specificare un singolo account IDs da escludere, scegli Specificare Account AWS ID e inserisci l'account IDs separato da virgole nel campo Account AWS ID. Scegli Escludi. Gli account vengono quindi elencati nella tabella Account AWS da escludere.

    b. Per scegliere da un elenco di account dell'organizzazione da escludere, seleziona Scegli dall'organizzazione.

        i. Puoi cercare gli account per nome, e-mail e ID account nel campo Escludi account dall'organizzazione.

        ii. Scegli Gerarchia per visualizzare gli account per unità organizzativa o scegli Elenco per visualizzare un elenco di tutti i singoli account dell'organizzazione.

        iii. Scegli Escludi tutti gli account correnti per escludere tutti gli account in un'unità organizzativa o scegli Escludi per escludere singoli account.

        Gli account vengono quindi elencati nella tabella Account AWS da escludere.

    c. Per rimuovere gli account da escludere, scegli Rimuovi accanto all'account nella tabella Account AWS da escludere.

    d. Scegli Save changes (Salva modifiche).

> **Note**
>
> - Gli account esclusi non possono includere l'account proprietario del sistema di analisi dell'organizzazione.
> - Quando vengono aggiunti nuovi account all'organizzazione, questi non vengono esclusi dall'analisi anche se in precedenza erano stati esclusi tutti gli account correnti all'interno di un'unità organizzativa.
> - Dopo aver aggiornato le esclusioni per un sistema di analisi, possono essere necessari fino a due giorni per l'elenco degli account esclusi da aggiornare.

5. Nella scheda Esclusione, scegli Gestisci nella sezione Utenti e ruoli IAM esclusi con tag.
   a. Puoi specificare coppie chiave-valore per utenti e ruoli IAM da escludere dall'analisi degli accessi inutilizzati. Per Chiave tag, immetti un numero di caratteri compreso tra 1 e 128 che non abbia il prefisso `aws:`. Per Valore, puoi inserire un valore di lunghezza compresa tra 0 e 256 caratteri. Se non inserisci un valore, la regola viene applicata a tutti i principali con la chiave tag specificata.
   b. Scegli Aggiungi nuova esclusione per aggiungere altre coppie chiave-valore da escludere.
   c. Per rimuovere le coppie chiave-valore da escludere, scegli Rimuovi accanto alla coppia chiave-valore.
   d. Scegli Save changes (Salva modifiche).
6. Nella scheda Regole di archiviazione, puoi creare, modificare o eliminare le regole di archiviazione per il sistema di analisi. Per ulteriori informazioni, consulta Regole di archiviazione.
7. Nella scheda Tag, puoi gestire e creare tag per il sistema di analisi. Per ulteriori informazioni, consulta Tag per AWS Identity and Access Management risorse.

Eliminare un sistema di analisi degli accessi inutilizzati

Per eliminare un sistema di analisi degli accessi inutilizzati, utilizza la procedura seguente. Quando si elimina un sistema di analisi, le risorse non vengono più monitorate e non vengono generati nuovi risultati. Tutti i risultati generati dall'analizzatore vengono eliminati.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Sistema di analisi degli accessi, scegli Accesso inutilizzato.

3. In Sistema di analisi degli accessi, scegli Impostazioni dell'analizzatore.
4. Nella sezione Analizzatori, scegli il nome dell'analizzatore degli accessi non utilizzati da eliminare.
5. Scegli Elimina analizzatore.
6. Inserisci delete e scegli Elimina per confermare l'eliminazione del sistema di analisi.

Visualizzare il pannello di controllo dei risultati di Sistema di analisi degli accessi IAM

AWS Identity and Access Management Access Analyzer organizza i risultati di accesso esterni, interni e non utilizzati in una dashboard di riepilogo visivo. La dashboard consente di ottenere visibilità sull'uso efficace delle autorizzazioni su larga scala e di identificare gli account e AWS le risorse che richiedono attenzione. Puoi utilizzare la dashboard per esaminare i risultati per AWS organizzazione, account e tipo di risultato.

Per gli esiti relativi agli accessi esterni e interni:

- Il pannello di controllo evidenzia la suddivisione tra gli esiti relativi all'accesso pubblico, esiti relativi all'accesso esterno e esiti relativi all'accesso interno.
- Il pannello di controllo mostra una suddivisione dei risultati per tipo di risorsa.

Per i risultati degli accessi inutilizzati:

- La dashboard evidenzia Account AWS i risultati di accesso più inutilizzati.
- Il pannello di controllo mostra una suddivisione dei risultati per tipo.

Dopo aver creato un analizzatore degli accessi di qualsiasi tipo, Sistema di analisi degli accessi IAM aggiunge automaticamente nuovi esiti al pannello di controllo pertinente. Ciò consente di identificare e dare priorità alle aree con maggiori problemi di sicurezza.

Le dashboard di riepilogo offrono una visione di alto livello dei problemi di accesso rilevati da IAM Access Analyzer in tutto l'ambiente. AWS È quindi possibile approfondire i singoli risultati per analizzarli ulteriormente e intraprendere le azioni appropriate per risolverli.

Visualizzazione del pannello di controllo di riepilogo per gli analizzatori degli accessi esterni e interni

> **Note**
>
> Dopo aver creato o aggiornato un analizzatore, può essere necessario del tempo prima che la dashboard riassuntiva rifletta gli aggiornamenti dei risultati.

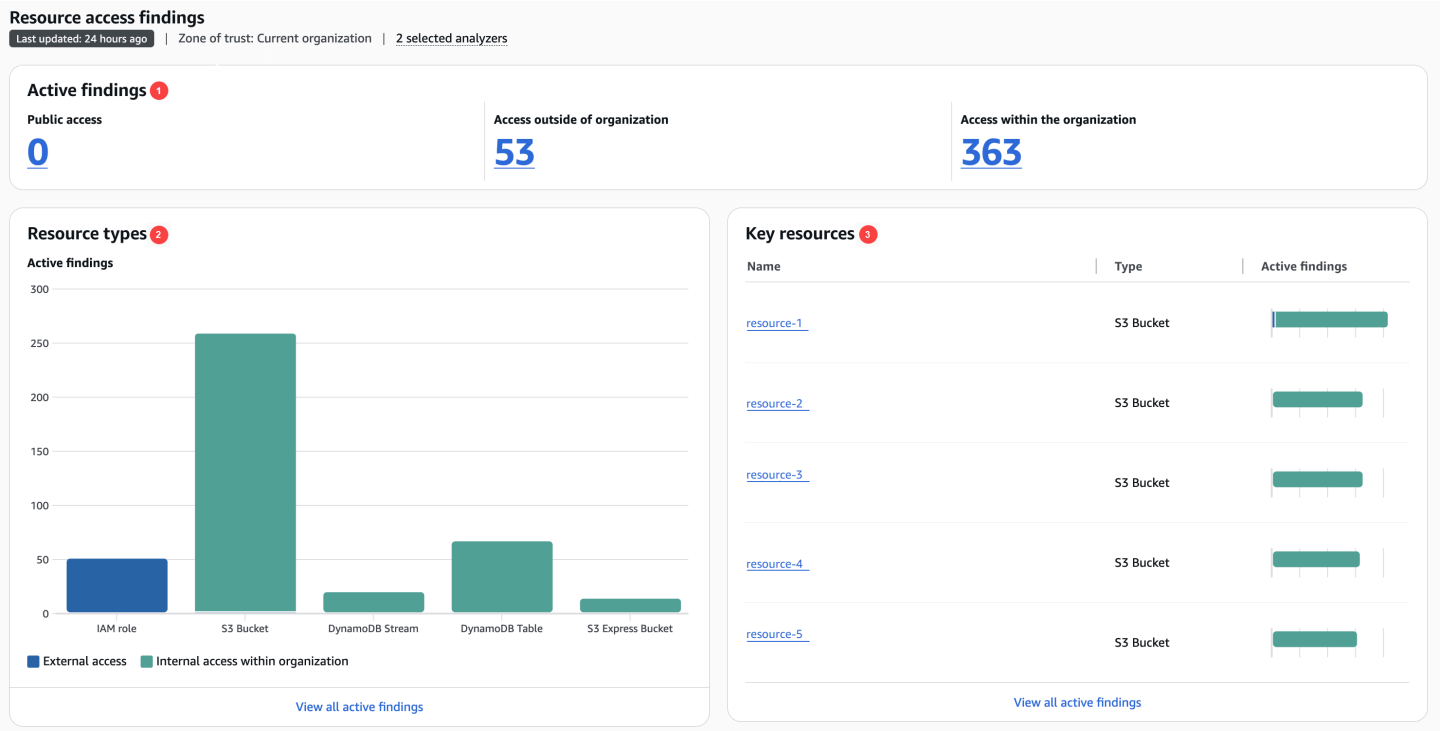
1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Sistema di analisi degli accessi. Viene visualizzata la finestra Riepilogo.
3. Scegli Seleziona analizzatori.
4. Nella finestra Seleziona analizzatori, scegli Organizzazione o Account per Zona di attendibilità.

   > **Note**
   >
   > Solo l'account AWS Organizations di gestione o l'amministratore delegato può scegliere Organizzazione come zona di fiducia.

5. Scegli analizzatori degli accessi esterni e interni dal menu a discesa Strumenti di analisi degli accessi alle risorse.

   > **Note**
   >
   > È possibile selezionare un massimo di un analizzatore degli accessi esterni e un massimo di un analizzatore degli accessi interni.

6. Scegliere Aggiorna. Appare un riepilogo degli esiti degli analizzatori degli accessi esterni e interni selezionati nella sezione Esiti relativi all'accesso alle risorse.



Nell'immagine precedente, il pannello di controllo degli esiti relativi alle risorse è visibile nella pagina Riepilogo.

1. La sezione Esiti attivi include il numero di esiti attivi relativi all'accesso pubblico, il numero di esiti attivi che forniscono l'accesso all'esterno dell'account o dell'organizzazione e il numero di esiti attivi degli accessi interni per gli analizzatori selezionati. Scegli un numero per elencare tutti i risultati attivi di ogni tipo.
2. La sezione Tipi di risorse include una suddivisione dei tipi di risorse con esiti attivi per gli analizzatori selezionati. Scegli Visualizza tutti gli esiti attivi per ottenere un elenco completo dgli esiti attivi per gli analizzatori selezionati.
3. La sezione Risorse chiave include un riepilogo delle risorse chiave con esiti attivi. Queste informazioni ti aiutano a dare priorità agli esiti relativi alle risorse aziendali critiche. Scegli Visualizza tutti gli esiti attivi per ottenere un elenco completo dgli esiti attivi per gli analizzatori selezionati.

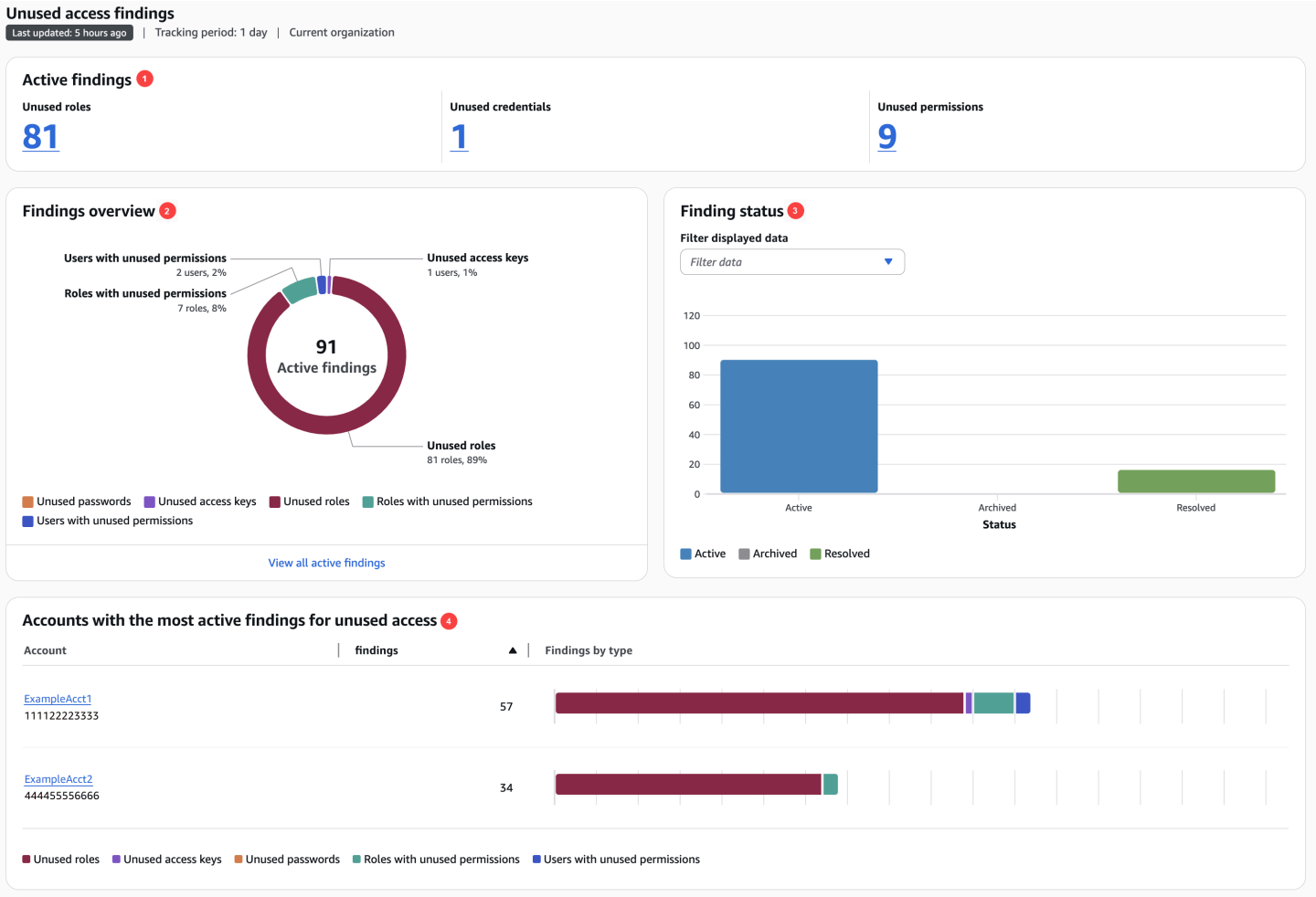
Visualizzazione del pannello di controllo di riepilogo per gli analizzatori degli accessi non utilizzati

> **Note**
>
> Dopo aver creato o aggiornato un analizzatore, in base al numero di utenti e ruoli, può essere necessario del tempo prima che la dashboard riassuntiva rifletta gli aggiornamenti dei risultati.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Sistema di analisi degli accessi. Viene visualizzata la finestra Riepilogo del sistema di analisi degli accessi.
3. Scegli Seleziona analizzatori.
4. Nella finestra Seleziona analizzatori, scegli Organizzazione o Account per Zona di attendibilità.

   > **Note**
   >
   > Solo l'account di AWS Organizations gestione o l'amministratore delegato può scegliere Organizzazione come zona di fiducia.

5. Scegli un analizzatore degli accessi non utilizzati dal menu a discesa Strumenti di analisi degli accessi non utilizzati.
6. Scegli Aggiorna riepilogo. Appare un riepilogo dei risultati dell'analizzatore degli accessi non utilizzati selezionato nella sezione Esiti di accesso inutilizzato.



Nell'immagine precedente, il pannello di controllo degli esiti relativi agli accessi non utilizzati è visibile nella pagina Riepilogo.

1. La sezione Risultati attivi include il numero di risultati attivi per ruoli , credenziali e autorizzazioni inutilizzati nell'account o nell'organizzazione. Le credenziali inutilizzate includono i risultati relativi alla chiave di accesso e alle password inutilizzate. Le autorizzazioni inutilizzate includono sia gli utenti che i ruoli con autorizzazioni inutilizzate. Scegli un numero per elencare tutti i risultati attivi di ogni tipo.
2. La sezione Panoramica dei risultati include una suddivisione del tipo di risultati attivi. Scegli Visualizza tutti i risultati attivi per ottenere un elenco completo dei risultati attivi per l'account o l'organizzazione dell'analizzatore.
3. La sezione Stato dell'esito include un'analisi dettagliata dello stato degli esiti (Attivo, Archiviato e Risolto) per l'account o l'organizzazione. Puoi selezionare gli stati degli esiti da visualizzare nel menu a discesa Filtra i dati visualizzati.

4. La sezione Account con il numero maggiore di risultati relativi agli accessi inutilizzati viene visualizzata solo se gli account selezionati dell'analizzatore degli accessi inutilizzati sono a livello di organizzazione. Include una suddivisione degli account dell'organizzazione con i risultati più attivi. Questo elenco non contiene tutti gli account dell'organizzazione. L'analizzatore potrebbe avere risultati attivi per altri account non elencati in questa sezione.

Rivedere i risultati di Sistema di analisi degli accessi IAM

Dopo aver abilitato Sistema di analisi degli accessi IAM, il passaggio successivo consiste nell'esaminare i risultati per determinare se l'accesso identificato nel risultato è intenzionale o meno. Puoi inoltre esaminare i risultati per determinare i risultati simili per l'accesso intenzionale e quindi creare una regola di archiviazione per archiviare tali risultati automaticamente. Puoi esaminare i risultati archiviati e risolti.

Devi esaminare tutti gli esiti del tuo account per determinare se l'accesso esterno, interno o non utilizzato è previsto e approvato. Se la condivisione identificata nell'esito è prevista, è possibile archiviare l'esito. Quando si archivia un risultato, lo stato viene modificato in Archiviato e il risultato viene rimosso dall'elenco dei risultati attivi. Il risultato non viene eliminato. Puoi visualizzare i risultati archiviati in qualsiasi momento. Elabora tutti i risultati nel tuo account fino a quando non hai più risultati attivi. Quando non hai più risultati, sai che eventuali nuovi risultati Attivi generati provengono da una modifica recente dell'ambiente.

> **Tip**
>
> Se hai un gran numero di risultati da esaminare, utilizza l'API di IAM Access Analyzer o i comandi AWS CLI per recuperare e gestire i risultati in modo programmatico.

Per esaminare gli esiti attivi per tutti i tipi di analizzatori degli accessi

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegliere Access Analyzer. Viene visualizzata la dashboard dei risultati.
3. Scegli Seleziona analizzatori.
4. Nella finestra Seleziona analizzatori, scegli al massimo un analizzatore degli accessi esterni e un massimo di un analizzatore degli accessi interni dal menu a discesa Analizzatori degli accessi alle risorse. Scegli un analizzatore degli accessi non utilizzati dal menu a discesa Strumenti di analisi degli accessi non utilizzati.

5. Scegli Aggiorna riepilogo. Nel pannello di controllo viene visualizzato un riepilogo degli esiti attivi per gli analizzatori degli accessi selezionati. Scegli un tipo di esito nelle sezioni Esiti relativi all'accesso alle risorse o Esiti di accesso inutilizzato per visualizzare tutti gli esiti attivi del tipo selezionato.

   Per ulteriori informazioni sulla visualizzazione della dashboard dei risultati, consulta Visualizzare il pannello di controllo dei risultati di Sistema di analisi degli accessi IAM.

> **Note**
>
> I risultati vengono visualizzati solo se si dispone dell'autorizzazione per visualizzare i risultati per l'analizzatore.

Esiti relativi agli accessi esterni e interni

> **Note**
>
> Sistema di analisi degli accessi IAM addebita l'analisi degli accessi interni in base al numero di risorse monitorate per Regione al mese. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi IAM.

1. In Sistema di analisi degli accessi, scegli Analisi delle risorse.
2. Scegli Seleziona analizzatori.
3. Nella finestra Seleziona analizzatori, scegli al massimo un analizzatore degli accessi esterni e un massimo di un analizzatore degli accessi interni dal menu a discesa Analizzatori degli accessi alle risorse.
4. Scegli Aggiorna riepilogo.

   La pagina Analisi delle risorse mostra i seguenti dettagli sulle risorse con esiti attivi per gli analizzatori degli accessi selezionati:

Nome

Il nome della risorsa con esiti attivi.

Tipo

Il tipo di risorsa.

Account proprietario

Questa colonna viene visualizzata solo se si utilizza un'organizzazione come zona di attendibilità per uno o più degli analizzatori selezionati. L'account dell'organizzazione proprietaria della risorsa riportata nel risultato.

Esiti attivi

Una rappresentazione visiva del numero e del tipo di esiti attivi per la risorsa. Passa il mouse sul campo per visualizzare ulteriori informazioni sugli esiti relativi alla risorsa.

Accesso pubblico

Indica se uno qualsiasi degli esiti della risorsa consente l'accesso pubblico.

Risultati degli accessi inutilizzati

> **Note**
>
> Sistema di analisi degli accessi AWS IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

1. In Sistema di analisi degli accessi, scegli Accesso non utilizzato.
2. Scegli Seleziona analizzatori.
3. Nella finestra Seleziona strumenti di analisi degli accessi, scegli un analizzatore degli accessi non utilizzati dal menu a discesa Sistema di analisi degli accessi non utilizzati.
4. Scegli Aggiorna riepilogo.

   La pagina Accesso non utilizzato visualizza i seguenti dettagli sull'entità IAM che ha generato gli esiti per l'analizzatore degli accessi selezionato:

ID risultato

L'ID univoco assegnato al risultato. Scegli l'ID risultato per visualizzare ulteriori dettagli sull'entità IAM che ha generato il risultato.

Tipo di risultato

Il tipo di risultato di accesso inutilizzato: chiave di accesso inutilizzata, password inutilizzata, autorizzazione inutilizzata o ruolo inutilizzato.

Entità IAM

L'entità IAM riportata nel risultato. Può trattarsi di un utente o di un ruolo IAM.

Account AWS ID

Questa colonna viene visualizzata solo se si imposta l'analizzatore per tutti gli Account AWS dell'organizzazione. All' Account AWS interno dell'organizzazione proprietaria dell'entità IAM riportata nel risultato.

Ultimo aggiornamento

L'ultima volta che l'entità IAM riportata nel risultato è stata aggiornata o, se non sono stati eseguiti aggiornamenti, quando l'entità è stata creata.

Stato

Lo stato del risultato (Attivo, Archiviato o Risolto).

Filtrare i risultati di Sistema di analisi degli accessi IAM

Il filtro predefinito per la pagina degli esiti visualizza tutti gli esiti. Per visualizzare tutti gli esiti, scegli Tutti dal menu a discesa Stato. Per visualizzare gli esiti archiviati, scegli la scheda Archiviato. Per visualizzare gli esiti risolti, scegli Risolto. Quando inizi a utilizzare per la prima volta Sistema di analisi degli accessi IAM, non ci sono risultati archiviati.

Utilizza i filtri per visualizzare solo i risultati che soddisfano i criteri di proprietà specificati. Per creare un filtro, seleziona la proprietà su cui filtrare, quindi scegli se la proprietà è uguale o contiene un valore e seleziona un valore di proprietà su cui filtrare.

Per l'elenco delle chiavi di filtro che puoi utilizzare per creare o aggiornare una regola di archivio, consulta Chiavi di filtro di Sistema di analisi degli accessi IAM.

Filtraggio delle risorse con esiti attivi

È possibile visualizzare e filtrare gli esiti attivi per risorsa per un massimo di un analizzatore degli accessi esterni e un massimo di un analizzatore degli accessi interni.

Per filtrare le risorse con esiti attivi

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Analisi delle risorse.
3. Per filtrare in base alla risorsa, digita il nome parziale o completo della risorsa nella casella di ricerca.
4. Nel menu a discesa Filtra il tipo di accesso, scegli il tipo di accesso:
   - Tutti i tipi: visualizza le risorse con tutti i tipi di esiti di accesso.
   - Accesso pubblico: visualizza solo le risorse con esiti di accesso pubblico.
   - Accesso esterno: visualizza solo le risorse con esiti di accesso esterno.
   - Accesso interno nell'ambito dell'organizzazione: visualizza solo le risorse con esiti di accesso interno.
5. Nel menu a discesa Filtra il tipo di risorsa, scegli un tipo di risorsa per visualizzare solo le risorse del tipo selezionato.

Filtraggio dei risultati degli accessi esterni

Per filtrare i risultati degli accessi esterni

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Impostazioni dell'analizzatore, quindi scegli l'analizzatore degli accessi esterni nella sezione Analizzatori.
3. Scegli Visualizza esiti.
4. Scegli la casella di ricerca per visualizzare un elenco di proprietà disponibili.
5. Scegliere la proprietà da utilizzare per filtrare i risultati visualizzati.
6. Scegliere il valore da corrispondere per la proprietà. Vengono visualizzati solo i risultati con tale valore presente nel risultato.

   Ad esempio, scegli Risorsa come proprietà e poi Risorsa:, digita il nome parziale o completo di un bucket, quindi premi Invio. Vengono visualizzati solo i risultati per il bucket che corrisponde ai criteri del filtro. Per creare un filtro che visualizzi solo i risultati per le risorse che consentono l'accesso pubblico, puoi scegliere la proprietà Accesso pubblico, quindi selezionare Accesso pubblico = e poi scegliere Accesso pubblico = true.

Puoi aggiungere altre proprietà per filtrare ulteriormente i risultati visualizzati. Quando aggiungi altre proprietà, vengono visualizzati solo i risultati che corrispondono a tutte le condizioni del filtro. La definizione di un filtro per visualizzare i risultati che corrispondono a una proprietà O a un'altra proprietà non è supportata. Scegli Cancella filtri per cancellare tutti i filtri definiti e visualizzare tutti i risultati con lo stato specificato per l'analizzatore.

Alcuni campi sono visibili solo quando si visualizzano i risultati di un analizzatore con un'organizzazione come zona di attendibilità.

Per la definizione dei filtri per l'accesso esterno sono disponibili le seguenti proprietà:

- Accesso pubblico: per filtrare in base ai risultati delle risorse che consentono l'accesso pubblico, usa il filtro Accesso pubblico, quindi scegli Accesso pubblico: true.
- Risorsa: per filtrare in base alla risorsa, digita il nome parziale o completo della risorsa.
- Tipo di risorsa: per filtrare in base al tipo di risorsa, scegli il tipo dall'elenco visualizzato.
- Account del proprietario della risorsa: utilizza questa proprietà per filtrare in base all'account dell'organizzazione proprietaria della risorsa riportata nel risultato.
- Limitazione della policy di controllo delle risorse: utilizza questa proprietà per filtrare in base al tipo di limitazione applicata da una policy di controllo delle risorse (RCP) di Organizations. Per ulteriori informazioni, consulta le politiche di controllo delle risorse (RCPs) nella Guida AWS Organizations per l'utente.
  - Valutazione RCP non riuscita: si è verificato un errore durante la valutazione dell'RCP.
  - Non applicabile: nessuna RCP limita questa risorsa o il principale. Sono incluse anche le risorse che non RCPs sono ancora supportate.
  - Applicabile: l'amministratore dell'organizzazione ha impostato delle limitazioni tramite una RCP che influiscono sulla risorsa o sul tipo di risorsa. Contatta l'amministratore dell'organizzazione per maggiori dettagli.
- AWS Account: utilizza questa proprietà per filtrare in base a Account AWS ciò a cui è concesso l'accesso nella sezione Principal di una dichiarazione politica. Per filtrare in base Account AWS, digita tutto o parte dell' Account AWS ID a 12 cifre oppure tutto o parte dell'ARN completo dell'account dell' AWS utente o del ruolo esterno che ha accesso alle risorse nell'account corrente.
- Utente canonico: per filtrare in base all'utente canonico, digita l'ID utente canonico come definito per i bucket Amazon S3. Per ulteriori informazioni, consulta ID account di AWS.
- Utente federato: per filtrare in base all'utente federato, digita l'ARN parziale o completo dell'identità federata. Per ulteriori informazioni, consulta Provider di identità e federazione.

- ID risultato: per filtrare in base all'ID risultato, digita l'ID parziale o completo.
- Errore: per filtrare in base al tipo di errore, scegli Accesso negato o Errore interno.
- ARN principale: utilizza questa proprietà per filtrare l'ARN del principale (utente, ruolo o gruppo IAM) utilizzato in una chiave di condizione aws:. PrincipalArn Per filtrare in base all'ARN principale, digita tutto o parte dell'ARN dell'utente, del ruolo o del gruppo IAM da un report esterno Account AWS riportato in un risultato.
- OrgID principale: per filtrare in base all'OrgID principale, digitare tutto o parte dell'ID dell'organizzazione associato ai principali esterni che appartengono all'organizzazione specificata come condizione AWS nel risultato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- Principal OrgPaths: per filtrare per Principal OrgPaths, digita tutto o parte dell'ID dell' AWS organizzazione o dell'unità organizzativa (OU) che consente l'accesso a tutti i responsabili esterni che sono membri dell'account dell'organizzazione o dell'unità organizzativa specificata come condizione nella politica. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- Account di origine: per filtrare in base all'account di origine, digita tutto o parte dell' Account AWS ID associato alle risorse, come utilizzato in alcune autorizzazioni interservizi di. AWS Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- ARN di origine: per filtrare in base all'ARN di origine, digita l'ARN parziale o completo specificato come condizione nel risultato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- IP di origine: per filtrare in base all'IP di origine, digita l'indirizzo IP parziale o completo che consente alle entità esterne di accedere alle risorse nell'account corrente quando utilizza l'indirizzo IP specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- Source OrgPaths: per filtrare in base a Source OrgPaths, digita una parte o tutto l'ID organizzazione associato alle risorse, come utilizzato in alcune autorizzazioni tra servizi in AWS. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- Origine OrgPaths: per filtrare in base all'origine OrgPaths, digita tutta o parte dell'unità organizzativa (OU) associata alle risorse, come utilizzato in alcune autorizzazioni interservizi di. AWS Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- VPC di origine: per filtrare in base al VPC di origine, digita l'ID parziale o completo del VPC che consente alle entità esterne di accedere alle risorse nell'account corrente quando utilizza il VPC specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.

- ARN VPC di origine: per filtrare in base all'ARN VPC di origine, digita tutto o parte dell'ARN VPC che consente alle entità esterne di accedere alle risorse nell'account corrente quando si utilizza il VPC specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- VPCE di origine: per filtrare in base al VPCE di origine, digita l'ID parziale o completo dell'endpoint VPC che consente alle entità esterne di accedere alle risorse nell'account corrente quando utilizza l'endpoint VPC specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- Account VPCE: per filtrare in base all'account VPCE, digita tutto o parte dell'ID a 12 cifre Account AWS che possiede le entità esterne dell'endpoint VPC e consente alle entità esterne l'accesso alle risorse. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- OrgID VPCE: per filtrare in base a OrgID VPCE, digita in modo parziale o completo l'ID dell'organizzazione che possiede le entità esterne dell'endpoint VPC e consente alle entità esterne di accedere alle risorse. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- VPCE OrgPaths: per filtrare in base a VPCE OrgPaths, digita tutta o parte dell'unità organizzativa (OU) proprietaria delle entità esterne dell'endpoint VPC e consente alle entità esterne l'accesso alle risorse. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- ID utente: per filtrare in base all'ID utente, digita tutto o parte dell'ID utente IAM di un utente esterno Account AWS a cui è consentito l'accesso alle risorse dell'account corrente. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- ID chiave KMS: per filtrare in base all'ID chiave KMS, digita l'ID parziale o completo della chiave KMS specificata come condizione per l'accesso agli oggetti Amazon S3 crittografati con AWS KMS nell'account corrente.
- Modalità della sessione: per filtrare in base alla modalità della sessione per i bucket di directory Amazon S3 (`ReadOnly` oppure `ReadWrite`), digita in modo parziale o completo la modalità di sessione. Per ulteriori informazioni, consulta il riferimento CreateSessionalle API di Amazon Simple Storage Service.
- Destinatari Google: per filtrare in base ai destinatari Google, digita l'ID parziale o completo dell'applicazione Google specificato come condizione per l'accesso del ruolo IAM nell'account corrente. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni IAM e AWS STS.
- Destinatari Cognito: per filtrare in base ai destinatari Amazon Cognito, digita l'ID parziale o completo del pool di identità Amazon Cognito specificato come condizione per l'accesso del ruolo IAM nell'account corrente. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni IAM e AWS STS.

- Account chiamante: l' Account AWS ID dell'account che possiede o contiene l'entità chiamante, ad esempio un ruolo IAM, un utente o un utente root dell'account. Viene utilizzato dai servizi che chiamano AWS KMS. Per filtrare in base all'account del chiamante, digita tutto o parte dell' Account AWS ID.
- ID app Facebook: per filtrare in base all'ID app Facebook, digita l'ID parziale o completo dell'applicazione Facebook (o l'ID del sito) specificato come condizione per consentire l'accesso con la federazione Facebook a un ruolo IAM nell'account corrente. Per saperne di più, consulta la sezione id in IAM e AWS STS condition context keys.
- ID app Amazon: per filtrare in base all'ID app Amazon, digita l'ID parziale o completo dell'applicazione Amazon (o l'ID del sito) specificato come condizione per consentire l'accesso della federazione Login with Amazon a un ruolo IAM nell'account corrente. Per saperne di più, consulta la sezione id in IAM e AWS STS condition context keys.
- Token di origine evento Lambda: per filtrare il token di origine evento Lambda passato con le integrazioni Alexa, digita la stringa parziale o completa del token.

Filtraggio degli esiti degli accessi esterni

Per filtrare i risultati degli accessi interni

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Impostazioni dell'analizzatore, quindi scegli l'analizzatore degli accessi interni nella sezione Analizzatori.
3. Scegli Visualizza esiti.
4. Scegli la casella di ricerca per visualizzare un elenco di proprietà disponibili.
5. Scegliere la proprietà da utilizzare per filtrare i risultati visualizzati.
6. Scegliere il valore da corrispondere per la proprietà. Vengono visualizzati solo i risultati con tale valore presente nel risultato.

   Ad esempio, scegli Risorsa come proprietà e poi Risorsa:, digita il nome parziale o completo di un bucket, quindi premi Invio. Vengono visualizzati solo i risultati per il bucket che corrisponde ai criteri del filtro.

Puoi aggiungere altre proprietà per filtrare ulteriormente i risultati visualizzati. Quando aggiungi altre proprietà, vengono visualizzati solo i risultati che corrispondono a tutte le condizioni del filtro. La definizione di un filtro per visualizzare i risultati che corrispondono a una proprietà O a un'altra

proprietà non è supportata. Scegli Cancella filtri per cancellare tutti i filtri definiti e visualizzare tutti i risultati con lo stato specificato per l'analizzatore.

Alcuni campi sono visibili solo quando si visualizzano i risultati di un analizzatore con un'organizzazione come zona di attendibilità.

I campi seguenti sono mostrati solo quando si visualizzano gli esiti per un analizzatore che monitora gli accessi non utilizzati:

- Risorsa: per filtrare in base alla risorsa, digita il nome parziale o completo della risorsa.
- Tipo di risorsa: per filtrare in base al tipo di risorsa, scegli il tipo dall'elenco visualizzato.
- Account del proprietario della risorsa: utilizza questa proprietà per filtrare in base all'account dell'organizzazione proprietaria della risorsa riportata nel risultato.
- ID risultato: per filtrare in base all'ID risultato, digita parte o tutto l'ID risultato.

Filtraggio di risultati degli accessi inutilizzati

Per filtrare i risultati degli accessi inutilizzati

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Accesso inutilizzato, quindi seleziona l'analizzatore dal menu a discesa Visualizza analizzatore.
3. Scegli la casella di ricerca per visualizzare un elenco di proprietà disponibili.
4. Scegliere la proprietà da utilizzare per filtrare i risultati visualizzati.
5. Scegliere il valore da corrispondere per la proprietà. Vengono visualizzati solo i risultati con tale valore presente nel risultato.

   Ad esempio, scegli Tipo di esito come proprietà, quindi scegli Tipo di esito =, quindi Tipo di esito = Ruolo non utilizzato. Vengono visualizzati solo gli esiti con un tipo di Ruolo non utilizzato.

Puoi aggiungere altre proprietà per filtrare ulteriormente i risultati visualizzati. Quando aggiungi altre proprietà, vengono visualizzati solo i risultati che corrispondono a tutte le condizioni del filtro. La definizione di un filtro per visualizzare i risultati che corrispondono a una proprietà O a un'altra proprietà non è supportata. Scegli Cancella filtri per cancellare tutti i filtri definiti e visualizzare tutti i risultati con lo stato specificato per l'analizzatore.

I campi seguenti sono visualizzati solo quando si visualizzano i risultati per un sistema di analisi che monitora l'accesso inutilizzato:

- Tipo di risultato: per filtrare in base al tipo di risultato, filtra per Tipo di risultato, quindi scegli il tipo di risultato.
- Risorsa: per filtrare in base alla risorsa, digita il nome parziale o completo della risorsa.
- Tipo di risorsa: per filtrare in base al tipo di risorsa, scegli il tipo dall'elenco visualizzato.
- Account del proprietario della risorsa: utilizza questa proprietà per filtrare in base all'account dell'organizzazione proprietaria della risorsa riportata nel risultato.
- ID risultato: per filtrare in base all'ID risultato, digita parte o tutto l'ID risultato.

Archiviare i risultati di Sistema di analisi degli accessi IAM

Quando ricevi un risultato per l'accesso a una risorsa che è intenzionale, puoi archiviarlo. Ad esempio, un esito di accesso interno o esterno per un bucket Amazon S3 al quale viene effettuato l'accesso per flussi di lavoro approvati oppure l'esito di un accesso non utilizzato per una chiave di accesso che potrebbe essere ancora necessaria. Quando si archivia un risultato, questo viene cancellato dall'elenco dei risultati attivi. I risultati archiviati non vengono eliminati. Puoi filtrare la pagina Risultati per visualizzare i risultati archiviati e annullarne l'archiviazione in qualsiasi momento.

Per archiviare i risultati dalla pagina Findings (Risultati)

1. Selezionare la casella di controllo accanto a uno o più risultati da archiviare.
2. Scegli Operazioni, quindi seleziona Archivia.

   Viene visualizzata una richiesta di conferma nella parte superiore dello schermo.

Per archiviare i risultati dalla pagina Dettagli risultati

1. Scegliere Finding ID (ID risultato) del risultato da archiviare.
2. Scegliere Archive (Archivia).

   Viene visualizzata una richiesta di conferma nella parte superiore dello schermo.

Per annullare l'archiviazione dei risultati, ripetere i passaggi precedenti, ma scegliere Unarchive (Annulla archiviazione) anziché Archive (Archivia). Quando si annulla l'archiviazione di un risultato, lo stato diventa Attivo.

Risolvere i risultati di Sistema di analisi degli accessi IAM

Risoluzione degli esiti relativi alle risorse

Per risolvere gli esiti relativi agli accessi esterni e interni generati dall'accesso non intenzionale, devi modificare l'istruzione della policy per rimuovere le autorizzazioni che consentono l'accesso alla risorsa identificata.

Ad esempio, per i risultati relativi ai bucket Amazon S3, utilizza la console Amazon S3 per configurare le autorizzazioni sul bucket.

Per i ruoli IAM, utilizza la console IAM per modificare la policy di attendibilità per il ruolo IAM elencato.

Per altre risorse supportate, utilizza la console per modificare le istruzioni della policy che hanno portato a un risultato generato.

Dopo aver apportato una modifica per risolvere un esito relativo a una risorsa, ad esempio la modifica di una policy applicata a un ruolo IAM, Sistema di analisi degli accessi IAM esegue nuovamente la scansione della risorsa. Se l'accesso alla risorsa viene rimosso, lo stato dell'esito viene modificato in Risolto. Il risultato verrà quindi visualizzato nell'elenco di risultati risolti anziché nell'elenco dei risultati attivi.

> **Note**
>
> Questo non si applica ai risultati Errore. Se Sistema di analisi degli accessi IAM non è in grado di accedere a una risorsa, genera un risultato di errore. Se risolvi il problema che ha impedito a Sistema di analisi degli accessi IAM di analizzare la risorsa, il risultato dell'errore verrà rimosso completamente invece di passare a un risultato risolto. Per ulteriori informazioni, consulta Esiti di errore di Sistema di analisi degli accessi IAM.

Se le modifiche applicate hanno determinato l'accesso esterno o interno alla risorsa ma in un modo diverso, ad esempio con un principale differente o per un'autorizzazione diversa, Sistema di analisi degli accessi IAM risolve l'esito originale e genera un nuovo esito Attivo. Se le modifiche apportate hanno determinato errori interni o errori di accesso negato, tutti gli esiti attivi non relativi all'accesso specifico della risorsa vengono risolti e viene generato un nuovo esito di errore.

> **Note**
>
> Per gli analizzatori degli accessi esterni, dopo la modifica di una policy possono occorrere fino a 30 minuti affinché Sistema di analisi degli accessi IAM analizzi nuovamente la risorsa e aggiorni l'esito.
> Per gli analizzatori degli accessi interni, possono occorrere diversi minuti oppure ore affinché Sistema di analisi degli accessi IAM analizzi nuovamente la risorsa e aggiorni l'esito.
> Strumento di analisi degli accessi IAM esegue automaticamente una nuova scansione di tutte le policy ogni 24 ore.
> I risultati risolti vengono eliminati 90 giorni dopo l'ultimo aggiornamento dello stato del risultato.

Risoluzione dei risultati degli accessi inutilizzati

Il Sistema di analisi degli accessi IAM fornisce i passaggi consigliati per risolvere i risultati del sistema di analisi degli accessi inutilizzati in base al tipo di risultato.

Dopo aver apportato una modifica per risolvere un risultato di accesso inutilizzato, lo stato del risultato viene modificato in Risolto alla successiva esecuzione dell'analizzatore di accessi inutilizzati. Il risultato non viene più visualizzato nella lista dei risultati attivi, bensì nella lista dei risultati risolti. Se si apporta una modifica che risolve solo parzialmente un risultato di accesso inutilizzato, il risultato esistente viene modificato in Risolto ma viene generato un nuovo risultato. Ad esempio, se si rimuovono solo alcune delle autorizzazioni inutilizzate di un risultato, ma non tutte.

Sistema di analisi degli accessi IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

Risoluzione dei risultati delle autorizzazioni inutilizzate

Per i risultati delle autorizzazioni non utilizzati, il Sistema di analisi degli accessi IAM può consigliare le policy da rimuovere da un utente o ruolo IAM e fornire nuove policy per sostituire le policy di autorizzazione esistenti. Il suggerimento delle policy non è supportato per gli scenari seguenti:

- Il risultato delle autorizzazioni inutilizzate si riferisce a un utente IAM che fa parte di un gruppo di utenti.
- Il risultato delle autorizzazioni inutilizzate riguarda un ruolo IAM per il Centro identità IAM.

- Il risultato delle autorizzazioni inutilizzate include una policy di autorizzazioni esistente che include l'elemento `notAction`.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Accesso inutilizzato.
3. Scegli un risultato con Tipo di risultato impostato su Autorizzazioni inutilizzate.
4. Nella sezione Suggerimenti, se ci sono policy elencate nella colonna Policy consigliata, scegli Anteprima della policy per visualizzare la policy esistente con la policy consigliata per sostituire la policy esistente. Se sono presenti più criteri consigliati, è possibile scegliere Criterio successivo e Criterio precedente per visualizzare ogni criterio esistente e consigliato.
5. Scegli Scarica JSON per scaricare un file .zip con i file JSON di tutte le policy consigliate.
6. Crea e collega le policy consigliate all'utente o al ruolo IAM. Per ulteriori informazioni, consulta Modifica delle autorizzazioni per un utente (console) e Modifica di una policy di autorizzazione di un ruolo (console).
7. Rimuovi le policy elencate nella colonna Policy di autorizzazione esistente dall'utente o dal ruolo IAM. Per ulteriori informazioni, vedere Rimuovere le autorizzazioni da un utente (console) e Modifica dei criteri di autorizzazione dei ruoli (console).

Risoluzione dei risultati relativi ai ruoli inutilizzati

Per i risultati dei ruoli inutilizzati, il Sistema di analisi degli accessi IAM consiglia di eliminare il ruolo IAM inutilizzato.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Accesso inutilizzato.
3. Scegli un risultato con Tipo di risultato impostato su Ruolo inutilizzato.
4. Nella sezione Suggerimenti, esamina i dettagli del ruolo IAM.
5. Elimina il ruolo IAM. Per ulteriori informazioni, consulta Eliminazione di un ruolo IAM (console).

Risoluzione dei risultati delle chiavi di accesso inutilizzate

Per i risultati delle chiavi di accesso inutilizzate, il Sistema di analisi degli accessi IAM consiglia di disattivare o eliminare la chiave di accesso inutilizzata.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Scegli Accesso inutilizzato.
3. Scegli un risultato con Tipo di risultato impostato su Chiavi di accesso inutilizzate.
4. Nella sezione Suggerimenti, esamina i dettagli della chiave di accesso.
5. Disattiva o elimina la chiave di accesso. Per ulteriori informazioni, consulta Gestione delle chiavi di accesso (console).

Risoluzione dei risultati relativi alle password inutilizzate

Per i risultati delle password inutilizzatei, il Sistema di analisi degli accessi IAM consiglia di eliminare la password inutilizzata per l'utente IAM.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Accesso inutilizzato.
3. Scegli un risultato con Tipo di risultato impostato su Password inutilizzata.
4. Nella sezione Suggerimenti, esamina i dettagli dell'utente IAM.
5. Elimina la password per l'utente IAM. Per ulteriori informazioni, consulta Creazione, modifica o eliminazione di una password utente IAM (console).

Esiti di errore di Sistema di analisi degli accessi IAM

Quando Strumento di analisi degli accessi IAM analizza le risorse, solitamente genera degli esiti che mostrano chi ha accesso alle tue risorse. Tuttavia, in alcuni casi, l'analizzatore potrebbe riscontrare dei problemi che gli impediscono di completare l'analisi. In questi casi, Sistema di analisi degli accessi IAM genera invece esiti di errore.

Gli esiti di errore indicano che Strumento di analisi degli accessi IAM non è riuscito a completare l'analisi per una risorsa specifica o per una specifica coppia principale-risorsa. Questi esiti aiutano a identificare le risorse che potrebbero richiedere attenzione per garantire un'analisi corretta.

Esiti di errore relativi agli accessi esterni

Gli analizzatori degli accessi esterni, che identificano le risorse condivise all'esterno dell'account o dell'organizzazione, possono generare due tipi di esiti di errore:

- INTERNAL_ERROR: indica che Strumento di analisi degli accessi IAM ha riscontrato un problema interno durante l'analisi della risorsa. Questo potrebbe essere dovuto a limitazioni dei servizi o a problemi temporanei.

```
{
 "findingDetails": [
  {
   "externalAccessDetails": {}
  }
 ],
 "resource": "arn:aws:iam::941407043048:role/TestAccessAnalyzer",
 "status": "ACTIVE",
 "error": "INTERNAL_ERROR",
 "createdAt": "2022-07-14T01:31:43.085000+00:00",
 "resourceType": "AWS::IAM::Role",
 "findingType": "ExternalAccess",
 "resourceOwnerAccount": "941407043048",
 "analyzedAt": "2025-03-19T06:51:46.109000+00:00",
 "id": "4b035c7d-b7d2-40e4-a6c3-9887d1a995df",
 "updatedAt": "2022-07-14T01:31:43.085000+00:00"
}
```


- ACCESS_DENIED: indica che Strumento di analisi degli accessi IAM non dispone delle autorizzazioni necessarie per analizzare la risorsa. Questo accade in genere quando al ruolo collegato ai servizi (SLR) per Sistema di analisi degli accessi IAM viene negato l'accesso alla risorsa.


```
{
 "findingDetails": [
  {
   "externalAccessDetails": {}
  }
 ],
 "resource": "arn:aws:kms:us-west-2:941407043048:key/01cae123-
b7f2-4488-9a05-0070a072ea2c",
 "status": "ACTIVE",
 "error": "ACCESS_DENIED",
 "createdAt": "2022-07-14T01:31:43.104000+00:00",
 "resourceType": "AWS::KMS::Key",
 "findingType": "ExternalAccess",
 "resourceOwnerAccount": "941407043048",
 "analyzedAt": "2025-03-19T06:51:46.090000+00:00",
 "id": "7ef6f04a-9d2c-4038-9cc0-2a5f00a4d8f8",
 "updatedAt": "2022-07-14T01:31:43.104000+00:00"
}
```

Esiti di errore relativi agli accessi interni

Gli analizzatori degli accessi interni, che identificano l'accesso all'interno dell'account o dell'organizzazione, possono generare quattro tipi di esiti di errore:

- PRINCIPAL_LIMIT_EXCEEDED: viene generato quando più di 3.000 principali hanno accesso a una risorsa critica. Questo errore consente di identificare le risorse con un accesso troppo ampio che potrebbe essere necessario limitare.

  Se si apportano modifiche alla risorsa o ai principali dell'ambiente che portano il numero di principali al di sotto del limite, l'analizzatore genererà esiti normali durante la scansione successiva e l'esito di errore verrà contrassegnato come risolto.

```
{
 "id": "efec28fe-b304-412f-af0f-704d0d70c79c",
 "status": "ACTIVE",
 "error": "PRINCIPAL_LIMIT_EXCEEDED",
 "resource": "arn:aws:s3:::critical-data",
 "resourceType": "AWS::S3::Bucket",
 "resourceOwnerAccount": "111122223333",
 "createdAt": "2023-11-30T00:56:56.437000+00:00",
 "analyzedAt": "2024-03-06T04:11:54.406000+00:00",
 "updatedAt": "2023-11-30T00:56:56.437000+00:00",
 "findingType": "InternalAccess",
 "findingDetails": [
  {
   "internalAccessDetails": {}
  }
 ]
}
```

- Errori a livello di risorsa (INTERNAL_ERROR o ACCESS_DENIED): simili agli errori di accesso esterno, indicano che l'analizzatore non è riuscito ad analizzare una risorsa specifica a causa di problemi interni o problemi di autorizzazione. Quando si verifica un errore a livello di risorsa, l'analizzatore genera un singolo esito di errore per la risorsa anziché gli esiti normali.

```
{
 "id": "efec28fe-b304-412f-af0f-704d0d70c79c",
 "status": "ACTIVE",
 "error": "INTERNAL_ERROR", // can be INTERNAL_ERROR or ACCESS_DENIED
 "resource": "arn:aws:s3:::critical-data",
 "resourceType": "AWS::S3::Bucket",
```

```
 "resourceOwnerAccount": "111122223333",
 "createdAt": "2023-11-30T00:56:56.437000+00:00",
 "analyzedAt": "2024-03-06T04:11:54.406000+00:00",
 "updatedAt": "2023-11-30T00:56:56.437000+00:00",
 "findingType": "InternalAccess",
 "findingDetails": [
  {
   "internalAccessDetails": {}
  }
 ]
}
```

- Errori a livello di principale (INTERNAL_ERROR o ACCESS_DENIED): indica che l'analizzatore non è riuscito ad analizzare l'accesso di un principale specifico a una risorsa specifica. A differenza degli errori a livello di risorsa, una risorsa può avere esiti normali per alcuni principali ed esiti di errore per altri principali.

```
{
 "id": "efec28fe-b304-412f-af0f-704d0d70c79c",
 "status": "ACTIVE",
 "error": "INTERNAL_ERROR", // can be INTERNAL_ERROR or ACCESS_DENIED
 "resource": "arn:aws:s3:::critical-data",
 "resourceType": "AWS::S3::Bucket",
 "resourceOwnerAccount": "111122223333",
 "createdAt": "2023-11-30T00:56:56.437000+00:00",
 "analyzedAt": "2024-03-06T04:11:54.406000+00:00",
 "updatedAt": "2023-11-30T00:56:56.437000+00:00",
 "findingType": "InternalAccess",
 "findingDetails": [
  {
   "internalAccessDetails": {
    "principal": {
     "AWS": "arn:aws:iam::111122223333:role/MyRole_1"
    },
    "principalOwnerAccount": "111122223333",
    "principalType": "IAM_ROLE",
    "accessType": "INTRA_ACCOUNT"
   }
  }
 ]
}
```

- PRINCIPAL_ERRORS_LIMIT_EXCEEDED: viene generato quando vengono rilevati troppi errori a livello di principale per una singola risorsa. Si tratta di un esito di errore a livello di risorsa che può comparire insieme agli esiti normali della medesima risorsa.


```
{
 "id": "efec28fe-b304-412f-af0f-704d0d70c79c",
 "status": "ACTIVE",
 "error": "PRINCIPAL_ERRORS_LIMIT_EXCEEDED",
 "resource": "arn:aws:s3:::critical-data",
 "resourceType": "AWS::S3::Bucket",
 "resourceOwnerAccount": "111122223333",
 "createdAt": "2023-11-30T00:56:56.437000+00:00",
 "analyzedAt": "2024-03-06T04:11:54.406000+00:00",
 "updatedAt": "2023-11-30T00:56:56.437000+00:00",
 "findingType": "InternalAccess",
 "resourceControlPolicyRestriction": "NOT_APPLICABLE",
 "serviceControlPolicyRestriction": "NOT_APPLICABLE",
 "findingDetails": [
  {
   "internalAccessDetails": {}
  }
 ]
}
```


Risoluzione degli esiti di errore

Se risolvi il problema che ha impedito a Sistema di analisi degli accessi IAM di analizzare la risorsa, il risultato dell'errore verrà rimosso completamente invece di passare a un risultato risolto.

Per risolvere gli esiti di errore, considera i seguenti approcci in base al tipo di errore:

- Per gli errori ACCESS_DENIED, verifica che il ruolo collegato ai servizi di Strumento di analisi degli accessi IAM disponga delle autorizzazioni necessarie per accedere alla risorsa.
- Per gli errori PRINCIPAL_LIMIT_EXCEEDED, esamina le policy di accesso della risorsa e valuta la possibilità di limitare l'accesso a un numero inferiore di principali.
- Per i risultati di INTERNAL_ERROR, potrebbe essere necessario attendere un ciclo di analisi successivo o contattare l' AWS assistenza se il problema persiste.
- Per PRINCIPAL_ERRORS_LIMIT_EXCEEDED, rivedi e potenzialmente semplifica i modelli di accesso per la risorsa interessata.

Dopo aver apportato modifiche per risolvere i problemi sottostanti, Strumento di analisi degli accessi IAM tenterà di analizzare nuovamente le risorse durante il ciclo di scansione successivo.

Tipi di risorse di Sistema di analisi degli accessi IAM supportate per gli accessi esterni e interni

Per gli analizzatori di accesso esterni e interni, IAM Access Analyzer analizza le politiche basate sulle risorse applicate alle risorse nella regione in cui è stato abilitato IAM Access Analyzer. AWS Vengono analizzate solo le policy basate sulle risorse. Per informazioni su come Sistema di analisi degli accessi IAM genera gli esiti per ciascun tipo di risorsa, consulta le informazioni sul tipo di risorsa.

> **Note**
>
> I tipi di risorse supportati elencati si riferiscono agli analizzatori degli accessi interni ed esterni. Gli analizzatori degli accessi interni non supportano tutti i tipi di risorse supportati dagli analizzatori degli accessi esterni. Gli analizzatori degli accessi inutilizzati supportano solo utenti e ruoli IAM. Per ulteriori informazioni, consulta Come funzionano i risultati di Sistema di analisi degli accessi IAM.

Tipi di risorse supportati per gli accessi esterni

- Bucket Amazon Simple Storage Service
- Bucket di directory di Amazon Simple Storage Service
- AWS Identity and Access Management ruoli
- AWS Key Management Service chiavi
- AWS Lambda funzioni e livelli
- Code Amazon Simple Queue Service
- AWS Secrets Manager segreti
- Argomenti su Amazon Simple Notification Service
- Volumi e snapshot di Amazon Elastic Block Store
- Amazon Relational Database Service
- Snapshot di cluster di database di Amazon Relational Database Service
- Repository di Amazon Elastic Container Registry

- File system di Amazon Elastic File System
- Flussi Amazon DynamoDB
- Tabelle Amazon DynamoDB

Tipi di risorse supportati per gli accessi interni

- Bucket Amazon Simple Storage Service
- Bucket di directory di Amazon Simple Storage Service
- Amazon Relational Database Service
- Snapshot di cluster di database di Amazon Relational Database Service
- Flussi Amazon DynamoDB
- Tabelle Amazon DynamoDB

Bucket Amazon Simple Storage Service

Quando Sistema di analisi degli accessi IAM analizza i bucket Amazon S3 per gli analizzatori degli accessi esterni, genera un esito quando la policy di bucket Amazon S3, una lista di controllo degli accessi (ACL) o un punto di accesso, incluso un punto di accesso multi-Regione, applicato a un bucket concede l'accesso a un'entità esterna. Un'entità esterna è un'entità principale o un'altra entità che puoi utilizzare per creare un filtro che non si trova all'interno della zona di attendibilità. Ad esempio, se la policy di un bucket concede l'accesso a un altro account o consente l'accesso pubblico, Sistema di analisi degli accessi IAM genera un risultato. Tuttavia, se abiliti Block Public Access (Blocca accesso pubblico) nel bucket, puoi bloccare l'accesso a livello di account o bucket.

Per gli analizzatori degli accessi interni, Sistema di analisi degli accessi IAM genera un esito quando un principale (utente o ruolo) all'interno dell'organizzazione o dell'account ha accesso a un bucket Amazon S3 specificato.

> **Note**
>
> Sistema di analisi degli accessi IAM non analizza la policy dei punti di accesso associata ai punti di accesso multi-account perché il punto di accesso e la relativa policy sono esterni all'account dell'analizzatore. Sistema di analisi degli accessi IAM genera un risultato pubblico quando un bucket delega l'accesso a un punto di accesso multi-account e il blocco dell'accesso pubblico non è abilitato sul bucket o sull'account. Quando abiliti l'opzione di

blocco dell'accesso pubblico, il rilevamento pubblico viene risolto e Sistema di analisi degli accessi IAM genera un risultato tra account per il punto di accesso multi-account.

Le impostazioni per il blocco dell'accesso pubblico di Amazon S3 sostituiscono le policy applicate al bucket. Le impostazioni sovrascrivono anche le policy applicate ai punti di accesso del bucket. Sistema di analisi degli accessi IAM analizza le impostazioni del blocco dell'accesso pubblico a livello di bucket ogni volta che cambia una policy. Tuttavia, valuta le impostazioni del blocco dell'accesso pubblico a livello di account solo una volta ogni 6 ore. Ciò significa che Sistema di analisi degli accessi IAM potrebbe non generare o risolvere un risultato per l'accesso pubblico a un bucket per un massimo di 6 ore. Ad esempio, se hai una policy del bucket che consente l'accesso pubblico, Sistema di analisi degli accessi IAM genera un risultato per tale accesso. Se quindi abiliti il blocco dell'accesso pubblico per bloccare tutti gli accessi pubblici al bucket a livello di account, Sistema di analisi degli accessi IAM non risolve il risultato per la policy del bucket per un massimo di 6 ore, anche se tutti gli accessi pubblici al bucket sono bloccati. La risoluzione dei dati pubblici relativi ai punti di accesso multi-account può inoltre richiedere fino a 6 ore dopo l'abilitazione del blocco dell'accesso pubblico a livello di account. Le modifiche a una policy di controllo delle risorse (RCP) senza una modifica della policy del bucket non attivano una nuova scansione del bucket riportato nel risultato. In questo caso, Sistema di analisi degli accessi IAM analizza la policy nuova o aggiornata durante la scansione periodica successiva che avviene entro 24 ore.

Per un punto di accesso multi-regione, Sistema di analisi degli accessi IAM utilizza una policy stabilita per la generazione dei risultati. Sistema di analisi degli accessi IAM valuta le modifiche apportate ai punti di accesso multi-regione una volta ogni 6 ore. Ciò significa che Sistema di analisi degli accessi IAM non genera né risolve un risultato per un massimo di 6 ore, anche se viene creato o eliminato un punto di accesso multi-regione o se ne aggiorna la policy.

Bucket di directory di Amazon Simple Storage Service

I bucket di directory di Amazon S3 organizzano i dati in modo gerarchico in directory, a differenza della struttura di archiviazione piatta dei bucket per uso generico, consigliata per carichi di lavoro o applicazioni critici in termini di prestazioni. Per gli analizzatori degli accessi esterni, Sistema di analisi degli accessi IAM analizza la policy di bucket della directory, incluse le istruzioni di condizione in una policy, che consentono a un'entità esterna di accedere a un bucket di directory.

Per gli analizzatori degli accessi esterni, Sistema di analisi degli accessi IAM genera un esito quando un principale (utente o ruolo) all'interno dell'organizzazione o dell'account ha accesso a un bucket di una directory Amazon S3 specificato.

I bucket di directory Amazon S3 supportano anche i punti di accesso, che applicano autorizzazioni e controlli di rete distinti per tutte le richieste effettuate al bucket di directory tramite il punto di accesso. Ogni punto di accesso può avere una policy del punto di accesso che funziona insieme alla policy di bucket collegata al bucket di directory sottostante. Con i punti di accesso per i bucket di directory, puoi limitare l'accesso a prefissi specifici, operazioni API o un cloud privato virtuale (VPC).

> **Note**
>
> Sistema di analisi degli accessi IAM non analizza la policy dei punti di accesso associata ai punti di accesso multi-account perché il punto di accesso e la relativa policy sono esterni all'account dell'analizzatore. Sistema di analisi degli accessi IAM genera un risultato pubblico quando un bucket delega l'accesso a un punto di accesso multi-account e il blocco dell'accesso pubblico non è abilitato sul bucket o sull'account. Quando abiliti l'opzione di blocco dell'accesso pubblico, il rilevamento pubblico viene risolto e Sistema di analisi degli accessi IAM genera un risultato tra account per il punto di accesso multi-account.

Per ulteriori informazioni sui bucket di directory di Amazon S3, consulta Working with directory buckets nella Guida per l'utente di Amazon Simple Storage Service.

AWS Identity and Access Management ruoli

Per i ruoli IAM, Sistema di analisi degli accessi IAM analizza le policy di attendibilità. In una policy di attendibilità per il ruolo si definiscono le entità attendibili per assumere il ruolo. Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo in IAM. Sistema di analisi degli accessi IAM genera i risultati per i ruoli all'interno della zona di attendibilità a cui può accedere un'entità esterna che si trova al di fuori della zona di attendibilità.

> **Note**
>
> Un ruolo IAM è una risorsa globale. Se una policy di attendibilità per il ruolo concede l'accesso a un'entità esterna, Sistema di analisi degli accessi IAM genera un risultato in ogni regione abilitata.

AWS Key Management Service chiavi

Infatti AWS KMS keys, IAM Access Analyzer analizza le politiche e le concessioni chiave applicate a una chiave. Sistema di analisi degli accessi IAM genera un risultato se una policy di chiave o una

concessione consente a un'entità esterna di accedere alla chiave. Ad esempio, se utilizzi la chiave kms: CallerAccount condition in un'informativa politica per consentire l'accesso a tutti gli utenti di un AWS account specifico e specifichi un account diverso dall'account corrente (la zona di fiducia per l'analizzatore corrente), IAM Access Analyzer genera un risultato. Per ulteriori informazioni sulle chiavi di AWS KMS condizione nelle dichiarazioni delle politiche IAM, consulta AWS KMS Condition Keys.

Quando Sistema di analisi degli accessi IAM analizza una chiave KMS, legge i metadati della chiave, ad esempio la policy della chiave e l'elenco delle concessioni. Se la policy della chiave non consente al ruolo di Sistema di analisi degli accessi IAM di leggere i metadati della chiave, viene generato un errore di accesso negato. Ad esempio, se l'istruzione della policy di esempio seguente è l'unica policy applicata a una chiave, viene generato un errore di accesso negato in Sistema di analisi degli accessi IAM:

```
{
    "Sid": "Allow access for Key Administrators",
    "Effect": "Allow",
    "Principal": {
       "AWS": "arn:aws:iam::111122223333:role/Admin"
    },
    "Action": "kms:*",
    "Resource": "*"
}
```

Poiché questa istruzione consente solo al ruolo denominato Admin dell' AWS account 111122223333 di accedere alla chiave, viene generata una ricerca di errore di accesso negato perché IAM Access Analyzer non è in grado di analizzare completamente la chiave. Un risultato di errore viene visualizzato in rosso nella tabella Findings (Risultati). Il risultato è simile al seguente:

```
{
    "error": "ACCESS_DENIED",
    "id": "12345678-1234-abcd-dcba-111122223333",
    "analyzedAt": "2019-09-16T14:24:33.352Z",
    "resource": "arn:aws:kms:us-
west-2:1234567890:key/1a2b3c4d-5e6f-7a8b-9c0d-1a2b3c4d5e6f7g8a",
    "resourceType": "AWS::KMS::Key",
    "status": "ACTIVE",
    "updatedAt": "2019-09-16T14:24:33.352Z"
}
```

Quando crei una chiave KMS, le autorizzazioni concesse per accedere alla chiave dipendono dalla modalità di creazione della chiave. Se viene visualizzato un errore di tipo Accesso negato per una risorsa chiave, applica la seguente istruzione della policy alla risorsa chiave per concedere a Sistema di analisi degli accessi IAM l'autorizzazione per accedere alla chiave.

```
{
    "Sid": "Allow IAM Access Analyzer access to key metadata",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/aws-service-role/access-
analyzer.amazonaws.com/AWSServiceRoleForAccessAnalyzer"
        },
    "Action": [
        "kms:DescribeKey",
        "kms:GetKeyPolicy",
        "kms:List*"
    ],
    "Resource": "*"
},
```

Dopo aver ricevuto un risultato di accesso negato per una risorsa chiave KMS e quindi averlo risolto aggiornando la policy della chiave, il risultato viene aggiornato sullo stato Risolto. Se sono presenti istruzioni di policy o concessioni della chiave che concedono l'autorizzazione alla chiave per un'entità esterna, è possibile che vengano visualizzati ulteriori risultati per la risorsa chiave.

AWS Lambda funzioni e livelli

Per quanto riguarda AWS Lambda le funzioni, IAM Access Analyzer analizza le policy, incluse le dichiarazioni di condizione contenute in una policy, che concedono l'accesso alla funzione a un'entità esterna. Con Lambda, puoi collegare policy basate sulle risorse univoche a funzioni, versioni, alias e livelli. Il Sistema di analisi degli accessi IAM riporta gli accessi esterni in base a policy basate sulle risorse collegate a funzioni e livelli. Il Sistema di analisi degli accessi IAM non riporta l'accesso esterno in base a policy basate su risorse collegate ad alias e versioni specifiche richiamate utilizzando un ARN completo.

Per ulteriori informazioni, consulta Using Resource-based policy for Lambda e Using versions nella Developer Guide. AWS Lambda

Code Amazon Simple Queue Service

Per le code Amazon SQS, Sistema di analisi degli accessi IAM analizza le policy, incluse le istruzioni di condizione in una policy che consentono a un'entità esterna di accedere a una coda.

AWS Secrets Manager segreti

Per quanto riguarda AWS Secrets Manager i segreti, IAM Access Analyzer analizza le policy, incluse le condizioni contenute in una policy, che consentono a un'entità esterna di accedere a un segreto.

Argomenti su Amazon Simple Notification Service

Sistema di analisi degli accessi IAM analizza le policy basate sulle risorse allegate agli argomenti di Amazon SNS, incluse le istruzioni delle condizioni nelle policy che consentono l'accesso esterno a un argomento. Puoi consentire agli account esterni di eseguire azioni Amazon SNS come la sottoscrizione e la pubblicazione di argomenti tramite una policy basata sulle risorse. Un argomento Amazon SNS è accessibile dall'esterno se i principali di un account esterno alla tua zona di fiducia possono eseguire operazioni sull'argomento. Se scegli `Everyone` nella tua policy quando crei un argomento Amazon SNS, rendi l'argomento accessibile al pubblico. `AddPermission` è un altro modo per aggiungere una policy basata sulle risorse a un argomento Amazon SNS che consente l'accesso esterno.

Volumi e snapshot di Amazon Elastic Block Store

Gli snapshot di volumi di Amazon Elastic Block Store non hanno policy basate sulle risorse. Uno snapshot viene condiviso tramite le autorizzazioni di condivisione di Amazon EBS. Per gli snapshot di volume Amazon EBS, Sistema di analisi degli accessi IAM analizza le liste di controllo degli accessi che consentono a un'entità esterna di accedere a uno snapshot. Se crittografato, uno snapshot di volume Amazon EBS può essere condiviso con account esterni. Uno snapshot di volume non crittografato può essere condiviso con account esterni e garantire l'accesso pubblico. Le impostazioni di condivisione sono nell'attributo `CreateVolumePermissions` dello snapshot. Quando i clienti visualizzano in anteprima l'accesso esterno di uno snapshot di Amazon EBS, possono specificare la chiave di crittografia come indicatore del fatto che lo snapshot è crittografato, in modo simile a come l'anteprima di Sistema di analisi degli accessi IAM gestisce i segreti di Gestione dei segreti.

Amazon Relational Database Service

Gli snapshot del database Amazon RDS non hanno policy basate su risorse. Uno snapshot del database viene condiviso tramite le autorizzazioni del database Amazon RDS e possono essere

condivisi solo snapshot manuali del database. Per gli analizzatori degli accessi esterni, Sistema di analisi degli accessi IAM analizza le liste di controllo degli accessi che consentono a un'entità esterna di accedere a uno snapshot del database Amazon RDS. Gli snapshot del database non crittografati possono essere pubblici. Gli snapshot del database crittografati non possono essere condivisi pubblicamente, ma possono essere condivisi con un massimo di altri 20 account. Per ulteriori informazioni, consulta Creazione di uno snapshot DB. Sistema di analisi degli accessi IAM considera la capacità di esportare uno snapshot manuale del database (ad esempio, in un bucket Amazon S3) come accesso attendibile.

Per gli analizzatori degli accessi interni, Sistema di analisi degli accessi IAM genera un esito quando un principale (utente o ruolo) all'interno dell'organizzazione o dell'account ha accesso a uno snapshot del database Amazon RDS specificato.

> **Note**
>
> Sistema di analisi degli accessi IAM non identifica l'accesso pubblico o l'accesso multi-account configurato direttamente sul database stesso. Sistema di analisi degli accessi IAM identifica solo i risultati per l'accesso pubblico o l'accesso multi-account configurati nello snapshot del database di Amazon RDS.

Snapshot di cluster di database di Amazon Relational Database Service

Gli snapshot del cluster di database Amazon RDS non hanno policy basate su risorse. Uno snapshot viene condiviso tramite le autorizzazioni del cluster di database di Amazon RDS. Per gli analizzatori degli accessi esterni, Sistema di analisi degli accessi IAM analizza le liste di controllo degli accessi che consentono a un'entità esterna di accedere a un cluster di database Amazon RDS. Gli snapshot del cluster non crittografati possono essere pubblici. Gli snapshot del cluster crittografati non possono essere condivisi pubblicamente. Gli snapshot del cluster non crittografati e crittografati possono essere condivisi con al massimo altri 20 account. Per ulteriori informazioni, consulta la sezione Creating a DB Cluster Snapshot (Creazione di uno snapshot cluster database). Sistema di analisi degli accessi IAM considera la capacità di esportare uno snapshot del cluster di database (ad esempio, in un bucket Amazon S3) come accesso attendibile.

Per gli analizzatori degli accessi interni, Sistema di analisi degli accessi IAM genera un esito quando un principale (utente o ruolo) all'interno dell'organizzazione o dell'account ha accesso a uno snapshot del cluster di database Amazon RDS specificato.

> **Note**
>
> I risultati di IAM Access Analyzer non includono il monitoraggio di alcuna condivisione di cluster e cloni di Amazon RDS DB con altri Account AWS utenti o organizzazioni. AWS Resource Access Manager Sistema di analisi degli accessi IAM identifica solo i risultati per l'accesso pubblico o l'accesso multi-account configurati nello snapshot del cluster di database di Amazon RDS.

Repository di Amazon Elastic Container Registry

Per i repository Amazon ECR, Sistema di analisi degli accessi IAM analizza le policy basate su risorse, incluse le istruzioni di condizione in una policy che consentono a un'entità esterna di accedere a un repository (in modo simile ad altri tipi di risorse come gli argomenti di Amazon SNS e i file system Amazon EFS). Per i repository Amazon ECR, un principale deve avere l'autorizzazione per `ecr:GetAuthorizationToken` tramite una policy basata sull'identità per essere considerato disponibile esternamente.

File system di Amazon Elastic File System

Per le code Amazon SQS, Sistema di analisi degli accessi IAM analizza le policy, incluse le istruzioni di condizione in una policy che consentono a un'entità esterna di accedere a una coda. Un file system Amazon EFS è accessibile dall'esterno se i principali di un account esterno alla tua zona di attendibilità possono eseguire operazioni su quel file system. L'accesso è definito da una policy del file system che utilizza IAM e da come viene montato il file system. Ad esempio, il montaggio del file system Amazon EFS in un altro account è considerato accessibile dall'esterno, a meno che tale account non si trovi nella tua organizzazione e tu non abbia definito l'organizzazione come la tua zona di attendibilità. Se monti il file system da un cloud privato virtuale con una sottorete pubblica, il file system sarà accessibile dall'esterno. Quando usi Amazon EFS con AWS Transfer Family, le richieste di accesso al file system ricevute da un server Transfer Family di proprietà di un account diverso dal file system vengono bloccate se il file system consente l'accesso pubblico.

Flussi Amazon DynamoDB

Per gli analizzatori degli accessi esterni, Sistema di analisi degli accessi IAM genera un esito se una policy di DynamoDB consente almeno un'operazione tra account che consente a un'entità esterna di accedere a un flusso di DynamoDB. Per ulteriori informazioni sulle azioni multi-account supportate

per DynamoDB, consulta Azioni IAM supportate da policy basate sulle risorse nella Guida per gli sviluppatori di Amazon DynamoDB.

Per gli analizzatori degli accessi interni, Sistema di analisi degli accessi IAM genera un esito quando un principale (utente o ruolo) all'interno dell'organizzazione o dell'account ha accesso a un flusso di DynamoDB specificato.

Tabelle Amazon DynamoDB

Per gli analizzatori degli accessi esterni, Sistema di analisi degli accessi IAM genera un esito per una tabella di DynamoDB se una policy di DynamoDB consente almeno un'operazione tra account che consente a un'entità esterna di accedere a una tabella o un indice DynamoDB. Per ulteriori informazioni sulle azioni multi-account supportate per DynamoDB, consulta Azioni IAM supportate da policy basate sulle risorse nella Guida per gli sviluppatori di Amazon DynamoDB.

Per gli analizzatori degli accessi interni, Sistema di analisi degli accessi IAM genera un esito quando un principale (utente o ruolo) all'interno dell'organizzazione o dell'account ha accesso a una tabella DynamoDB specificata.

Amministratore delegato per Sistema di analisi degli accessi IAM.

Se esegui la configurazione AWS Identity and Access Management Access Analyzer nel tuo account di AWS Organizations gestione, puoi aggiungere un account membro nell'organizzazione come amministratore delegato per gestire IAM Access Analyzer per la tua organizzazione. L'amministratore delegato dispone delle autorizzazioni per creare e gestire gli analizzatori nell'organizzazione. Solo l'account di gestione può aggiungere un amministratore delegato.

L'amministratore delegato per Sistema di analisi degli accessi IAM è un account membro all'interno dell'organizzazione che dispone delle autorizzazioni per creare e gestire gli analizzatori che analizzano gli accessi nell'organizzazione. Solo l'account di gestione può aggiungere, rimuovere o modificare un amministratore delegato.

Se aggiungi un amministratore delegato, puoi in un secondo momento passare a un account diverso per l'amministratore delegato. In questo caso, l'account amministratore delegato precedente perde l'autorizzazione per tutti gli analizzatori creati utilizzando tale account per analizzare gli accessi nell'organizzazione. Questi analizzatori passano a uno stato disabilitato e non generano più nuovi risultati, né aggiornano quelli esistenti. Anche i risultati esistenti per questi analizzatori non sono più accessibili. Puoi accedervi nuovamente in futuro configurando l'account come amministratore delegato. Se ritieni che uno stesso account di amministratore delegato non verrà più utilizzato, è consigliabile eliminare gli analizzatori prima di modificare l'amministratore delegato. Questa

operazione elimina tutti i risultati generati. Quando il nuovo amministratore delegato crea nuovi analizzatori, vengono generate nuove istanze degli stessi risultati. Non perdi alcun risultato, vengono solo generati per il nuovo analizzatore in un altro account. Puoi continuare ad accedere ai risultati dell'organizzazione anche utilizzando l'account di gestione dell'organizzazione che dispone anche di autorizzazioni di amministratore. Il nuovo amministratore delegato deve creare i nuovi analizzatori per Sistema di analisi degli accessi IAM per avviare il monitoraggio delle risorse nell'organizzazione.

Se l'amministratore delegato lascia l' AWS organizzazione, i privilegi di amministrazione delegata vengono rimossi dall'account. Tutti gli analizzatori nell'account con l'organizzazione come zona di attendibilità passano a uno stato disabilitato. Anche i risultati esistenti per questi analizzatori non sono più accessibili.

La prima volta che si configurano gli analizzatori nell'account di gestione, puoi scegliere l'opzione Aggiungi amministratore delegato nella pagina Impostazioni dell'analizzatore nella console di Sistema di analisi degli accessi IAM.

> **Note**
>
> Sistema di analisi degli accessi IAM addebita i costi per gli analizzatori degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese da ogni analizzatore. Se crei un analizzatore degli accessi inutilizzati nell'account di gestione e uno nell'account dell'amministratore delegato, ti verranno addebitati i costi per entrambi gli analizzatori. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

Dopo aver cambiato l'amministratore delegato, il nuovo amministratore deve creare gli analizzatori per avviare il monitoraggio dell'accesso alle risorse dell'organizzazione.

Aggiungere un amministratore delegato per Sistema di analisi degli accessi IAM.

Se esegui la configurazione AWS Identity and Access Management Access Analyzer nel tuo account di AWS Organizations gestione, puoi aggiungere un account membro nell'organizzazione come amministratore delegato per gestire IAM Access Analyzer per la tua organizzazione. L'amministratore delegato dispone delle autorizzazioni per creare e gestire gli analizzatori nell'organizzazione. Solo l'account di gestione può aggiungere un amministratore delegato.

Per aggiungere un amministratore delegato utilizzando la console

1. Accedi alla AWS console utilizzando l'account di gestione dell'organizzazione.

2. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
3. In Access Analyzer, scegli Impostazioni dell'analizzatore.
4. Scegliere Add delegated administrator (Aggiungi amministratore delegato).
5. Nel campo Amministratore delegato, inserisci il numero Account AWS di un account membro dell'organizzazione per creare l'amministratore delegato.

   L'account deve essere un membro dell'organizzazione.
6. Scegli Save changes (Salva modifiche).

Per aggiungere un amministratore delegato utilizzando AWS CLI o AWS SDKs

Quando crei un analizzatore per analizzare l'accesso all'interno dell'organizzazione in un account amministratore delegato utilizzando la AWS CLI, l' AWS API (utilizzando AWS SDKs) oppure CloudFormation, devi utilizzare per abilitare l'accesso AWS Organizations APIs al servizio per IAM Access Analyzer e registrare l'account membro come amministratore delegato.

1. Abilita l'accesso affidabile ai servizi per IAM Access Analyzer in. AWS Organizations Scopri come abilitare o disabilitare l'accesso affidabile nella Guida per l' AWS Organizations utente.
2. Registra un account membro valido della tua AWS organizzazione come amministratore delegato utilizzando l'operazione AWS Organizations `RegisterDelegatedAdministrator`API o il `register-delegated-administrator` AWS CLI comando.

Regole di archiviazione

Le regole di archiviazione archiviano automaticamente i nuovi risultati che soddisfano i criteri definiti al momento della creazione della regola. È inoltre possibile applicare le regole di archiviazione retroattivamente per archiviare i risultati esistenti che soddisfano i criteri delle regole di archiviazione. Ad esempio, puoi creare una regola di archiviazione per archiviare automaticamente tutti i risultati per un bucket Amazon S3 specifico a cui concedi regolarmente l'accesso. In alternativa, se concedi l'accesso a più risorse a un responsabile specifico, puoi creare una regola che archivi automaticamente tutti i nuovi risultati generati per l'accesso concesso a tale responsabile. In questo modo puoi concentrarti solo sui risultati attivi che possono indicare un rischio per la sicurezza.

Quando si crea una regola di archiviazione, vengono archiviati automaticamente solo i nuovi risultati che corrispondono ai criteri della regola. I risultati esistenti non vengono archiviati automaticamente.

Quando si crea una regola, è possibile includere fino a 20 valori per criterio nella regola. Per l'elenco delle chiavi di filtro che puoi utilizzare per creare o aggiornare una regola di archivio, consulta Chiavi di filtro di Sistema di analisi degli accessi IAM.

> **Note**
>
> Quando crei o modifichi una regola di archiviazione, Sistema di analisi degli accessi IAM non convalida i valori inclusi nel filtro per la regola. Ad esempio, se aggiungi una regola per la corrispondenza con un Account AWS, Sistema di analisi degli accessi IAM accetta qualsiasi valore nel campo, anche se non è un numero di account AWS valido.

Per creare una regola di archiviazione

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Access Analyzer e poi Impostazioni dell'analizzatore.
3. Nella sezione Analizzatori, scegli l'analizzatore per il quale desideri creare una regola di archivio.
4. Nella scheda Regole di archivio, scegli Crea regola di archivio.
5. Immettere un nome per la regola se si desidera modificare il nome predefinito.
6. Nella sezione Rule (Regola) in Criteria (Criteri), selezionare una proprietà da corrispondere per la regola.
7. Scegli una condizione per il valore della proprietà, ad esempio Contiene, È o Non è uguale.

   Gli operatori disponibili dipendono dalla proprietà scelta.
8. Facoltativamente, aggiungere altri valori per la proprietà o altri criteri per la regola. Per i risultati degli accessi esterni, per assicurarti che la regola non archivi i nuovi risultati per l'accesso pubblico, puoi anche includere il criterio Accesso pubblico e impostarlo su false.

   Per aggiungere un altro valore per un criterio, scegliere Add another value (Aggiungi un altro valore). Per aggiungere un altro criterio per la regola, scegli Aggiungi criterio.
9. Al termine dell'aggiunta di criteri e valori, scegli Create rule (Crea regola) per applicare la regola solo ai nuovi risultati. Scegli Create and archive active findings (Crea e archivi i risultati attivi) per archiviare i risultati nuovi ed esistenti in base ai criteri della regola. Nella sezione Results (Risultati) puoi esaminare l'elenco dei risultati attivi a cui si applica la regola di archiviazione.

Ad esempio, per creare una regola per i risultati degli accessi esterni che archivia automaticamente tutti i risultati per i bucket Amazon S3: scegliere Tipo di risorsa e poi è per la condizione. Quindi scegli Bucket S3 dall'elenco Valori.

Per creare una regola per i risultati degli accessi inutilizzati che archivia automaticamente tutti i risultati per un determinato account, scegli Account del proprietario della risorsa e poi Uguale per la condizione. Digitate l' Account AWS ID nella casella di testo Valore.

Continua a definire i criteri per personalizzare la regola in base all'ambiente, quindi scegli Crea regola.

Se stai creando una nuova regola e aggiungi più criteri, puoi rimuovere un singolo criterio dalla regola scegliendo Rimuovi questo criterio. È possibile rimuovere un valore aggiunto per un criterio scegliendo Remove value (Rimuovi valore).

Per modificare una regola di archiviazione

1. Scegli il nome della regola da modificare nella colonna Nome.

   È possibile modificare una sola regola di archiviazione alla volta.
2. Per ogni criterio, aggiungi nuovi criteri e valori o rimuovi quelli esistenti.
3. Scegli Save changes (Salva modifiche) per applicare la regola solo ai nuovi risultati. Scegli Save and archive active findings (Salva e archivia i risultati attivi) per archiviare i risultati nuovi ed esistenti in base ai criteri della regola.

Per eliminare una regola di archiviazione

1. Seleziona la casella di controllo per la regola da eliminare.
2. Scegli Elimina.
3. Digitare **`delete`** nella finestra di dialogo di conferma Delete archive rule (Elimina regola di archiviazione) e quindi scegliere Delete (Elimina).

Le regole vengono eliminate solo dall'analizzatore nella regione corrente. È necessario eliminare le regole di archiviazione separatamente per ogni analizzatore creato in altre regioni.

Monitoraggio AWS Identity and Access Management Access Analyzer con Amazon EventBridge

Utilizza le informazioni contenute in questo argomento per scoprire come monitorare i risultati di IAM Access Analyzer e accedere alle anteprime con Amazon. EventBridge EventBridge è la nuova versione di Amazon CloudWatch Events.

Eventi dei risultati

IAM Access Analyzer invia un evento EventBridge per ogni risultato generato, per modificare lo stato di un risultato esistente e quando un risultato viene eliminato. Per ricevere risultati e notifiche sui risultati, devi creare una regola di evento in Amazon EventBridge. Quando si crea una regola di evento, è anche possibile specificare un'operazione di destinazione da attivare in base alla regola. Ad esempio, puoi creare una regola di evento che attiva un argomento Amazon SNS quando viene ricevuto un evento per un nuovo risultato da Sistema di analisi degli accessi IAM. I dettagli sulla policy di controllo delle risorse (RCP) sono disponibili nella sezione dei dettagli dell'evento.

Accesso a eventi di anteprima

IAM Access Analyzer invia un evento EventBridge a ogni anteprima di accesso e modifica del relativo stato. Ciò include un evento quando viene creata per la prima volta l'anteprima di accesso (stato Creazione), quando l'anteprima di accesso è completata (stato Completato) o quando la creazione dell'anteprima di accesso non è riuscita (stato Non riuscito). Per ricevere notifiche sulle anteprime di accesso, devi creare una regola di evento in. EventBridge Quando crei una regola di evento, puoi anche specificare un'operazione di destinazione da attivare in base alla regola. Ad esempio, puoi creare una regola di evento che attiva un argomento Amazon SNS quando viene ricevuta un'anteprima dell'accesso completato da Sistema di analisi degli accessi IAM.

Frequenza delle notifiche di evento

IAM Access Analyzer invia gli eventi relativi a nuove scoperte e scoperte con aggiornamenti di stato EventBridge entro circa un'ora dal momento in cui si verifica l'evento nel tuo account. IAM Access Analyzer invia anche eventi EventBridge quando un risultato risolto viene eliminato perché il periodo di conservazione è scaduto. Per i risultati che vengono eliminati perché l'analizzatore che li ha generati viene eliminato, l'evento viene inviato a EventBridge circa 24 ore dall'eliminazione dell'analizzatore. Quando un risultato viene eliminato, lo stato del risultato non viene modificato. L'attributo `isDeleted` viene impostato su `true`. IAM Access Analyzer invia anche eventi per le anteprime di accesso appena create e le modifiche allo stato delle anteprime di accesso a. EventBridge

Esempi di eventi relativi ai risultati degli accessi esterni

Di seguito è riportato un esempio di evento di ricerca degli accessi esterni di IAM Access Analyzer inviato a. EventBridge L'`id`elenco è l'ID dell'evento in EventBridge. Per ulteriori informazioni, consulta Eventi e modelli di eventi in EventBridge.

Nell'oggetto `detail`, i valori per gli attributi `accountId` e `region` si riferiscono all'account e alla regione riportati nel risultato. L'attributo `isDeleted` indica se l'evento è derivato dal risultato eliminato. L'`id` è l'ID risultato. L'array `resources` è un singleton con l'ARN dell'analizzatore che ha generato il risultato.


```
{
    "account": "111122223333",
    "detail": {
        "accountId": "111122223333",
        "action": [
            "s3:GetObject"
        ],
        "analyzedAt": "2019-11-21T01:22:22Z",
        "condition": {},
        "createdAt": "2019-11-20T04:58:50Z",
        "id": "22222222-dcba-4444-dcba-333333333333",
        "isDeleted": false,
        "isPublic": false,
        "principal": {
            "AWS": "999988887777"
        },
        "region": "us-west-2",
        "resource": "arn:aws:s3:::amzn-s3-demo-bucket",
        "resourceType": "AWS::S3::Bucket",
        "status": "ACTIVE",
        "updatedAt": "2019-11-21T01:14:07Z",
        "version": "1.0"
    },
    "detail-type": "Access Analyzer Finding",
    "id": "11111111-2222-4444-aaaa-333333333333",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
    "source": "aws.access-analyzer",
    "time": "2019-11-21T01:22:33Z",
```

```
    "version": "0"
}
```

IAM Access Analyzer invia anche gli eventi a EventBridge per rilevare gli errori. Un risultato di errore è generato quando Sistema di analisi degli accessi IAM non è in grado di analizzare la risorsa. Gli eventi per i risultati di errore includono un attributo `error` come illustrato nell'esempio seguente.

```
{
    "account": "111122223333",
    "detail": {
        "accountId": "111122223333",
        "analyzedAt": "2019-11-21T01:22:22Z",
        "createdAt": "2019-11-20T04:58:50Z",
        "error": "ACCESS_DENIED",
        "id": "22222222-dcba-4444-dcba-333333333333",
        "isDeleted": false,
        "region": "us-west-2",
        "resource": "arn:aws:s3:::amzn-s3-demo-bucket",
        "resourceType": "AWS::S3::Bucket",
        "status": "ACTIVE",
        "updatedAt": "2019-11-21T01:14:07Z",
        "version": "1.0"
    },
    "detail-type": "Access Analyzer Finding",
    "id": "11111111-2222-4444-aaaa-333333333333",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
    "source": "aws.access-analyzer",
    "time": "2019-11-21T01:22:33Z",
    "version": "0"
}
```

Esempi di eventi relativi agli esiti degli accessi esterni

Di seguito è riportato un esempio di evento di ricerca degli accessi interni di IAM Access Analyzer inviato a. EventBridge L'`id`elenco è l'ID dell'evento in EventBridge. Per ulteriori informazioni, consulta Eventi e modelli di eventi in EventBridge.

Nell'oggetto `detail`, i valori per gli attributi `accountId` e `principalOwnerAccount` si riferiscono all'account del principale riportato nell'esito. L'attributo `isDeleted` indica se l'evento è derivato dal

risultato eliminato. L'`id` è l'ID risultato. Il valore `resource` è l'ARN dell'analizzatore che ha generato l'esito.


```
{
    "version": "0",
    "id": "b45c3678-c278-b593-6121-c155259ce1b5",
    "detail-type": "Internal Access Finding",
    "source": "aws.access-analyzer",
    "account": "111122223333",
    "time": "2025-04-08T19:42:49Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:access-analyzer:us-east-1:111122223333:analyzer/testAnalyzer"
    ],
    "detail": {
        "accessType": "INTRA_ACCOUNT",
        "action": [
            "s3:GetObject"
        ],
        "analyzedAt": "2025-04-08T03:18:43.509465073Z",
        "condition": {},
        "createdAt": "2025-04-07T21:33:49.914099224Z",
        "id": "11111111-2222-4444-aaaa-333333333333",
        "isDeleted": false,
        "findingType": "InternalAccess",
        "principal": {
            "AWS": "arn:aws:iam::111122223333:role/MyRole_6"
        },
        "principalOwnerAccount": "111122223333",
        "principalType": "IAM_ROLE",
        "resource": "arn:aws:s3:::critical-data",
        "resourceControlPolicyRestrictionType": "NOT_APPLICABLE",
        "accountId": "111122223333",
        "resourceType": "AWS::S3::Bucket",
        "serviceControlPolicyRestrictionType": "NOT_APPLICABLE",
        "status": "ACTIVE",
        "updatedAt": "2025-04-08T03:22:12.654688231Z",
        "version": "1.0"
    }
}
```

IAM Access Analyzer invia anche gli eventi a EventBridge per rilevare gli errori. Un risultato di errore è generato quando Sistema di analisi degli accessi IAM non è in grado di analizzare la risorsa. Gli eventi per i risultati di errore includono un attributo `error` come illustrato nell'esempio seguente.

```
{
    "version": "0",
    "id": "5a94b99b-e87d-a6a7-58c7-f47871532860",
    "detail-type": "Internal Access Finding",
    "source": "aws.access-analyzer-test",
    "account": "444455556666",
    "time": "2025-05-07T11:57:54Z",
    "region": "us-west-2",
    "resources": ["arn:aws:access-analyzer-beta:us-west-2:444455556666:analyzer/
example-analyzer"],
    "detail": {
        "analyzedAt": "2025-03-24T19:58:52.512329448Z",
        "createdAt": "2025-03-22T03:30:46.920200692Z",
        "id": "ef573afd-12a5-4095-87a6-bf2f25109895",
        "isDeleted": false,
        "findingType": "InternalAccess",
        "resource": "arn:aws:s3:::test-entity-88",
        "accountId": "111122223333",
        "resourceControlPolicyRestrictionType": "NOT_APPLICABLE",
        "resourceType": "AWS::S3::Bucket",
        "serviceControlPolicyRestrictionType": "NOT_APPLICABLE",
        "error": "ACCESS_DENIED", // can be INTERNAL_ERROR and ACCESS_DENIED
        "status": "ACTIVE",
        "updatedAt": "2025-03-24T20:09:39.176075014Z",
        "version": "1.0"
    }
}
```

Esempi di eventi relativi ai risultati degli accesso inutilizzati

Di seguito è riportato un esempio di evento di ricerca degli accessi non utilizzati di IAM Access Analyzer inviato a. EventBridge L'`id`elenco è l'ID dell'evento in. EventBridge Per ulteriori informazioni, consulta Eventi e modelli di eventi in EventBridge.

Nell'oggetto `detail`, i valori per gli attributi `accountId` e `region` si riferiscono all'account e alla regione riportati nel risultato. L'attributo `isDeleted` indica se l'evento è derivato dal risultato eliminato. L'`id` è l'ID risultato.

```
{
    "version": "0",
    "id": "dc7ce3ee-114b-3243-e249-7f10f9054b21",
    "detail-type": "Unused Access Finding for IAM entities",
    "source": "aws.access-analyzer",
    "account": "123456789012",
    "time": "2023-09-29T17:31:40Z",
    "region": "us-west-2",
    "resources": [
       "arn:aws:access-analyzer:us-west-2:123456789012:analyzer/
integTestLongLivingAnalyzer-DO-NOT-DELETE"
       ],
    "detail": {
        "findingId": "b8ae0460-5d29-4922-b92a-ba956c986277",
        "resource": "arn:aws:iam::111122223333:role/FindingIntegTestFakeRole",
        "resourceType": "AWS::IAM::Role",
        "accountId": "111122223333",
        "createdAt": "2023-09-29T17:29:18.758Z",
        "updatedAt": "2023-09-29T17:29:18.758Z",
        "analyzedAt": "2023-09-29T17:29:18.758Z",
        "previousStatus": "",
        "status": "ACTIVE",
        "version": "62160bda-8e94-46d6-ac97-9670930d8ffb",
        "isDeleted": false,
        "findingType": "UnusedPermission",
        "numberOfUnusedServices": 0,
        "numberOfUnusedActions": 1
        }
    }
```

IAM Access Analyzer invia anche gli eventi a EventBridge per rilevare gli errori. Un risultato di errore è generato quando Sistema di analisi degli accessi IAM non è in grado di analizzare la risorsa. Gli eventi per i risultati di errore includono un attributo `error` come illustrato nell'esempio seguente.

```
{
    "version": "0",
    "id": "c2e7aa1a-4df7-7652-f33e-64113b8997d4",
    "detail-type": "Unused Access Finding for IAM entities",
    "source": "aws.access-analyzer",
    "account": "111122223333",
    "time": "2023-10-31T20:26:12Z",
    "region": "us-west-2",
```

```
    "resources": [
      "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/ba811f91-
de99-41a4-97c0-7481898b53f2"
      ],
    "detail": {
        "findingId": "b01a34f2-e118-46c9-aef8-0d8526b495c7",
        "resource": "arn:aws:iam::123456789012:role/TestRole",
        "resourceType": "AWS::IAM::Role",
        "accountId": "444455556666",
        "createdAt": "2023-10-31T20:26:08.647Z",
        "updatedAt": "2023-10-31T20:26:09.245Z",
        "analyzedAt": "2023-10-31T20:26:08.525Z",
        "previousStatus": "",
        "status": "ACTIVE",
        "version": "7c7a72a2-7963-4c59-ac71-f0be597010f7",
        "isDeleted": false,
        "findingType": "UnusedIAMRole",
        "error": "INTERNAL_ERROR"
        }
  }
```

Esempio di eventi di anteprima di accesso

L'esempio seguente mostra i dati per il primo evento a cui viene inviato EventBridge quando si crea un'anteprima di accesso. L'array `resources` è un singleton con l'ARN dell'analizzatore a cui è associata l'anteprima di accesso. Nell'oggetto `detail`, `id` si riferisce all'ID di anteprima dell'accesso e `configuredResources` fa riferimento alla risorsa per la quale è stata creata l'anteprima di accesso. `status` è `Creating` e fa riferimento allo stato dell'anteprima dell'accesso. `previousStatus` non è specificato perché l'anteprima di accesso è stata appena creata.

```
{
    "account": "111122223333",
    "detail": {
        "accessPreviewId": "aaaabbbb-cccc-dddd-eeee-ffffaaaabbbb",
        "configuredResources": [
            "arn:aws:s3:::amzn-s3-demo-bucket"
        ],
        "createdAt": "2020-02-20T00:00:00.00Z",
        "region": "us-west-2",
        "status": "CREATING",
        "version": "1.0"
    },
```

```
    "detail-type": "Access Preview State Change",
    "id": "aaaabbbb-2222-3333-4444-555566667777",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
    "source": "aws.access-analyzer",
    "time": "2020-02-20T00:00:00.00Z",
    "version": "0"
}
```

L'esempio seguente mostra i dati di un evento inviato a EventBridge per un'anteprima di accesso con una modifica dello stato da `Creating` a`Completed`. Nell'oggetto dettaglio, `id` fa riferimento all'ID di anteprima dell'accesso. `status` e `previousStatus` fanno riferimento allo stato dell'anteprima dell'accesso, in cui lo stato precedente era `Creating` e lo stato corrente è `Completed`.

```
{
    "account": "111122223333",
    "detail": {
        "accessPreviewId": "aaaabbbb-cccc-dddd-eeee-ffffaaaabbbb",
        "configuredResources": [
            "arn:aws:s3:::amzn-s3-demo-bucket"
        ],
        "createdAt": "2020-02-20T00:00:00.000Z",
        "previousStatus": "CREATING",
        "region": "us-west-2",
        "status": "COMPLETED",
        "version": "1.0"
    },
    "detail-type": "Access Preview State Change",
    "id": "11112222-3333-4444-5555-666677778888",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
    "source": "aws.access-analyzer",
    "time": "2020-02-20T00:00:00.00Z",
    "version": "0"
}
```

L'esempio seguente mostra i dati di un evento inviato a EventBridge per un'anteprima di accesso con una modifica dello stato da `Creating` a`Failed`. Nell'oggetto `detail`, `id` fa riferimento all'ID

di anteprima dell'accesso. `status` e `previousStatus` fanno riferimento allo stato dell'anteprima dell'accesso, in cui lo stato precedente era `Creating` e lo stato corrente è `Failed`. Il campo `statusReason` fornisce il codice motivo che indica che l'anteprima di accesso non è riuscita a causa di una configurazione di risorse non valida.


```
{
    "account": "111122223333",
    "detail": {
        "accessPreviewId": "aaaabbbb-cccc-dddd-eeee-ffffaaaabbbb",
        "configuredResources": [
            "arn:aws:s3:::amzn-s3-demo-bucket"
        ],
        "createdAt": "2020-02-20T00:00:00.00Z",
        "previousStatus": "CREATING",
        "region": "us-west-2",
        "status": "FAILED",
        "statusReason": {
            "code": "INVALID_CONFIGURATION"
        },
        "version": "1.0"
    },
    "detail-type": "Access Preview State Change",
    "id": "99998888-7777-6666-5555-444433332222",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
    "source": "aws.access-analyzer",
    "time": "2020-02-20T00:00:00.00Z",
    "version": "0"
}
```


Creazione di una regola di evento mediante la console

La procedura seguente descrive come creare una regola di evento utilizzando la console.

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Utilizzando i seguenti valori, crea una EventBridge regola che monitori la ricerca di eventi o acceda agli eventi di anteprima:

   - Per Tipo di regola, scegli Regola con un modello di eventi.

- In Event source (Origine eventi), scegli Other (Altro).
- In Event pattern (Modello di eventi), scegli Custom patterns (JSON editor) (Modelli personalizzati [editor JSON]) e incolla uno dei seguenti esempi di modelli di eventi nell'area di testo:
  - Per creare una regola basata su un evento di Sistema di analisi degli accessi IAM, utilizza il seguente esempio di modello:

```
{
  "source": [
    "aws.access-analyzer"
  ]
}
```

  - Per creare una regola basata su un evento relativo agli esiti degli accessi esterni, interni o non utilizzati, usa il seguente schema di esempio:

```
{
  "source": [
    "aws.access-analyzer"
  ],
  "detail-type": [
    "Access Analyzer Finding",
    "Internal Access Finding",
    "Unused Access Finding for IAM entities"
  ]
}
```

  - Per creare una regola basata su un evento relativo ai risultati degli accessi esterni, utilizza il seguente esempio di modello:

```
{
  "source": [
    "aws.access-analyzer"
  ],
  "detail-type": [
    "Access Analyzer Finding"
  ]
}
```

  - Per creare una regola basata solo su un evento relativo agli esiti degli accessi interni, usa il seguente schema di esempio:

```
{
  "source": [
    "aws.access-analyzer"
  ],
  "detail-type": [
    "Internal Access Finding"
  ]
}
```

- Per creare una regola basata solo su un evento relativo ai risultati degli accessi inutilizzati, utilizza il seguente esempio di modello:

```
{
  "source": [
    "aws.access-analyzer"
  ],
  "detail-type": [
    "Unused Access Finding for IAM entities"
  ]
}
```

- Per creare una regola basata su un evento di anteprima di accesso, utilizza il seguente esempio di modello:

```
{
  "source": [
    "aws.access-analyzer"
  ],
  "detail-type": [
    "Access Preview State Change"
  ]
}
```

- Per i tipi di Target, scegli il AWS servizio e per Seleziona una destinazione, scegli una destinazione come un argomento o AWS Lambda una funzione di Amazon SNS. La destinazione viene attivata quando viene ricevuto un evento che corrisponde al modello di evento definito nella regola.

Per ulteriori informazioni sulla creazione di regole, consulta la sezione Creazione di EventBridge regole Amazon che reagiscono agli eventi nella Amazon EventBridge User Guide.

Creazione di una regola evento mediante la CLI

1. Utilizza quanto segue per creare una regola per Amazon EventBridge utilizzando il AWS CLI. Sostituisci il nome della regola *TestRule* con il nome della regola.

```
aws events put-rule --name TestRule --event-pattern "{\"source\":[\"aws.access-analyzer\"]}"
```

2. È possibile personalizzare la regola per attivare operazioni di destinazione solo per un sottoinsieme di risultati generati, ad esempio risultati con attributi specifici. Nell'esempio seguente viene illustrato come creare una regola che attiva un'operazione di destinazione solo per i risultati con stato Attivo.

```
aws events put-rule --name TestRule --event-pattern "{\"source\":[\"aws.access-analyzer\"],\"detail-type\":[\"Access Analyzer Finding\"],\"detail\":{\"status\":[\"ACTIVE\"]}}"
```

   Nell'esempio seguente viene illustrato come creare una regola che attiva un'operazione di destinazione solo per le anteprime di accesso con stato da `Creating` a `Completed`.

```
aws events put-rule --name TestRule --event-pattern "{\"source\":[\"aws.access-analyzer\"],\"detail-type\":[\"Access Preview State Change\"],\"detail\":{\"status\":[\"COMPLETED\"]}}"
```

3. Per definire una funzione Lambda come destinazione per la regola creata, utilizza il seguente comando di esempio. Sostituire la regione e il nome della funzione nell'ARN come appropriato per l'ambiente in uso.

```
aws events put-targets --rule TestRule --targets Id=1,Arn=arn:aws:lambda:us-east-1:111122223333:function:MyFunction
```

4. Aggiungere le autorizzazioni necessarie per richiamare la destinazione della regola. Nell'esempio seguente viene illustrato come concedere autorizzazioni a una funzione Lambda seguendo gli esempi precedenti.

```
aws lambda add-permission --function-name MyFunction --statement-id 1 --action
 'lambda:InvokeFunction' --principal events.amazonaws.com
```

Integra IAM Access Analyzer con AWS Security Hub CSPM

AWS Security Hub CSPM fornisce una visione completa dello stato di sicurezza in AWS. Consente di valutare l'ambiente rispetto agli standard e alle best practice del settore della sicurezza. Security Hub CSPM raccoglie dati di sicurezza da tutti Account AWS i servizi e dai prodotti partner di terze parti supportati. Quindi analizza le tendenze in materia di sicurezza e identifica i problemi di sicurezza con la massima priorità.

Quando integri IAM Access Analyzer con Security Hub CSPM, puoi inviare i risultati da IAM Access Analyzer a Security Hub CSPM. Security Hub CSPM può quindi includere tali risultati nella sua analisi del livello di sicurezza complessivo.

Indice

- In che modo IAM Access Analyzer invia i risultati al Security Hub CSPM
  - Tipi di risultati inviati da Sistema di analisi degli accessi IAM
  - Latenza per l'invio degli esiti
  - Riprovare quando Security Hub CSPM non è disponibile
  - Aggiornamento degli esiti esistenti CSPM Security Hub
- Visualizzazione dei risultati di IAM Access Analyzer in Security Hub CSPM
  - Interpretazione dei nomi di ricerca di IAM Access Analyzer in Security Hub (CSPM)
- Risultato tipico da Sistema di analisi degli accessi IAM
- Abilitazione e configurazione dell'integrazione
- Come interrompere l'invio degli esiti

In che modo IAM Access Analyzer invia i risultati al Security Hub CSPM

In CSPM Security Hub, i problemi di sicurezza vengono monitorati come esiti. Alcuni risultati derivano da problemi rilevati da altri partner Servizi AWS o da partner di terze parti. Security Hub CSPM dispone anche di una serie di regole che utilizza per rilevare problemi di sicurezza e generare risultati.

CSPM Security Hub fornisce strumenti per gestire gli esiti da tutte queste origini. È possibile visualizzare e filtrare gli elenchi di esiti e visualizzare informazioni dettagliate per ogni esito. Per ulteriori informazioni, consulta Visualizzazione degli esiti nella Guida per l'utente AWS Security Hub . È inoltre possibile monitorare lo stato delle indagini in un risultato. Per ulteriori informazioni, consulta Operazioni sugli esiti nella Guida per l'utente di AWS Security Hub .

Tutti i risultati in Security Hub CSPM utilizzano un formato JSON standard chiamato AWS Security Finding Format (ASFF). L'ASFF include dettagli sull'origine del problema, sulle risorse interessate e sullo stato corrente dell'esito. Per ulteriori informazioni, consulta AWS Security Finding Format (ASFF) nella Guida per l'utente di AWS Security Hub .

AWS Identity and Access Management Access Analyzer è uno di quelli Servizi AWS che invia i risultati al Security Hub CSPM. Per gli accessi inutilizzati, Sistema di analisi degli accessi IAM rileva l'accesso inutilizzato concesso a ruoli o utenti IAM e genera un risultato per ognuno di essi. IAM Access Analyzer invia quindi questi risultati a Security Hub CSPM.

Per gli accessi esterni, Sistema di analisi degli accessi IAM rileva le istruzioni della policy che consentono a un principale esterno l'accesso pubblico o l'accesso multi-account su risorse supportate nell'organizzazione o nell'account. IAM Access Analyzer genera un risultato per l'accesso pubblico e lo invia a Security Hub CSPM. Per l'accesso su più account, IAM Access Analyzer invia un singolo risultato per un principale esterno alla volta a Security Hub CSPM. Se sono presenti più risultati tra account in IAM Access Analyzer, è necessario risolvere il risultato CSPM di Security Hub per il singolo principale esterno prima che IAM Access Analyzer fornisca il successivo risultato tra account. Per un elenco completo dei principali esterni con accesso multi-account al di fuori della zona di attendibilità per il sistema di analisi, è necessario visualizzare i risultati nel Sistema di analisi degli accessi IAM. I dettagli della policy di controllo delle risorse (RCP) sono disponibili nella sezione dei dettagli della risorsa.

Tipi di risultati inviati da Sistema di analisi degli accessi IAM

IAM Access Analyzer invia i risultati a Security Hub CSPM utilizzando il AWS Security Finding Format (ASFF). In ASFF, il `Types` campo fornisce il tipo di esito. I risultati ottenuti da Sistema di analisi degli accessi IAM possono avere i seguenti valori per `Types`.

- Risultati di accesso esterno: esposizione/accesso esterno concesso Effects/Data
- Risultati relativi all'accesso esterno: controlli del software e della configurazione/sicurezza Miglior accesso garantito AWS Practices/External
- Risultati relativi agli accessi non utilizzati: controlli del software e della configurazione/migliore autorizzazione di sicurezza AWS Practices/Unused

- Risultati relativi agli accessi non utilizzati: controlli AWS del software e della configurazione/sicurezza Best IAM Role Practices/Unused
- Risultati di accesso non utilizzati: controlli del software e della configurazione/sicurezza Best IAM User Password AWS Practices/Unused
- Risultati di accesso non utilizzati: controlli del software e della AWS configurazione/sicurezza Best IAM User Access Key Practices/Unused

Latenza per l'invio degli esiti

Quando IAM Access Analyzer crea un nuovo risultato, di solito viene inviato a Security Hub CSPM entro 30 minuti. Tuttavia, ci sono rari casi in cui il Sistema di analisi degli accessi IAM potrebbe non ricevere notifiche in merito a una modifica della policy. Esempio:

- Le modifiche alle impostazioni di accesso pubblico in blocco a livello di account di Amazon S3 possono richiedere fino a 12 ore per essere applicate nel Sistema di analisi degli accessi IAM.
- Le modifiche a una policy di controllo delle risorse (RCP) senza una modifica della policy del bucket non attivano una nuova scansione della risorsa riportata nel risultato. In questo caso, Sistema di analisi degli accessi IAM analizza la policy nuova o aggiornata durante la scansione periodica successiva che avviene entro 24 ore.
- Se si verifica un problema di consegna con la consegna dei AWS CloudTrail log, una modifica della politica potrebbe non innescare una nuova scansione della risorsa segnalata nel risultato.

In questo caso, Sistema di analisi degli accessi IAM analizza la policy nuova o aggiornata durante la scansione periodica successiva.

Riprovare quando Security Hub CSPM non è disponibile

Se Security Hub CSPM non è disponibile, IAM Access Analyzer riprova a inviare i risultati su base periodica.

Aggiornamento degli esiti esistenti CSPM Security Hub

Dopo aver inviato un risultato a Security Hub CSPM, IAM Access Analyzer continua a inviare aggiornamenti per riflettere eventuali osservazioni aggiuntive sull'attività di ricerca a Security Hub CSPM. Questi aggiornamenti vengono riflessi all'interno dello stesso risultato.

Per i risultati degli accessi esterni, il Sistema di analisi degli accessi IAM li raggruppa per risorsa. In Security Hub CSPM, il risultato di una risorsa rimane attivo se almeno uno dei risultati per quella

risorsa è attivo in IAM Access Analyzer. Se tutti i risultati in IAM Access Analyzer per una risorsa vengono archiviati o risolti, viene archiviato anche il risultato CSPM di Security Hub. Il risultato del Security Hub CSPM viene aggiornato quando l'accesso alle policy cambia tra accesso pubblico e accesso tra account. Questo aggiornamento può includere modifiche al tipo, al titolo, alla descrizione e alla gravità del risultato.

Per i risultati degli accessi inutilizzati, il Sistema di analisi degli accessi IAM non li raggruppa per risorsa. Invece, se un risultato di accesso non utilizzato viene risolto in IAM Access Analyzer, viene risolto anche il corrispondente risultato CSPM di Security Hub. Il risultato CSPM di Security Hub viene aggiornato quando aggiorni l'utente o il ruolo IAM che ha generato il risultato di accesso inutilizzato.

Visualizzazione dei risultati di IAM Access Analyzer in Security Hub CSPM

Per visualizzare i risultati di IAM Access Analyzer in Security Hub CSPM, scegli Vedi risultati nella sezione AWS: IAM Access Analyzer della pagina di riepilogo. In alternativa, è possibile scegliere Risultati dal pannello di navigazione. Puoi quindi filtrare i risultati per visualizzare solo AWS Identity and Access Management Access Analyzer i risultati scegliendo il campo Nome prodotto: con un valore di. **`IAM Access Analyzer`**

Interpretazione dei nomi di ricerca di IAM Access Analyzer in Security Hub (CSPM)

AWS Identity and Access Management Access Analyzer invia i risultati a Security Hub CSPM utilizzando il AWS Security Finding Format (ASFF). In ASFF, il campo Tipi fornisce il tipo di risultato. I tipi ASFF utilizzano uno schema di denominazione diverso da. AWS Identity and Access Management Access Analyzer La tabella seguente include dettagli su tutti i tipi di ASFF associati ai AWS Identity and Access Management Access Analyzer risultati così come appaiono in Security Hub CSPM.

Tipo di risultati ASFF	Titolo della ricerca CSPM di Security Hub	Description
Effects/Data Exposure/ ExternalAccesso concesso	<resource ARN>consente l'accesso pubblico	Una politica basata sulle risorse collegata alla risorsa consente l'accesso pubblico alla risorsa a tutte le entità esterne.
Controlli del software e della configurazione/Miglior accesso	<resource ARN> consente l'accesso tra account	Una politica basata sulle risorse collegata alla risorsa

Tipo di risultati ASFF	Titolo della ricerca CSPM di Security Hub	Description
AWS di sicurezza garantito Practices/External		consente l'accesso tra account alle entità esterne all'area di trust per l'analizzatore.
Controlli del software e della configurazione/Migliori pratiche di AWS sicurezza/ Autorizzazioni non utilizzate	<resource ARN> contiene autorizzazioni inutilizzate	Un utente o un ruolo contiene autorizzazioni di servizio e operazioni inutilizzate.
Controlli del software e della configurazione/Sicurezza/Miglior ruolo IAM AWS Practices /Unused	<resource ARN> contiene un ruolo IAM inutilizzato	Un utente o un ruolo contiene un ruolo IAM inutilizzato.
Controlli del software e della AWS configurazione/sicurezza Migliore password utente IAM Practices/Unused	<resource ARN> contiene una password utente IAM inutilizzata	Un utente o un ruolo contiene una password utente IAM inutilizzata.
Controlli del software e della AWS configurazione/sicurezza Best Practices/Unused IAM User Access Key	<resource ARN> contiene una chiave di accesso dell'utente IAM inutilizzata	Un utente o un ruolo contiene una chiave di accesso dell'utente IAM inutilizzata.

Risultato tipico da Sistema di analisi degli accessi IAM

IAM Access Analyzer invia i risultati a Security Hub CSPM utilizzando il AWS Security Finding Format (ASFF).

Ecco un esempio di un tipico risultato da Sistema di analisi degli accessi IAM per i risultati degli accessi esterni.

```
{
    "SchemaVersion": "2018-10-08",
    "Id": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/my-analyzer/
arn:aws:s3:::amzn-s3-demo-bucket",
```

```
    "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/access-analyzer",
    "GeneratorId": "aws/access-analyzer",
    "AwsAccountId": "111122223333",
    "Types": ["Software and Configuration Checks/AWS Security Best Practices/External
 Access Granted"],
    "CreatedAt": "2020-11-10T16:17:47Z",
    "UpdatedAt": "2020-11-10T16:43:49Z",
    "Severity": {
        "Product": 1,
        "Label": "LOW",
        "Normalized": 1
    },
    "Title": "AwsS3Bucket/arn:aws:s3:::amzn-s3-demo-bucket/ allows cross-account
 access",
    "Description": "AWS::S3::Bucket/arn:aws:s3:::amzn-s3-demo-bucket/ allows cross-
account access from AWS 444455556666",
    "Remediation": {
        "Recommendation": {"Text": "If the access isn't intended, it indicates a
 potential security risk. Use the console for the resource to modify or remove the
 policy that grants the unintended access. You can use the Rescan button on the Finding
 details page in the Access Analyzer console to confirm whether the change removed the
 access. If the access is removed, the status changes to Resolved."}
    },
    "SourceUrl": "https://console.aws.amazon.com/access-analyzer/home?region=us-
west-2#/findings/details/dad90d5d-63b4-6575-b0fa-ef9c556ge798",
    "Resources": [
        {
            "Type": "AwsS3Bucket",
            "Id": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Details": {
                "Other": {
                    "External Principal Type": "AWS",
                    "Condition": "none",
                    "Action Granted": "s3:GetObject,s3:GetObjectVersion",
                    "External Principal": "444455556666"
                }
            }
        }
    ],
    "WorkflowState": "NEW",
    "Workflow": {"Status": "NEW"},
    "RecordState": "ACTIVE"
}
```

Ecco un esempio di un tipico risultato da Sistema di analisi degli accessi IAM per i risultati degli accessi inutilizzati.


```
{
    "Findings": [
    {
      "SchemaVersion": "2018-10-08",
      "Id": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/integTestAnalyzer-
DO-NOT-DELETE/arn:aws:iam::111122223333:role/TestRole/UnusedPermissions",
      "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/access-analyzer",
      "ProductName": "IAM Access Analyzer",
      "CompanyName": "AWS",
      "Region": "us-west-2",
      "GeneratorId": "aws/access-analyzer",
      "AwsAccountId": "111122223333",
      "Types": [
        "Software and Configuration Checks/AWS Security Best Practices/Unused
 Permission"
      ],
      "CreatedAt": "2023-09-18T16:29:09.657Z",
      "UpdatedAt": "2023-09-21T20:39:16.651Z",
      "Severity": {
        "Product": 1,
        "Label": "LOW",
        "Normalized": 1
      },
      "Title": "AwsIamRole/arn:aws:iam::111122223333:role/IsengardRole-DO-NOT-DELETE/
 contains unused permissions",
      "Description": "AWS::IAM::Role/arn:aws:iam::111122223333:role/IsengardRole-DO-
NOT-DELETE/ contains unused service and action-level permissions",
      "Remediation": {
        "Recommendation": {
          "Text":"If the unused permissions aren't required, delete the permissions to
 refine access to your account. Use the IAM console to modify or remove the policy that
 grants the unused permissions. If all the unused permissions are removed, the status
 of the finding changes to Resolved."
        }
      },
      "SourceUrl": "https://us-west-2.console.aws.amazon.com/access-analyzer/
home?region=us-west-2#/unused-access-findings?resource=arn%3Aaws%3Aiam%3A
%3A903798373645%3Arole%2FTestRole",
      "ProductFields": {
      "numberOfUnusedActions": "256",
```

```
      "numberOfUnusedServices": "15",
      "resourceOwnerAccount": "111122223333",
      "findingId": "DEMO24d8d-0d3f-4d3d-99f4-299fc8a62ee7",
      "findingType": "UnusedPermission",
      "aws/securityhub/FindingId": "arn:aws:securityhub:us-west-2::product/aws/access-
analyzer/arn:aws:access-analyzer:us-west-2:111122223333:analyzer/integTestAnalyzer-DO-
NOT-DELETE/arn:aws:iam::111122223333:role/TestRole/UnusedPermissions",
      "aws/securityhub/ProductName": "AM Access Analyzer",
      "aws/securityhub/CompanyName": "AWS"
    },
    "Resources": [
    {
      "Type": "AwsIamRole",
      "Id": "arn:aws:iam::111122223333:role/TestRole"
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
    },
  "RecordState": "ARCHIVED",
  "FindingProviderFields": {
    "Severity": {
      "Label": "LOW"
    },
    "Types": [
    "Software and Configuration Checks/AWS Security Best Practices/Unused Permission"
    ]
  }
  }
]
}
```

Abilitazione e configurazione dell'integrazione

Per utilizzare l'integrazione con Security Hub CSPM, è necessario abilitare Security Hub CSPM. Per informazioni su come abilitare Security Hub CSPM, vedere Configurazione di Security Hub nella Guida per l'AWS Security Hub utente.

Quando abiliti sia IAM Access Analyzer che Security Hub CSPM, l'integrazione viene abilitata automaticamente. IAM Access Analyzer inizia immediatamente a inviare i risultati al Security Hub CSPM.

Come interrompere l'invio degli esiti

Per interrompere l'invio degli esiti a CSPM Security Hub, è possibile utilizzare la console CSPM Security Hub o l'API.

Consulta Disabilitazione e abilitazione del flusso dei risultati da un'integrazione (console) o Disabilitazione del flusso di risultati da un'integrazione (API Security Hub, AWS CLI) nella Guida per l'utente di AWS Security Hub .

Registrazione delle chiamate API IAM Access Analyzer con AWS CloudTrail

IAM Access Analyzer è integrato con AWS CloudTrail un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in IAM Access Analyzer. CloudTrail acquisisce tutte le chiamate API per IAM Access Analyzer come eventi. Le chiamate acquisite includono le chiamate dalla console IAM Access Analyzer e le chiamate di codice alle operazioni API di IAM Access Analyzer.

Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per IAM Access Analyzer. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi.

Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata a IAM Access Analyzer, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

Informazioni su IAM Access Analyzer in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in IAM Access Analyzer, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per IAM Access Analyzer, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente

e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le azioni di IAM Access Analyzer vengono registrate CloudTrail e documentate nel riferimento all'API di riferimento di IAM Access Analyzer. Ad esempio, le chiamate a `CreateArchiveRule` e le `CreateAnalyzer` `ListFindings` azioni generano voci nei file di registro. CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Informazioni sulle voci dei file di log di IAM Access Analyzer

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che dimostra l'`CreateAnalyzer`operazione eseguita da una sessione con ruolo presunto denominata «14 `Alice-tempcreds` giugno 2021". La sessione del ruolo è stata emessa dal ruolo denominato `admin-tempcreds`.

```
{
  "eventVersion": "1.05",
  "userIdentity":   {
    "type": "AssumedRole",
    "principalId": "AROAIBKEVSQ6C2EXAMPLE:Alice-tempcreds",
    "arn": "arn:aws:sts::111122223333:assumed-role/admin-tempcreds/Alice-tempcreds",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "true",
        "creationDate": "2021-06-14T22:54:20Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin-tempcreds",
        "accountId": "111122223333",
        "userName": "admin-tempcreds"
      },
     "webIdFederationData": {},
    }
  },
  "eventTime": "2021-06-14T22:57:36Z",
  "eventSource": "access-analyzer.amazonaws.com",
  "eventName": "CreateAnalyzer",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "198.51.100.179",
  "userAgent": "aws-sdk-java/1.12.79 Linux/5.4.141-78.230 OpenJDK_64-
Bit_Server_VM/25.302-b08 java/1.8.0_302 vendor/Oracle_Corporation cfg/retry-mode/
standard",
  "requestParameters": {
    "analyzerName": "test",
    "type": "ACCOUNT",
    "clientToken": "11111111-abcd-2222-abcd-222222222222",
        "tags": {
            "tagkey1": "tagvalue1"
        }
  },
  "responseElements": {
    "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/test"
  },
  "requestID": "22222222-dcba-4444-dcba-333333333333",
```

```
  "eventID": "33333333-bcde-5555-bcde-444444444444",
  "readOnly": false,
  "eventType": "AwsApiCall",,
  "managementEvent": true,
  "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Chiavi di filtro di Sistema di analisi degli accessi IAM

Puoi utilizzare le chiavi filtro riportate di seguito per definire una regola di archiviazione (`CreateArchiveRule`), aggiornare una regola di archiviazione (`UpdateArchiveRule`), recuperare un elenco di risultati (`ListFindings` e `ListFindingsV2`) o recuperare un elenco di risultati di anteprima dell'accesso per una risorsa (`ListAccessPreviewFindings`). Non c'è differenza tra l'utilizzo dell'API IAM e CloudFormation la configurazione delle regole di archiviazione.

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
risorsa	Risorsa	L'ARN identifica in modo univoco la risorsa a cui l'entità principale esterna ha accesso. Per ulteriori informazioni, consulta Amazon resource names (ARNs).	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno Interno Non utilizzato
resourceType	Tipo di risorsa	Il tipo di risorsa a cui l'entità principale esterna ha accesso.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno Interno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
AWS::S3 Bucket | AWS::IAM :Role | AWS::SQS :Queue | AWS::Lambda::Function | AWS::Lambda::LayerVersion |AWS::KMS :Key | AWS::SecretsManager::Secret | AWS::EFS :FileSystem | AWS::EC2 :Snapshot | AWS::ECR :Reposi		**Note** Gli analizzatori degli accessi interni non supportano tutti i tipi di risorse supportati dagli analizzatori degli accessi esterni. Gli analizzatori degli accessi inutilizzati supportano solo utenti e ruoli IAM. Per ulteriori informazioni, consulta					Non utilizzato

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
ry | AWS::RDS :DBSnapshot | AWS::RDS :DBClusterSnapsho | AWS::SNS :Topic | AWS::S3Express::DirectoryBucket | AWS::DynamoDB::Table | AWS::DynamoDB::Stream | AWS::IAM :User		Tipi di risorse di Sistema di analisi degli accessi IAM supportate per gli accessi esterni e interni.					

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
resourceOwnerAccount	Account proprietario della risorsa	L'ID dell' AWS account a 12 cifre che possiede la risorsa. Per ulteriori informazioni, consulta ID account di AWS.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno Interno Non utilizzato
isPublic	Accesso pubblico	Indica se il risultato segnala una risorsa che dispone di una policy che consente l'accesso pubblico.	Booleano	✓ Sì	✓ Sì	✓ Sì	Esterno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
findingType `ExternalAccess` | `UnusedIAMRole` | `UnusedIAMUserAccessKey` | `UnusedIAMUserPassword` | `UnusedPermission` | `InternalAccess`	Tipo di esiti	Il tipo di risultato . Per gli analizzatori degli accessi esterni, il tipo è `ExternalAccess` . Per gli analizzatori degli accessi non utilizzati, il tipo può essere `UnusedIAMRole` , `UnusedIAMUserAccessKey` , `UnusedIAMUserPassword` o `UnusedPermission` . Per gli analizzatori degli accessi interni, il tipo è `InternalAccess` .	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno Interno Non utilizzato

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
resourceControlPolicyRestrizione `APPLIED | APPLICABLE | FAILED_ _EVALUA _RCP | NOT_APPLICABLE`	Limitazione di una policy di controllo delle risorse (RCP)	Il tipo di limitazione applicata dal proprietario della risorsa con una policy di controllo delle risorse (RCP) di Organizations. Per ulteriori informazioni sui valori di questa chiave di filtro, consulta ExternalAccessDetailse InternalAccessDetailsnel riferimento all'API di riferimento di IAM Access Analyzer.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno Interno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
serviceControlPolicyRestriction `APPLIED_SCP | APPLICABLE | FAILED_EVALUATING_SCP | NOT_APPLICABLE`	Limitazione di una policy di controllo dei servizi (SCP)	Il tipo di limitazione applicata da una policy di controllo dei servizi (SCP) di Organizations. Per ulteriori informazioni sui valori di questa chiave di filtro, consulta InternalAccessDetailsIAM Access Analyzer API Reference.	Stringa	✓ Sì	✓ Sì	✓ Sì	Interno
status `ACTIVE | ARCHIVED | RESOLVED`	Stato	Lo stato attuale del risultato.	Stringa	✗ No	✓ Sì	✓ Sì	Esterno Interno Non utilizzato
error	Errore	Indica l'errore segnalato per il risultato.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno Interno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
principal.AWS	AWS Account	L'account che ha concesso l'accesso alla risorsa nel campo `Principal` dell'esito. Inserisci l'ID AWS account a 12 cifre o l'ARN dell'utente o del ruolo esterno AWS . Per ulteriori informazioni, consulta ID account di AWS.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno
principal.Federated	Utente federato	L'ARN dell'identità federata che ha accesso alla risorsa nel risultato. Per ulteriori informazioni, consulta Provider di identità e federazione	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
condizione.aws: PrincipalArn	ARN principale	L'ARN del principale (utente IAM, ruolo o gruppo) indicato come condizione per l'accesso alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	Sì	Sì	Sì	Esterno
condition.aws: ID PrincipalOrg	OrgID principale	L'identificatore dell'organizzazione dell'entità principale indicata come condizione per l'accesso alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	Sì	Sì	Sì	Esterno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
condizione.aws:PrincipalOrgPaths	PrincipalOrgPaths	L'ID dell'organizzazione o dell'unità organizzativa (OU) indicato come condizione per l'accesso alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno
condizione.aws:SourceIp	IP di origine	L'indirizzo IP che consente all'entità principale di accedere alla risorsa quando si utilizza l'indirizzo IP specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	IP address (Indirizzo IP)	✓ Sì	✓ Sì	✓ Sì	Esterno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
condizione.aws:SourceVp	VPC di origine	L'ID VPC che consente l'accesso dell'entità principale alla risorsa quando si utilizza il VPC specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	Sì	Sì	Sì	Esterno
condizione.aws:UserId	ID utente	L'ID utente dell'utente IAM da un account esterno indicato come condizione per l'accesso alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	Sì	Sì	Sì	Esterno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
condizione.aws:VpceAccount	Account VPCE	L'ID account dell'endpoint VPC che consente al principale di accedere alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	Sì	Sì	Sì	Esterno Interno
condizione.aws:SourceVpcArn	Fonte: VPC Arn	L'ARN VPC che consente l'accesso principale alla risorsa quando si utilizza il VPC specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	ARN	Sì	Sì	Sì	Esterno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
condition.aws: IDVpceOrg	OrgID VPEC	L'ID organizzativo per l'endpoint VPC che consente al principale di accedere alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	Sì	Sì	Sì	Esterno Interno
condizione.aws: VpceOrgPaths	VPCE OrgPaths	L'unità organizzativa (UO) per l'endpoint VPC che consente al principale di accedere alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa (elenco)	Sì	Sì	Sì	Esterno Interno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
condition.cognito-identity.amazonav.com:aud	Pubblico di Cognito	L'ID pool di identità di Amazon Cognito specificato come condizione per l'accesso al ruolo IAM nel risultato. Per saperne di più, consulta IAM e AWS STS condition context keys.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno
condition.graph.facebook.com:app_id	ID dell'app di Facebook	L'ID dell'applicazione Facebook (o l'ID del sito) specificato come condizione per consentire l'accesso con la federazione Accedi con Facebook al ruolo IAM nel risultato. Per saperne di più, consulta IAM e AWS STS condition context keys.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
condition.accounts.google.com:aud	Pubblico di Google	L'ID dell'applicazione Google specificato come condizione per l'accesso al ruolo IAM. Per saperne di più, consulta IAM e AWS STS condition context keys.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno
condizione.kms:CallerAccount	ID chiave KMS	L'ID dell' AWS account che possiede l'entità chiamante (utente IAM, ruolo o utente root dell'account) utilizzata dalle chiamate AWS KMS ai servizi. Per ulteriori informazioni, consulta Condition keys for AWS Key Management Service.	Stringa	✓ Sì	✓ Sì	✓ Sì	Esterno

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
condition.www.amazon.com:app_id	ID dell'app Amazon	L'ID dell'applicazione Amazon (o l'ID del sito) specificato come condizione per consentire l'accesso con la federazione Login with Amazon al ruolo. Per ulteriori informazioni, consulta la sezione	Stringa	Sì	Sì	Sì	Esterno
id	ID risultato	L'ID del risultato.	Stringa	No	Sì	Sì	Esterno Interno Non utilizzato

Criterion	Console di gestione AWS campo	Descrizione	Tipo	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso	Tipi di analizzatori supportati
changeType CHANGED | NEW | UNCHANGI		Fornisce un contesto sul modo in cui il risultato dell'anteprima di accesso viene confrontato con l'accesso identificato esistente in Sistema di analisi degli accessi IAM.	Stringa	⊗ No	⊗ No	✓ Sì	Esterno
existingFindingId		L'ID esistente del risultato in Sistema di analisi degli accessi IAM, fornito solo per i risultati esistenti nell'anteprima dell'accesso.	Stringa	⊗ No	⊗ No	✓ Sì	Esterno
existingFindingStatus		Lo stato esistente del risultato in Access Analyzer, fornito solo per i risultati esistenti nell'anteprima dell'accesso.	Stringa	⊗ No	⊗ No	✓ Sì	Esterno

Utilizzo di ruoli collegati ai servizi per AWS Identity and Access Management Access Analyzer

AWS Identity and Access Management Access Analyzer utilizza un ruolo collegato al servizio IAM. Un ruolo collegato ai servizi è un tipo di ruolo IAM univoco collegato direttamente ad Sistema di analisi degli accessi IAM. I ruoli collegati ai servizi sono predefiniti da IAM Access Analyzer e includono tutte le autorizzazioni richieste dalla funzionalità per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato ai servizi semplifica la configurazione di Sistema di analisi degli accessi IAM perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. Sistema di analisi degli accessi IAM definisce le autorizzazioni dei relativi ruoli associati ai servizi e, salvo diversamente definito, solo Sistema di analisi degli accessi IAM potrà assumere i relativi ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta la sezione Servizi AWS che funzionano con IAM e cerca i servizi che riportano Sì nella colonna Ruolo associato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni di ruolo collegate ai servizi per AWS Identity and Access Management Access Analyzer

AWS Identity and Access Management Access Analyzer utilizza il ruolo collegato al servizio denominato AWSServiceRoleForAccessAnalyzer: Allow Access Analyzer per analizzare i metadati delle risorse per l'accesso esterno e per analizzare le attività per identificare gli accessi non utilizzati.

Il ruolo AWSService RoleForAccessAnalyzer collegato ai servizi prevede che i seguenti servizi assumano il ruolo:

- `access-analyzer.amazonaws.com`

La policy delle autorizzazioni del ruolo denominata `AccessAnalyzerServiceRolePolicy` consente ad Sistema di analisi degli accessi IAM di completare le operazioni riportate di seguito su risorse specifiche.

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato al servizio è necessario configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per Sistema di analisi degli accessi IAM

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando abiliti Access Analyzer nell'API Console di gestione AWS o nell' AWS API, IAM Access Analyzer crea automaticamente il ruolo collegato al servizio. Lo stesso ruolo collegato ai servizi viene utilizzato in tutte le regioni in cui abiliti Sistema di analisi degli accessi IAM. Sia l'accesso esterno che i risultati degli accessi inutilizzati utilizzano lo stesso ruolo collegato al servizio.

> **Note**
>
> Sistema di analisi degli accessi AWS IAM è un servizio regionale. Devi abilitare Sistema di analisi degli accessi IAM in ogni regione in modo indipendente.

Se elimini questo ruolo collegato ai servizi, Sistema di analisi degli accessi IAM crea nuovamente il ruolo alla successiva creazione di un analizzatore.

Puoi utilizzare la console IAM anche per creare un ruolo collegato ai servizi con il caso d'uso Access Analyzer. Nella AWS CLI o nell' AWS API, crea un ruolo collegato al servizio con il nome del servizio. `access-analyzer.amazonaws.com` Per ulteriori informazioni, consulta Creazione di un ruolo collegato ai servizi nella Guida per l'utente IAM. Se elimini il ruolo collegato ai servizi, è possibile utilizzare lo stesso processo per crearlo nuovamente.

Modifica di un ruolo collegato ai servizi per Sistema di analisi degli accessi IAM

IAM Access Analyzer non consente di modificare il AWSService RoleForAccessAnalyzer ruolo collegato al servizio. Dopo avere creato un ruolo collegato al servizio, non sarà possibile modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Sistema di analisi degli accessi IAM

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, consigliamo di eliminare il ruolo. In questo modo non hai un'entità non utilizzata che

non viene monitorata o gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

Se IAM Access Analyzer è abilitato in una o più regioni dell'organizzazione AWS Organizations, è necessario eliminare tutti gli analizzatori in tutte le regioni dell'organizzazione prima di tentare di eliminare questo ruolo.

> **Note**
>
> Se Sistema di analisi degli accessi IAM sta utilizzando il ruolo quando provi a eliminare le risorse, è possibile che l'eliminazione non riesca. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse IAM Access Analyzer utilizzate dal ruolo AWSService RoleForAccessAnalyzer

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nella sezione Access reports (Report di accesso) in Access analyzer (Analizzatore accesso) scegliere Analyzers (Analizzatori).
3. Scegli l'analizzatore da cui desideri eliminare le risorse di Strumento di analisi degli accessi IAM associate al ruolo collegato ai servizi.
4. Scegli Elimina.
5. Per confermare l'eliminazione dell'analizzatore, immettere **`delete`**, quindi scegliere Delete (Elimina).

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al AWSService RoleForAccessAnalyzer servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di Sistema di analisi degli accessi IAM

Sistema di analisi degli accessi IAM supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta Regioni ed endpoint di AWS.

Anteprima dell'accesso

Oltre a facilitare l'identificazione delle risorse condivise con un'entità esterna, Sistema di analisi degli accessi AWS IAM consente anche di visualizzare in anteprima i risultati di Sistema di analisi degli accessi IAM prima di implementare le autorizzazioni per le risorse in modo da poter confermare che le modifiche alle policy concedono solo l'accesso multi-account e pubblico previsto alla tua risorsa. In questo modo è possibile iniziare con l'accesso esterno previsto alle risorse.

È possibile visualizzare in anteprima e convalidare l'accesso pubblico e tra account ai propri bucket Amazon S3 nella sezione Console Amazon S3. Puoi anche utilizzare le API IAM Access Analyzer per visualizzare in anteprima l'accesso pubblico e tra più account ai bucket, alle AWS KMS chiavi, ai ruoli IAM, alle code di Amazon SQS e ai segreti di Secrets Manager di Amazon S3 fornendo le autorizzazioni proposte per la tua risorsa.

Argomenti

- Visualizzazione in anteprima dell'accesso nella console Amazon S3
- Visualizzazione in anteprima dell'accesso con IAM Access Analyzer APIs

Visualizzazione in anteprima dell'accesso nella console Amazon S3

Dopo aver completato la policy del bucket nella console Amazon S3, è possibile visualizzare in anteprima l'accesso multi-account e pubblico al proprio bucket Amazon S3. È possibile verificare che le modifiche alle policy concedano solo l'accesso esterno previsto prima di scegliere Salva modifiche. Questa fase facoltativa consente di visualizzare in anteprima i risultati di AWS Identity and Access Management Access Analyzer per il bucket. È possibile verificare se la modifica alle policy introduce nuovi risultati o risolve i risultati esistenti per l'accesso esterno. È possibile saltare questa fase di convalida e salvare la propria policy del bucket Amazon S3 in qualsiasi momento.

Per visualizzare in anteprima l'accesso esterno al bucket, è necessario disporre di un analizzatore account attivo nell'area del bucket con l'account come zona di attendibilità. Devi inoltre disporre delle autorizzazioni necessarie per utilizzare IAM Access Analyzer e l'accesso in anteprima. Per ulteriori informazioni sull'abilitazione di IAM Access Analyzer e sulle autorizzazioni richieste, consulta Nozioni di base su AWS Identity and Access Management Access Analyzer.

Come visualizzare in anteprima l'accesso al bucket Amazon S3 quando si crea o si modifica la policy del bucket

1. Una volta completata la creazione o la modifica della policy del bucket, accertarsi che la policy sia una policy del bucket Amazon S3 valida. L'ARN della policy deve corrispondere all'ARN del bucket e gli elementi delle policy devono essere validi.
2. Sotto la policy, in Anteprima accesso esterno, selezionare un analizzatore di account attivo, quindi scegliere Anteprima. Viene generata un'anteprima dei risultati di IAM Access Analyzer per il bucket. L'anteprima analizza la policy del bucket Amazon S3 visualizzata insieme alle autorizzazioni del bucket esistenti. Ciò include le impostazioni relative al bucket e al BPA dell'account, l'ACL del bucket, i punti di accesso Amazon S3 e i punti di accesso multi-regione collegati al bucket con le relative policy e impostazioni BPA.
3. Al termine dell'anteprima di accesso, viene visualizzata un'anteprima dei risultati di IAM Access Analyzer. Ogni risultato riporta un'istanza di un principale esterno all'account con accesso al bucket dopo aver salvato la policy. È possibile convalidare l'accesso al bucket esaminando ogni risultato. L'intestazione del risultato fornisce un riepilogo dell'accesso ed è possibile espanderla per esaminare i dettagli del risultato. I badge del risultato forniscono un contesto su come il salvataggio della policy del bucket cambierebbe l'accesso al bucket. Ad esempio, consentono verificare se la modifica alle policy introduce nuovi risultati o risolve i risultati esistenti per l'accesso esterno.
    a. Novità: indica un risultato per un nuovo accesso esterno che la policy introdurrebbe.
    b. Risolto: indica un risultato per l'accesso esterno esistente che la policy rimuoverebbe.
    c. Archiviato: indica un risultato per un nuovo accesso esterno che verrebbe archiviato automaticamente in base alle regole di archiviazione per l'analizzatore che definiscono quando i risultati devono essere contrassegnati come previsto.
    d. Esistente: indica un risultato esistente per l'accesso esterno che rimarrebbe invariato.
    e. Pubblico: se un risultato è per l'accesso pubblico alla risorsa, oltre a uno dei badge precedenti avrà un badge Pubblico.
4. Se si identifica un accesso esterno che non si desidera introdurre o rimuovere, è possibile modificare la policy e scegliere di nuovo Anteprima fino a quando non si raggiunge l'accesso esterno desiderato. Se si dispone di un risultato etichettato come Pubblico, si consiglia di rivedere la policy per rimuovere l'accesso pubblico prima di scegliere Salva modifiche. L'anteprima dell'accesso è una fase facoltativa ed è possibile scegliere Salva modifiche in qualsiasi momento.

Visualizzazione in anteprima dell'accesso con IAM Access Analyzer APIs

Puoi utilizzare IAM Access Analyzer APIs per visualizzare in anteprima l'accesso pubblico e tra account per i tuoi bucket Amazon S3 AWS KMS , le chiavi, i ruoli IAM, le code Amazon SQS e i segreti di Secrets Manager. È possibile visualizzare in anteprima l'accesso fornendo le autorizzazioni proposte per una risorsa esistente di cui si è proprietari o per una nuova risorsa che si desidera implementare.

Per visualizzare in anteprima l'accesso esterno alla risorsa, è necessario disporre di un analizzatore account attivo per l'account e la regione della risorsa. Devi inoltre disporre delle autorizzazioni necessarie per utilizzare IAM Access Analyzer e l'accesso in anteprima. Per ulteriori informazioni sull'abilitazione di IAM Access Analyzer e sulle autorizzazioni richieste, consulta Nozioni di base su AWS Identity and Access Management Access Analyzer.

Per visualizzare in anteprima l'accesso a una risorsa, è possibile utilizzare l'operazione `CreateAccessPreview` e fornire l'ARN dell'analizzatore e la configurazione del controllo degli accessi per la risorsa. Il servizio restituisce l'ID univoco per l'anteprima di accesso che è possibile utilizzare per verificare lo stato dell'anteprima di accesso con l'operazione `GetAccessPreview`. Quando lo stato è `Completed`, è possibile utilizzare l'operazione `ListAccessPreviewFindings` per recuperare i risultati generati per l'anteprima dell'accesso. Le operazioni `GetAccessPreview` e `ListAccessPreviewFindings` recupereranno le anteprime di accesso e i risultati creati entro circa 24 ore.

Ogni risultato recuperato contiene i dettagli del risultato che descrivono l'accesso. Uno stato di anteprima del risultato che descrive se il risultato sarebbe `Active`, `Archived` oppure `Resolved` dopo l'implementazione delle autorizzazioni e un `changeType`. Il `changeType` fornisce un contesto sul modo in cui il risultato dell'anteprima di accesso viene confrontato con l'accesso esistente identificato in IAM Access Analyzer.

- Novità: il risultato riguarda l'accesso appena introdotto.
- Invariato: il risultato dell'anteprima è un risultato esistente che rimarrebbe invariato.
- Modificato: il risultato dell'anteprima è un risultato esistente con un cambiamento di stato.

Le operazioni `status` e `changeType` consentono di capire come la configurazione delle risorse modificherebbe l'accesso delle risorse esistenti. Se `changeType` è `Unchanged` o Modificato, il risultato conterrà anche l'ID e lo stato del risultato esistenti in IAM Access Analyzer. Ad esempio, un risultato `Changed` con stato di anteprima `Resolved` e stato esistente `Active` indica che il risultato

`Active` esistente per la risorsa diventerebbe `Resolved` in seguito alla modifica delle autorizzazioni proposta.

È possibile utilizzare l'operazione `ListAccessPreviews` per recuperare un elenco di anteprime di accesso per l'analizzatore specificato. Questa operazione recupererà le informazioni sull'anteprima di accesso creata in un'ora circa.

In generale, se l'anteprima di accesso è per una risorsa esistente e si lascia un'opzione di configurazione non specificata, l'anteprima di accesso utilizzerà la configurazione della risorsa esistente per impostazione predefinita. Se l'anteprima di accesso è relativa a una nuova risorsa e si lascia un'opzione di configurazione non specificata, l'anteprima di accesso utilizzerà il valore predefinito in base al tipo di risorsa. Per i casi di configurazione per ciascun tipo di risorsa, consultare gli argomenti di seguito.

Visualizzazione in anteprima dell'accesso al bucket Amazon S3

Per creare un'anteprima di accesso per un nuovo bucket Amazon S3 o un bucket Amazon S3 esistente di tua proprietà, puoi proporre una configurazione del bucket specificando la policy del bucket Amazon S3, le impostazioni ACLs BPA del bucket, del bucket e i punti di accesso Amazon S3, inclusi i punti di accesso multiregionali, collegati al bucket.

> **Note**
>
> Prima di provare a creare un'anteprima di accesso per un nuovo bucket, ti consigliamo di chiamare l'operazione Amazon HeadBucketS3 per verificare se il bucket denominato esiste già. Questa operazione è utile per determinare se esiste un bucket e si dispone dell'autorizzazione per accedervi.

Policy del bucket: se la configurazione è per un bucket Amazon S3 esistente e non si specifica la policy del bucket Amazon S3, l'anteprima di accesso utilizza la policy esistente allegata al bucket. Se l'anteprima di accesso riguarda una nuova risorsa e non si specifica la policy del bucket di Amazon S3, l'anteprima di accesso presuppone un bucket senza policy. Per proporre l'eliminazione di una policy del bucket esistente, è possibile specificare una stringa vuota. Per ulteriori informazioni sui limiti delle policy del bucket supportati, consultare Esempi di policy di bucket.

Concessioni ACL del bucket: è possibile proporre fino a 100 concessioni ACL per bucket. Se la configurazione di concessione proposta è per un bucket esistente, l'anteprima dell'accesso utilizza

l'elenco proposto di configurazioni di concessioni al posto delle concessioni esistenti. In caso contrario, utilizzerà le concessioni esistenti per il bucket.

Punti di accesso del bucket: l'analisi supporta fino a 100 punti di accesso, inclusi i punti di accesso multi-regione, per bucket, con un massimo di dieci nuovi punti di accesso che è possibile proporre per bucket. Se la configurazione dei punti di accesso di Amazon S3 proposta è per un bucket esistente, l'anteprima dell'accesso utilizza la configurazione dei punti di accessi proposta al posto dei punti di accesso esistenti. Per proporre un punto di accesso senza una policy, è possibile fornire una stringa vuota come policy del punto di accesso. Per ulteriori informazioni sui limiti delle policy dei punti di accesso, consultare Restrizioni e limitazioni dei punti di accesso.

Configurazione dell'accesso pubblico ai blocchi: se la configurazione proposta è per un bucket Amazon S3 esistente e non si specifica la configurazione, l'anteprima dell'accesso utilizza l'impostazione esistente. Se la configurazione proposta riguarda un nuovo bucket e non si specifica la configurazione BPA del bucket, l'anteprima dell'accesso utilizza `false`. Se la configurazione proposta si riferisce a un nuovo punto di accesso o a un punto di accesso multi-regione e non si specifica la configurazione BPA del punto di accesso, l'anteprima dell'accesso utilizza `true`.

Visualizza l'anteprima dell'accesso alla tua chiave AWS KMS

Per creare un'anteprima di accesso per una nuova AWS KMS chiave o per una AWS KMS chiave esistente di tua proprietà, puoi proporre una configurazione delle AWS KMS chiavi specificando la politica della chiave e la configurazione della AWS KMS concessione.

AWS KMS politica chiave: se la configurazione riguarda una chiave esistente e non si specifica la politica della chiave, l'anteprima di accesso utilizza la politica esistente per la chiave. Se l'anteprima di accesso è relativa a una nuova risorsa e non si specifica la policy della chiave, l'anteprima di accesso utilizza la policy della chiave predefinita. La chiave della policy proposta non può essere una stringa vuota.

AWS KMS sovvenzioni: l'analisi supporta fino a 100 concessioni KMS per configurazione*.* Se la configurazione di concessione proposta riguarda una chiave esistente, l'anteprima di accesso utilizza l'elenco proposto di configurazioni di concessione al posto delle sovvenzioni esistenti. In caso contrario, utilizzerà le concessioni esistenti per la chiave.

Anteprima dell'accesso al ruolo IAM

Per creare un'anteprima di accesso per un nuovo ruolo IAM o un ruolo IAM esistente, puoi proporre una configurazione del ruolo IAM specificando la policy di attendibilità.

Policy di attendibilità del ruolo: se la configurazione riguarda un nuovo ruolo IAM, è necessario specificare la policy di attendibilità. Se la configurazione riguarda un ruolo IAM esistente di cui si è proprietari e non propone la policy di attendibilità, l'anteprima di accesso utilizza la policy di attendibilità esistente per il ruolo. La policy proposta non può essere una stringa vuota.

Anteprima dell'accesso alla coda Amazon SQS

Per creare un'anteprima di accesso per una nuova coda Amazon SQS o una coda Amazon SQS esistente di proprietà, è possibile proporre una configurazione di coda Amazon SQS specificando la policy di Amazon SQS per la coda.

Policy della coda Amazon SQS: se la configurazione è per una coda Amazon SQS esistente e non si specifica la policy di Amazon SQS, l'anteprima degli accessi utilizza la policy di Amazon SQS esistente per la coda. Se l'anteprima di accesso riguarda una nuova risorsa e non specifichi la policy, l'anteprima di accesso presuppone una coda Amazon SQS senza policy. Per proporre l'eliminazione di una policy di coda Amazon SQS esistente, è possibile specificare una stringa vuota per la policy di Amazon SQS.

Anteprima dell'accesso al segreto di Secrets Manager

Per creare un'anteprima di accesso per un nuovo segreto di Secrets Manager o un segreto di Secrets Manager esistente di tua proprietà, puoi proporre una configurazione segreta di Secrets Manager specificando la policy segreta e la chiave di AWS KMS crittografia opzionale.

Policy del segreto: se la configurazione è per un segreto esistente e non specifichi la policy del segreto, l'anteprima di accesso utilizza la policy esistente per il segreto. Se l'anteprima di accesso riguarda una nuova risorsa e non specifichi la policy, l'anteprima di accesso presuppone un segreto senza policy. Per proporre l'eliminazione di una policy esistente, puoi specificare una stringa vuota.

AWS KMS chiave di crittografia: se la configurazione proposta riguarda un nuovo segreto e non si specifica l'ID della AWS KMS chiave, l'anteprima di accesso utilizza la chiave KMS predefinita dell' AWS account. Se si specifica una stringa vuota per l'ID della AWS KMS chiave, l'anteprima di accesso utilizza la chiave KMS predefinita dell' AWS account.

Controlli per la convalida delle policy

Sistema di analisi degli accessi IAM fornisce controlli delle policy che aiutano a convalidare le policy IAM prima di collegarle a un'entità. Questi includono i controlli base forniti dalla convalida delle policy

per convalidare la policy rispetto alla sintassi e alle best practice AWS. È possibile visualizzare i risultati del controllo della convalida delle policy che includono avvisi di sicurezza, errori, avvisi generali e suggerimenti per la policy.

Puoi utilizzare i controlli delle policy personalizzati per verificare la presenza di nuovi accessi in base ai tuoi standard di sicurezza. Viene addebitato un costo per ogni controllo di un nuovo accesso. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi IAM.

Come funzionano i controlli delle policy personalizzati

È possibile convalidare le policy rispetto agli standard di sicurezza specificati utilizzando i controlli delle policy personalizzati di AWS Identity and Access Management Access Analyzer . È possibile eseguire i seguenti tipi di controlli delle policy personalizzati:

- Verifica in base a una policy di riferimento: quando si modifica una policy, è possibile controllare se la policy aggiornata concede un nuovo accesso rispetto a quella di riferimento, ad esempio una sua versione esistente. Puoi eseguire questo controllo quando modifichi una policy utilizzando AWS Command Line Interface (AWS CLI), IAM Access Analyzer API (API) o l'editor di policy JSON nella console IAM.

  > **Note**
  >
  > I controlli delle policy personalizzati del Sistema di analisi degli accessi IAM consentono l'inserimento di caratteri jolly nell'elemento `Principal` per le policy delle risorse di riferimento.

- Verifica in base a un elenco di risorse o operazioni IAM: puoi verificare che risorse oppure operazioni IAM specifiche non siano consentite dalla tua policy. Se vengono specificate solo le azioni, Sistema di analisi degli accessi IAM verifica l'accesso delle azioni su tutte le risorse della policy. Se vengono specificate solo risorse, Sistema di analisi degli accessi IAM verifica quali azioni hanno accesso alle risorse specificate. Se vengono specificate sia azioni che risorse, Sistema di analisi degli accessi IAM verifica quali tra le azioni specificate hanno accesso alle risorse specificate. Puoi eseguire questo controllo quando crei o modifichi una policy utilizzando AWS CLI o l'API.
- Verifica dell'accesso pubblico: puoi verificare se una policy delle risorse può concedere l'accesso pubblico a un tipo di risorsa specificato. Puoi eseguire questo controllo quando crei o modifichi una policy utilizzando l'API AWS CLI o. Questo tipo di controllo delle policy personalizzato è diverso dall'anteprima dell'accesso perché non richiede alcun account o contesto di analisi degli accessi

esterni. Le anteprime di accesso consentono di visualizzare in anteprima i risultati del Sistema di analisi degli accessi IAM prima di implementate le autorizzazioni delle risorse, mentre il controllo personalizzato determina se l'accesso pubblico può essere concesso da una policy.

Viene addebitato un costo per ogni controllo della policy personalizzato. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

È possibile eseguire controlli delle policy personalizzati sulle policy basate su identità e risorse. I controlli delle policy personalizzati non si basano su tecniche di corrispondenza dei modelli o sulla verifica dei log di accesso per determinare se un accesso nuovo o specifico è consentito da una policy. Analogamente ai risultati degli accessi esterni, i controlli delle policy personalizzati si basano su Zelkova. Zelkova traduce le policy IAM in istruzioni logiche equivalenti e gestisce una suite di risolutori logici generici e specializzati (teorie dei moduli di soddisfacibilità) per il problema. Per verificare gli accessi nuovi o specifici, Sistema di analisi degli accessi IAM applica ripetutamente Zelkova a una policy. Le query diventano sempre più specifiche per caratterizzare classi di comportamenti consentite dalla policy in base al contenuto della policy. Per ulteriori informazioni sulle teorie dei moduli di soddisfacibilità, consulta Teorie dei moduli di soddisfacibilità.

In rari casi, Sistema di analisi degli accessi IAM non è in grado di determinare completamente se un'istruzione della policy concede un accesso nuovo o specifico. In questi casi, dichiara erroneamente un falso positivo non superando il controllo delle policy personalizzate. Sistema di analisi degli accessi IAM è progettato per fornire una valutazione completa delle policy e si impegna per ridurre al minimo i falsi negativi. Con questo approccio, Sistema di analisi degli accessi IAM garantisce in modo piuttosto certo che un controllo superato significa che l'accesso non è stato concesso dalla policy. Puoi controllare manualmente i controlli non riusciti esaminando l'istruzione della policy riportata nella risposta di Sistema di analisi degli accessi IAM.

Esempi di policy di riferimento per verificare la presenza di nuovi accessi

Puoi trovare esempi di policy di riferimento e imparare come impostare ed eseguire un controllo personalizzato delle policy per i nuovi accessi nell'archivio di esempi di controlli personalizzati delle policy di IAM Access Analyzer su cui si trovano. GitHub

> **Prima di utilizzare questi esempi**
>
> Prima di utilizzare questi esempi di policy di riferimento, esegui queste operazioni:
>
> - Esamina attentamente e personalizza le policy di riferimento per i tuoi requisiti specifici.

- Testa accuratamente le policy di riferimento nel tuo ambiente con i servizi Servizi AWS che utilizzi.

  Le policy di riferimento illustrano l'implementazione e l'utilizzo di controlli delle policy personalizzati. Non devono essere interpretate come suggerimenti o best practice AWS ufficiali da implementare esattamente come mostrato. È tua responsabilità testare accuratamente la sostenibilità delle policy di riferimento per soddisfare i requisiti di sicurezza del tuo ambiente.
- I controlli delle policy personalizzati sono indipendenti dall'ambiente durante l'analisi. La loro analisi prende in considerazione solo le informazioni contenute nelle policy di input. Ad esempio, i controlli delle policy personalizzati non possono verificare se un account è membro di un'organizzazione specifica AWS . Pertanto, non possono confrontare i nuovi accessi in base ai valori delle chiavi di condizione per le chiavi di condizione `aws:PrincipalOrgId`e `aws:PrincipalAccount`.

Ispezione dei controlli delle policy personalizzate non riusciti

Quando un controllo delle policy personalizzate fallisce, la risposta di Sistema di analisi degli accessi IAM include l'ID istruzione (`Sid`) dell'istruzione che ha causato l'esito negativo del controllo. Sebbene l'ID istruzione sia un elemento di policy facoltativo, consigliamo di aggiungere un ID istruzione per ogni istruzione di policy. Il controllo delle policy personalizzato restituisce anche un indice delle istruzioni per aiutare a identificare il motivo dell'errore del controllo. L'indice delle istruzioni segue la numerazione a base zero, in cui la prima istruzione viene indicata come 0. Quando sono presenti più istruzioni che causano l'esito negativo di un controllo, il controllo restituisce un solo ID istruzione alla volta. Consigliamo di correggere l'istruzione evidenziata nel motivo e di eseguire nuovamente il controllo finché non viene superato.

Convalidare le policy con Sistema di analisi degli accessi IAM

Puoi convalidare le tue politiche utilizzando la convalida AWS Identity and Access Management Access Analyzer delle politiche. Puoi creare o modificare una policy utilizzando l' AWS API o AWS CLI l'editor di policy JSON nella console IAM. Sistema di analisi degli accessi IAM convalida la policy rispetto alla sintassi delle policy IAM e alle best practice AWS. È possibile visualizzare i risultati del controllo della convalida delle policy che includono avvisi di sicurezza, errori, avvisi generali e suggerimenti per la policy. Questi risultati forniscono suggerimenti utili che consentono di creare policy funzionali e conformi alle best practice per la sicurezza. Per visualizzare un elenco dei controlli

delle policy di base eseguiti da Sistema di analisi degli accessi IAM, consulta Riferimento per il controllo di convalida delle policy IAM.

Convalida delle policy in IAM (console)

È possibile visualizzare i risultati generati dalla convalida delle policy di Sistema di analisi degli accessi IAM quando si crea o si modifica una policy gestita nella console IAM. È inoltre possibile visualizzare questi risultati per le policy di utenti o ruoli in linea. Sistema di analisi degli accessi IAM non genera questi risultati per le policy di gruppo in linea.

Come visualizzare i risultati generati dai controlli delle policy per le policy JSON IAM

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Inizia a creare o modificare una policy utilizzando uno dei seguenti metodi:
   a. Per creare una nuova policy gestita, visita la pagina Policy e crea una nuova policy. Per ulteriori informazioni, consulta Creazione di policy utilizzando l'editor JSON.
   b. Per visualizzare i controlli della policy per una policy gestita dal cliente esistente, accedi alla pagina Policy, scegli il nome di una policy, quindi scegli Modifica. Per ulteriori informazioni, consulta Modifica di policy gestite dal cliente (console).
   c. Per visualizzare i controlli della policy relativi a una policy inline per un utente o un ruolo, accedi alla pagina Utenti o Ruoli, scegli il nome di un utente o di un ruolo, scegli il nome della policy nella scheda Autorizzazioni, quindi scegli Modifica. Per ulteriori informazioni, consulta Modifica delle policy in linea (console).
3. Sceglie la scheda JSON nell'editor di policy.
4. Nel riquadro di convalida delle policy sotto la policy scegli una o più schede delle seguenti opzioni. I nomi delle schede indicano anche il numero di ciascun tipo di ricerca per la policy.
   - Sicurezza: visualizza gli avvisi se la tua policy consente l'accesso che AWS considera un rischio per la sicurezza perché l'accesso è eccessivamente permissivo.
   - Errori: consente di visualizzare gli errori se la policy include righe che impediscono il funzionamento della policy.
   - Avvisi: visualizza gli avvisi se le policy non sono conformi alle best practice, ma i problemi non costituiscono rischi per la sicurezza.
   - Suggerimenti: visualizza i suggerimenti se AWS consiglia miglioramenti che non influenzano le autorizzazioni delle policy.

5. Esamina i dettagli della ricerca forniti dal controllo delle policy di Sistema di analisi degli accessi IAM. Ogni risultato indica la posizione del problema segnalato. Per ulteriori informazioni sulle cause del problema e su come risolverlo, scegli il collegamento Ulteriori informazioni accanto al risultato. È inoltre possibile ricercare il controllo delle policy associato a ogni ricerca nella pagina di riferimento Controlli delle policy di Access Analyzer.
6. Opzionale. Se stai modificando una policy esistente, puoi eseguire un controllo personalizzato delle policy per determinare se la policy aggiornata concede un nuovo accesso rispetto alla versione esistente. Nel riquadro di convalida delle policy sotto la policy, scegli la scheda Verifica nuovi accessi e seleziona Verifica policy. Se le autorizzazioni modificate concedono un nuovo accesso, l'istruzione verrà evidenziata nel riquadro di convalida della policy. Se non intendi concedere un nuovo accesso, aggiorna le istruzioni di policy e scegli Verifica policy finché non viene rilevato alcun nuovo accesso. Per ulteriori informazioni, consulta Convalidare le policy utilizzando i controlli delle policy personalizzate di Sistema di analisi degli accessi IAM.

   > **Note**
   >
   > Viene addebitato un costo per ogni controllo di un nuovo accesso. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi IAM.

7. Aggiorna la policy per risolvere i risultati.

   > **Important**
   >
   > Esegui accuratamente il test delle policy nuove o modificate prima di implementarle nel flusso di lavoro di produzione.

8. Quando hai terminato, seleziona Successivo. Il Validatore di policy segnala eventuali errori di sintassi non riportati da Sistema di analisi degli accessi IAM.

   > **Note**
   >
   > È possibile alternare le schede Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nella scheda Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

9. Per le nuove policy, nella pagina Rivedi e crea, immettere un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi le Autorizzazioni definite in

questa policy per visualizzare le autorizzazioni concesse dalla policy. Seleziona Create policy (Crea policy) per salvare il proprio lavoro.

Per le policy esistenti, nella pagina Rivedi e crea, controlla le Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy. Seleziona la casella di controllo Imposta questa nuova versione come predefinita per salvare la versione aggiornata come versione predefinita della policy. Quindi scegli Salva le modifiche per salvare il lavoro.

Convalida delle policy utilizzando IAM Access Analyzer (o API)AWS CLI AWS

È possibile visualizzare i risultati generati dalla convalida delle policy di Sistema di analisi degli accessi IAM da AWS Command Line Interface (AWS CLI).

Per visualizzare i risultati generati dalla convalida delle policy (o API) di IAM Access Analyzer AWS CLI AWS

Utilizzare una delle seguenti operazioni:

- AWS CLI: aws accessanalyzer validate-policy
- AWS API: ValidatePolicy

Riferimento per il controllo di convalida delle policy IAM

Puoi convalidare le tue politiche utilizzando la convalida AWS Identity and Access Management Access Analyzer delle politiche. Puoi creare o modificare una policy utilizzando l'editor di policy AWS CLI, AWS API o JSON nella console IAM. Sistema di analisi degli accessi IAM convalida la policy rispetto alla sintassi delle policy IAM e alle best practice AWS. È possibile visualizzare i risultati del controllo della convalida delle policy che includono avvisi di sicurezza, errori, avvisi generali e suggerimenti per la policy. Questi risultati forniscono suggerimenti utili che consentono di creare policy funzionali e conformi alle best practice per la sicurezza. L'elenco dei controlli di base delle policy forniti da Sistema di analisi degli accessi IAM è disponibile di seguito. L'esecuzione dei controlli di convalida delle policy non comporta costi aggiuntivi. Per ulteriori informazioni sulla convalida delle policy tramite l'apposito procedimento, consulta Convalidare le policy con Sistema di analisi degli accessi IAM.

Errore: account ARN non consentito

Codice del problema: ARN_ACCOUNT_NOT_ALLOWED

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
ARN account not allowed: The service {{service}} does not support specifying an account
 ID in the resource ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service {{service}} does not support specifying an account ID in
 the resource ARN."
```

Risoluzione dell'errore

Rimuovi l'ID account dall'ARN della risorsa. Gli ARN delle risorse per alcuni AWS servizi non supportano la specificazione di un ID account.

Ad esempio, Amazon S3 non supporta un ID account come namespace negli ARN del bucket. Il nome di un bucket Amazon S3 è univoco a livello globale e lo spazio dei nomi è condiviso da tutti gli account. AWS Per visualizzare tutti i tipi di risorse disponibili in Amazon S3, consulta Tipi di risorse definiti da Amazon S3 in Service Authorization Reference.

Termini correlati

- Risorse relative alle policy
- Identificatori account
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: regione ARN non consentita

Codice del problema: ARN_REGION_NOT_ALLOWED

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
ARN Region not allowed: The service {{service}} does not support specifying a Region in
 the resource ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service {{service}} does not support specifying a Region in the
 resource ARN."
```

Risoluzione dell'errore

Rimuovi la regione dall'ARN della risorsa. Gli ARN delle risorse per alcuni AWS servizi non supportano la specificazione di una regione.

Ad esempio, IAM è un servizio globale. La parte della regione di un ARN della risorsa IAM viene sempre mantenuta vuota. Le risorse IAM sono globali, come lo è oggi un AWS account. Ad esempio, dopo aver effettuato l'accesso come utente IAM, puoi accedere ai AWS servizi in qualsiasi area geografica.

- Risorse relative alle policy
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: mancata corrispondenza del tipo di dati

Codice del problema: DATA_TYPE_MISMATCH

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Data type mismatch: The text does not match the expected JSON data type {{data_type}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The text does not match the expected JSON data type {{data_type}}."
```

Risoluzione dell'errore

Aggiorna il testo per utilizzare il tipo di dati supportato.

Ad esempio, la chiave di condizione globale di `Version` richiede un tipo di dati `String`. Se specifichi una data o un numero intero, il tipo di dati non corrisponderà.

Termini correlati

- Chiavi della condizione globale
- Elementi della policy JSON IAM: operatori di condizione

Errore: chiavi duplicate con un diverso formato maiuscolo/minuscolo

Codice del problema: DUPLICATE_KEYS_WITH_DIFFERENT_CASE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Duplicate keys with different case: The condition key {{key}} appears more than once
 with different capitalization in the same condition block. Remove the duplicate
 condition keys.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{key}} appears more than once with different
 capitalization in the same condition block. Remove the duplicate condition keys."
```

Risoluzione dell'errore

Esamina le chiavi di condizione simili all'interno dello stesso blocco di condizione e utilizza lo stesso formato maiuscolo/minuscolo per tutte le istanze.

Un blocco di condizione è il testo all'interno dell'elemento `Condition` di una istruzione della policy. I nomi delle chiavi di condizione non distinguono tra maiuscole e minuscole. La distinzione tra

maiuscole e minuscole dei valori delle chiavi di condizione dipende dall'operatore di condizione utilizzato. Per ulteriori informazioni sul formato maiuscolo/minuscolo per le chiavi di condizione, consulta Elementi della policy JSON IAM: Condizione.

Termini correlati

- Condizioni
- Blocco di condizione
- Chiavi della condizione globale
- AWS chiavi relative alle condizioni del servizio

Errore: operazione non valida

Codice del problema: INVALID_ACTION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid action: The action {{action}} does not exist. Did you mean {{valid_action}}?
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The action {{action}} does not exist. Did you mean
 {{valid_action}}?"
```

Risoluzione dell'errore

L'azione specificata non è valida. Ciò può verificarsi se si digita male il prefisso del servizio o il nome dell'operazione. Per alcuni problemi comuni, il controllo delle policy restituisce un'operazione suggerita.

Termini correlati

- Operazioni di policy
- AWS azioni di servizio

AWS politiche gestite con questo errore

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Le seguenti politiche AWS gestite includono azioni non valide nelle relative dichiarazioni politiche. Le operazioni non valide non influenzano le autorizzazioni concesse dalla policy. Quando si utilizza una policy AWS gestita come riferimento per creare una policy gestita, AWS consiglia di rimuovere le azioni non valide dalla policy.

- AmazonEMRFullAccessPolicy_v2
- CloudWatchSyntheticsFullAccess

Errore. account ARN non valido

Codice del problema: INVALID_ARN_ACCOUNT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid ARN account: The resource ARN account ID {{account}} is not valid. Provide a
 12-digit account ID.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The resource ARN account ID {{account}} is not valid. Provide a 12-
digit account ID."
```

Risoluzione dell'errore

Aggiorna l'ID account nell'ARN della risorsa. Gli ID account sono numeri interi a 12 cifre. Per informazioni su come visualizzare l'ID del tuo account, vedi Trovare l'ID AWS del tuo account.

Termini correlati

- Risorse relative alle policy

- Identificatori account
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: prefisso ARN non valido

Codice del problema: INVALID_ARN_PREFIX

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid ARN prefix: Add the required prefix (arn) to the resource ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add the required prefix (arn) to the resource ARN."
```

Risoluzione dell'errore

AWS gli ARN delle risorse devono includere il prefisso richiesto. `arn:`

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: regione ARN non valida

Codice del problema: INVALID_ARN_REGION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid ARN Region: The Region {{region}} is not valid for this resource. Update the
 resource ARN to include a supported Region.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Region {{region}} is not valid for this resource. Update the
 resource ARN to include a supported Region."
```

Risoluzione dell'errore

Il tipo di risorsa non è supportato nella regione specificata. Per una tabella dei AWS servizi supportati in ciascuna regione, consulta la tabella delle regioni.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- Nomi e codici delle regioni

Errore: risorsa ARN non valida

Codice del problema: INVALID_ARN_RESOURCE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid ARN resource: Resource ARN does not match the expected ARN format. Update the
 resource portion of the ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Resource ARN does not match the expected ARN format. Update the
 resource portion of the ARN."
```

Risoluzione dell'errore

L'ARN della risorsa deve corrispondere alle specifiche per i tipi di risorse noti. Per visualizzare il formato ARN previsto per un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. Sceglie il nome del servizio per visualizzarne i tipi di risorse e i formati ARN.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: caso di servizio ARN non valido

Codice del problema: INVALID_ARN_SERVICE_CASE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid ARN service case: Update the service name {{service}} in the resource ARN to
 use all lowercase letters.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Update the service name {{service}} in the resource ARN to use all
 lowercase letters."
```

Risoluzione dell'errore

Il servizio nell'ARN della risorsa deve corrispondere alle specifiche (incluso il formato maiuscolo/minuscolo) per i prefissi del servizio. Per visualizzare il prefisso di un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. Scegli il nome del servizio e individua il suo prefisso nella prima frase.

Termini correlati

- Risorse relative alle policy

- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: tipo di dati di condizione non valido

Codice del problema: INVALID_CONDITION_DATA_TYPE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid condition data type: The condition value data types do not match. Use condition
 values of the same JSON data type.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition value data types do not match. Use condition values of
 the same JSON data type."
```

Risoluzione dell'errore

Il valore nella coppia chiave-valore condizione deve corrispondere al tipo di dati della chiave di condizione e dell'operatore di condizione. Per visualizzare il tipo di dati della chiave di condizione per un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. Scegli il nome del servizio per visualizzarne le chiavi di condizione.

Ad esempio, la chiave di condizione globale di `CurrentTime` supporta l'operatore di condizione `Date`. Se si specifica una stringa o un numero intero per il valore nel blocco di condizione, il tipo di dati non corrisponderà.

Termini correlati

- Condizioni
- Blocco di condizione
- Elementi della policy JSON IAM: operatori di condizione
- Chiavi della condizione globale

- AWS chiavi delle condizioni del servizio

Errore: formato della chiave di condizione non valido

Codice del problema: INVALID_CONDITION_KEY_FORMAT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid condition key format: The condition key format is not valid. Use the format
 service:keyname.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key format is not valid. Use the format
 service:keyname."
```

Risoluzione dell'errore

La chiave nella coppia chiave-valore condizione deve corrispondere alle specifiche del servizio. Per visualizzare le chiavi di condizione per un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. Scegli il nome del servizio per visualizzarne le chiavi di condizione.

Termini correlati

- Condizioni
- Chiavi della condizione globale
- AWS chiavi delle condizioni del servizio

Errore: booleano multiplo della condizione non valida

Codice del problema: INVALID_CONDITION_MULTIPLE_BOOLEAN

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid condition multiple Boolean: The condition key does not support multiple Boolean
 values. Use a single Boolean value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key does not support multiple Boolean values. Use a
 single Boolean value."
```

Risoluzione dell'errore

La chiave nella coppia chiave-valore della condizione prevede un singolo valore booleano. Quando si forniscono più valori booleani, la corrispondenza della condizione potrebbe non restituire i risultati previsti.

Per visualizzare le chiavi di condizione per un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. Scegli il nome del servizio per visualizzarne le chiavi di condizione.

- Condizioni
- Chiavi della condizione globale
- AWS chiavi delle condizioni del servizio

Errore: operatore di condizione non valido

Codice del problema: INVALID_CONDITION_OPERATOR

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid condition operator: The condition operator {{operator}} is not valid. Use a
 valid condition operator.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition operator {{operator}} is not valid. Use a valid
 condition operator."
```

Risoluzione dell'errore

Aggiorna la condizione per utilizzare un operatore di condizione supportato.

Termini correlati

- Elementi della policy JSON IAM: operatori di condizione
- Elemento condizione
- Panoramica delle policy JSON

Errore: effetto non valido

Codice del problema: INVALID_EFFECT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid effect: The effect {{effect}} is not valid. Use Allow or Deny.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The effect {{effect}} is not valid. Use Allow or Deny."
```

Risoluzione dell'errore

Aggiorna l'elemento `Effect` per utilizzare un effetto valido. I valori validi di `Effect` sono **`Allow`** e **`Deny`**.

Termini correlati

- Elemento Effetto
- Panoramica delle policy JSON

Errore: chiave di condizione globale non valida

Codice del problema: INVALID_GLOBAL_CONDITION_KEY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid global condition key: The condition key {{key}} does not exist. Use a valid
 condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{key}} does not exist. Use a valid condition
 key."
```

Risoluzione dell'errore

Aggiorna la chiave di condizione nella coppia chiave-valore della condizione per utilizzare una chiave di condizione globale supportata.

Le chiavi di condizione globali sono chiavi di condizione con un `aws:` prefisso. AWS i servizi possono supportare chiavi di condizione globali o fornire chiavi specifiche del servizio che includono il relativo prefisso di servizio. Ad esempio, le chiavi di condizione IAM includono il prefisso `iam:`. Per ulteriori informazioni, consulta Azioni, risorse e chiavi di condizione per AWS i servizi e scegli il servizio di cui desideri visualizzare le chiavi.

Termini correlati

- Chiavi della condizione globale

Errore: partizione non valida

Codice del problema: INVALID_PARTITION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid partition: The resource ARN for the service {{service}} does not support the
 partition {{partition}}. Use the supported values: {{partitions}}
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The resource ARN for the service {{service}} does not support the
 partition {{partition}}. Use the supported values: {{partitions}}"
```

Risoluzione dell'errore

Aggiorna l'ARN della risorsa per includere una partizione supportata. Se hai incluso una partizione supportata, il servizio o la risorsa potrebbe non supportare la partizione inclusa.

Una partizione è un gruppo di regioni. AWS Ogni AWS account è associato a una partizione. Nelle regioni classiche, utilizza la partizione `aws`. Nelle regioni della Cina, utilizza `aws-cn`.

Termini correlati

- Amazon Resource Name (ARN) - Partizioni

Errore: elemento della policy non valido

Codice del problema: INVALID_POLICY_ELEMENT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid policy element: The policy element {{element}} is not valid.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy element {{element}} is not valid."
```

Risoluzione dell'errore

Aggiorna la policy per includere solo gli elementi della policy JSON supportati.

Termini correlati

- Elementi delle policy JSON

Errore: formato principale non valido

Codice del problema: INVALID_PRINCIPAL_FORMAT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid principal format: The Principal element contents are not valid. Specify a key-value pair in the Principal element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Principal element contents are not valid. Specify a key-value pair in the Principal element."
```

Risoluzione dell'errore

Aggiorna il principale per utilizzare un formato di coppia chiave-valore supportato.

Puoi specificare un principale in una policy basata sulle risorse, ma non in una policy basata sulle identità.

Ad esempio, per definire l'accesso per tutti gli utenti di un AWS account, utilizza il seguente principio nella tua policy:

```
"Principal": { "AWS": "123456789012" }
```

Termini correlati

- Elementi delle policy JSON: principale
- Identity-based politiche e politiche basate sulle risorse

Errore: chiave principale non valida

Codice del problema: INVALID_PRINCIPAL_KEY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid principal key: The principal key {{principal-key}} is not valid.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The principal key {{principal-key}} is not valid."
```

Risoluzione dell'errore

Aggiorna la chiave nella coppia chiave-valore del principale per utilizzare una chiave principale supportata. Le chiavi principali supportate sono le seguenti:

- AWS
- CanonicalUser
- Federato
- Servizio

Termini correlati

- Elemento principale

Errore: regione non valida

Codice del problema: INVALID_REGION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid Region: The Region {{region}} is not valid. Update the condition value to a
 supported Region.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Region {{region}} is not valid. Update the condition value to a
 supported Region."
```

Risoluzione dell'errore

Aggiorna il valore della coppia chiave-valore della condizione per includere una regione supportata. Per una tabella dei AWS servizi supportati in ciascuna regione, consulta la tabella delle regioni.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- Nomi e codici delle regioni

Errore: servizio non valido

Codice del problema: INVALID_SERVICE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid service: The service {{service}} does not exist. Use a valid service name.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service {{service}} does not exist. Use a valid service name."
```

Risoluzione dell'errore

Il prefisso del servizio nell'operazione o nella chiave di condizione deve corrispondere alle specifiche (incluso il formato maiuscolo/minuscolo) per i prefissi del servizio. Per visualizzare il prefisso di un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. Scegli il nome del servizio e individua il suo prefisso nella prima frase.

Termini correlati

- Servizi noti e relative operazioni, risorse e chiavi di condizione

Errore: chiave di condizione del servizio non valida

Codice del problema: INVALID_SERVICE_CONDITION_KEY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid service condition key: The condition key {{key}} does not exist in the service
 {{service}}. Use a valid condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{key}} does not exist in the service {{service}}.
 Use a valid condition key."
```

Risoluzione dell'errore

Aggiorna la chiave nella coppia chiave-valore della condizione per utilizzare una chiave di condizione nota per il servizio. Le chiavi di condizione globali sono chiavi di condizione con un prefisso `aws`. I servizi AWS possono fornire chiavi specifiche del servizio che includono il prefisso del servizio. Per visualizzare il prefisso di un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.

Termini correlati

- Chiavi della condizione globale
- Servizi noti e relative operazioni, risorse e chiavi di condizione

Errore: servizio non valido nell'operazione

Codice del problema: INVALID_SERVICE_IN_ACTION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid service in action: The service {{service}} specified in the action does not
 exist. Did you mean {{service2}}?
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service {{service}} specified in the action does not exist. Did
 you mean {{service2}}?"
```

Risoluzione dell'errore

Il prefisso del servizio nell'operazione deve corrispondere alle specifiche (incluso il formato maiuscolo/minuscolo) per i prefissi del servizio. Per visualizzare il prefisso di un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. Scegli il nome del servizio e individua il suo prefisso nella prima frase.

Termini correlati

- Elemento dell'operazione
- Servizi noti e relative operazioni

Errore: variabile non valida per l'operatore

Codice del problema: INVALID_VARIABLE_FOR_OPERATOR

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid variable for operator: Policy variables can only be used with String and ARN
 operators.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Policy variables can only be used with String and ARN operators."
```

Risoluzione dell'errore

Le variabili di policy possono essere utilizzate nell'elemento `Resource` e per confrontare stringhe nell'elemento `Condition`. Le condizioni supportano le variabili quando si utilizzano operatori stringa o operatori ARN. Gli operatori stringa includono `StringEquals`, `StringLike` e `StringNotLike`. Gli operatori ARN includono `ArnEquals` e `ArnLike`. Non è possibile utilizzare una variabile della policy con altri operatori, ad esempio Numeric, Date, Boolean, Binary, IP Address o Null.

Termini correlati

- Utilizzo delle variabili delle policy nell'elemento Condizione
- Elemento condizione

Errore: versione non valida

Codice del problema: INVALID_VERSION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid version: The version {{version}} is not valid. Use one of the following
 versions: {{versions}}
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The version {{version}} is not valid. Use one of the following
 versions: {{versions}}"
```

Risoluzione dell'errore

L'elemento `Version` policy specifica le regole di sintassi del linguaggio AWS utilizzate per elaborare una policy. Per utilizzare tutte le funzionalità della policy disponibili, includi il seguente elemento `Version` prima dell'elemento `Statement` in tutte le policy.

```
"Version": "2012-10-17"
```

Termini correlati

- Elemento della versione

Errore: errore di sintassi JSON

Codice del problema: JSON_SYNTAX_ERROR

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Json syntax error: Fix the JSON syntax error at index {{index}} line {{line}} column
 {{column}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Fix the JSON syntax error at index {{index}} line {{line}} column
 {{column}}."
```

Risoluzione dell'errore

La policy include un errore di sintassi. Controlla la sintassi JSON.

Termini correlati

- Validatore JSON
- Documentazione di riferimento degli elementi delle policy JSON IAM
- Panoramica delle policy JSON

Errore: errore di sintassi JSON

Codice del problema: JSON_SYNTAX_ERROR

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Json syntax error: Fix the JSON syntax error.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Fix the JSON syntax error."
```

Risoluzione dell'errore

La policy include un errore di sintassi. Controlla la sintassi JSON.

Termini correlati

- Validatore JSON
- Documentazione di riferimento degli elementi delle policy JSON IAM
- Panoramica delle policy JSON

Errore: operazione mancante

Codice del problema: MISSING_ACTION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing action: Add an Action or NotAction element to the policy statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add an Action or NotAction element to the policy statement."
```

Risoluzione dell'errore

AWS Le policy JSON devono includere un elemento `Action` or`NotAction`.

Termini correlati

- Elemento dell'operazione
- NotAction elemento
- Panoramica delle policy JSON

Errore: campo ARN mancante

Codice del problema: MISSING_ARN_FIELD

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing ARN field: Resource ARNs must include at least {{fields}} fields in the
 following structure: arn:partition:service:region:account:resource
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Resource ARNs must include at least {{fields}} fields in the
 following structure: arn:partition:service:region:account:resource"
```

Risoluzione dell'errore

Tutti i campi nell'ARN della risorsa devono corrispondere alle specifiche per i tipi di risorse noti. Per visualizzare il formato ARN previsto per un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. Sceglie il nome del servizio per visualizzarne i tipi di risorse e i formati ARN.

Termini correlati

- Risorse relative alle policy

- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: regione ARN mancante

Codice del problema: MISSING_ARN_REGION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing ARN Region: Add a Region to the {{service}} resource ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a Region to the {{service}} resource ARN."
```

Risoluzione dell'errore

Gli ARN delle risorse per la maggior parte dei AWS servizi richiedono la specificazione di una regione. Per una tabella dei AWS servizi supportati in ciascuna regione, consulta la tabella delle regioni.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- Nomi e codici delle regioni

Errore: effetto mancante

Codice del problema: MISSING_EFFECT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing effect: Add an Effect element to the policy statement with a value of Allow or
 Deny.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add an Effect element to the policy statement with a value of Allow
 or Deny."
```

Risoluzione dell'errore

AWS Le policy JSON devono includere un `Effect` elemento con il valore e. **`Allow` `Deny`**

Termini correlati

- Elemento Effetto
- Panoramica delle policy JSON

Errore: principale mancante

Codice del problema: MISSING_PRINCIPAL

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing principal: Add a Principal element to the policy statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a Principal element to the policy statement."
```

Risoluzione dell'errore

Resource-based le politiche devono includere un `Principal` elemento.

Ad esempio, per definire l'accesso per tutti gli utenti di un AWS account, utilizza il seguente principio nella tua policy:

```
"Principal": { "AWS": "123456789012" }
```

Termini correlati

- Elemento principale
- Identity-based politiche e politiche basate sulle risorse

Errore: qualificatore mancante

Codice del problema: MISSING_QUALIFIER

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing qualifier: The request context key {{key}} has multiple values. Use the
 ForAllValues or ForAnyValue condition key qualifiers in your policy.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The request context key {{key}} has multiple values. Use the
 ForAllValues or ForAnyValue condition key qualifiers in your policy."
```

Risoluzione dell'errore

Nell'elemento `Condition` è possibile creare espressioni in cui utilizzare operatori condizionali ("uguale a", "minore di" e così via) per confrontare le chiavi e i valori della policy rispetto alle chiavi e ai valori del contesto della richiesta. Per le richieste che includono più valori per una singola chiave, è necessario racchiudere le condizioni tra parentesi come un array ("Key2":["Value2A", "Value2B"]). È inoltre necessario utilizzare gli operatori su set `ForAllValues` o `ForAnyValue` con l'operatori di condizione `StringLike`. Questi qualificatori aggiungono funzionalità di operazione set all'operatore della condizione, in modo che sia possibile testare più valori di richieste con più valori di condizione.

Termini correlati

- Chiavi di contesto multivalore
- Elemento condizione

AWS politiche gestite con questo errore

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Le seguenti politiche AWS gestite includono un qualificatore mancante per le chiavi di condizione nelle relative dichiarazioni delle politiche. Quando si utilizza la policy AWS gestita come riferimento per creare una policy gestita dal cliente, AWS consiglia di aggiungere i qualificatori della chiave di `ForAnyValue` condizione `ForAllValues` o all'elemento. `Condition`

- AWSGlueConsoleSageMakerNotebookFullAccess

Errore: risorsa mancante

Codice del problema: MISSING_RESOURCE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing resource: Add a Resource or NotResource element to the policy statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a Resource or NotResource element to the policy statement."
```

Risoluzione dell'errore

Tutte le policy, ad eccezione delle policy di attendibilità dei ruoli, devono includere un elemento `Resource` o `NotResource`.

Termini correlati

- Elemento risorsa
- NotResource elemento
- Identity-based politiche e politiche basate sulle risorse
- Panoramica delle policy JSON

Errore: istruzione mancante

Codice del problema: MISSING_STATEMENT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing statement: Add a statement to the policy
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a statement to the policy"
```

Risoluzione dell'errore

Una policy JSON deve includere un'istruzione.

Termini correlati

- Elementi delle policy JSON

Errore: null con if esiste

Codice del problema: NULL_WITH_IF_EXISTS

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Null with if exists: The Null condition operator cannot be used with the IfExists
 suffix. Update the operator or the suffix.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Null condition operator cannot be used with the IfExists suffix.
 Update the operator or the suffix."
```

Risoluzione dell'errore

È possibile aggiungere `IfExists` alla fine di qualsiasi nome dell'operatore di condizione ad eccezione dell'operatore di condizione `Null`. Utilizza un operatore di condizione `Null` per verificare se una chiave di condizione è presente al momento dell'autorizzazione. Utilizza `...ifExists` per dichiarare che "Se la chiave di policy è presente nel contesto della richiesta, la chiave deve essere elaborata come specificato nella policy. Se la chiave non è presente, l'elemento della condizione viene valutato come "true".

Termini correlati

- ... IfExists operatori di condizione
- Operatore di condizione null
- Elemento condizione

Errore: carattere jolly dell'operazione con errore di sintassi SCP

Codice del problema: SCP_SYNTAX_ERROR_ACTION_WILDCARD

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
SCP syntax error action wildcard: SCP actions can include wildcards (*) only at the end
 of a string. Update {{action}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "SCP actions can include wildcards (*) only at the end of a string.
 Update {{action}}."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo del servizio (SCP) supportano la specificazione di valori negli `Action` elementi or. `NotAction` Tuttavia, questi valori possono includere caratteri jolly (*) solo alla fine della stringa. Ciò significa che puoi specificare `iam:Get*` ma non `iam:*role`.

Per specificare più azioni, AWS consiglia di elencarle singolarmente.

Termini correlati

- Azione ed NotAction elementi SCP
- Valutazione SCP
- AWS Organizations politiche di controllo del servizio
- Elementi delle policy JSON IAM: Action

Errore: principale dell'errore di sintassi SCP

Codice del problema: SCP_SYNTAX_ERROR_PRINCIPAL

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
SCP syntax error principal: SCPs do not support specifying principals. Remove the
 Principal or NotPrincipal element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "SCPs do not support specifying principals. Remove the Principal or
 NotPrincipal element."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo del servizio (SCP) non supportano gli elementi `Principal` or`NotPrincipal`.

Puoi specificare l'Amazon Resource Name (ARN) utilizzando la chiave di condizione globale `aws:PrincipalArn` nell'elemento `Condition`.

Termini correlati

- Sintassi delle SCP
- Chiavi di condizione globali per i principali

Errore: sid univoci richiesti

Codice del problema: UNIQUE_SIDS_REQUIRED

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unique Sids required: Duplicate statement IDs are not supported for this policy type.
 Update the Sid value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Duplicate statement IDs are not supported for this policy type.
 Update the Sid value."
```

Risoluzione dell'errore

Per alcuni tipi di policy , gli ID delle istruzioni devono essere univoci. L'elemento `Sid` (ID istruzione) consente di immettere un identificatore opzionale fornito per l'istruzione della policy. Puoi assegnare un valore ID di istruzione a ogni istruzione in una matrice di istruzioni utilizzando l'elemento `SID`. Nei servizi che consentono di specificare un elemento ID, ad esempio SQS e SNS, il valore `Sid` è semplicemente un ID secondario dell'ID del documento di policy. Ad esempio, in IAM il valore `Sid` deve essere univoco all'interno di una policy JSON.

Termini correlati

- Elementi delle policy JSON IAM: Sid

Errore: operazione non supportata nella policy

Codice del problema: UNSUPPORTED_ACTION_IN_POLICY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported action in policy: The action {{action}} is not supported for the resource-
based policy attached to the resource type {{resourceType}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The action {{action}} is not supported for the resource-based policy
 attached to the resource type {{resourceType}}."
```

Risoluzione dell'errore

Alcune operazioni non sono supportate nell'elemento `Action` nella policy basata su risorse collegato a un tipo di risorsa diverso. Ad esempio, AWS Key Management Service le azioni non sono supportate nelle policy dei bucket di Amazon S3. Specificare un'operazione supportata dal tipo di risorsa collegato alla policy basata su risorse.

Termini correlati

- Elementi delle policy JSON: Action

Errore: combinazione di elementi non supportata

Codice del problema: UNSUPPORTED_ELEMENT_COMBINATION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported element combination: The policy elements {{element1}} and {{element2}} can
 not be used in the same statement. Remove one of these elements.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy elements {{element1}} and {{element2}} can not be used in
 the same statement. Remove one of these elements."
```

Risoluzione dell'errore

Alcune combinazioni di elementi delle policy JSON non possono essere utilizzate insieme. Ad esempio, non è possibile utilizzare `Action` e `NotAction` nella stessa dichiarazione di policy. Altre coppie che si escludono reciprocamente sono `Principal/NotPrincipal` e `Resource/NotResource`.

Termini correlati

- Documentazione di riferimento degli elementi delle policy JSON IAM
- Panoramica delle policy JSON

Errore: chiave di condizione globale non supportata

Codice del problema: UNSUPPORTED_GLOBAL_CONDITION_KEY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported global condition key: The condition key aws:ARN is not supported. Use
 aws:PrincipalArn or aws:SourceArn instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key aws:ARN is not supported. Use aws:PrincipalArn or
 aws:SourceArn instead."
```

Risoluzione dell'errore

AWS non supporta l'utilizzo della chiave di condizione globale specificata. A seconda del tuo caso d'uso, puoi utilizzare le chiavi di condizione globali `aws:PrincipalArn` o `aws:SourceArn`. Ad esempio, invece di `aws:ARN` utilizza `aws:PrincipalArn` per confrontare l'Amazon Resource Name (ARN del principale che ha effettuato la richiesta con l'ARN specificato nella policy. In alternativa, utilizza la chiave di condizione globale `aws:SourceArn` per confrontare l'Amazon Resource Name (ARN) della risorsa che effettua una richiesta da servizio a servizio con l'ARN specificato nella policy.

Termini correlati

- AWS chiavi di contesto della condizione globale

Errore: principale non supportato

Codice del problema: UNSUPPORTED_PRINCIPAL

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported principal: The policy type {{policy_type}} does not support the Principal
 element. Remove the Principal element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy type {{policy_type}} does not support the Principal
 element. Remove the Principal element."
```

Risoluzione dell'errore

L'elemento `Principal` specifica il principale a cui è consentito o rifiutato l'accesso a una risorsa. Non è possibile utilizzare l'elemento `Principal` in una policy basata sull'identità IAM. È possibile utilizzarlo nelle politiche di fiducia per i ruoli IAM e nelle politiche basate sulle risorse. Resource-based le policy sono policy che puoi incorporare direttamente in una risorsa. Ad esempio, puoi incorporare le policy in un bucket Amazon S3 o in una AWS chiave KMS.

Termini correlati

- AWS Elementi delle policy JSON: principale
- Accesso alle risorse multi-account in IAM

Errore: ARN della risorsa non supportata nella policy

Codice del problema: UNSUPPORTED_RESOURCE_ARN_IN_POLICY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported resource ARN in policy: The resource ARN is not supported for the resource-
based policy attached to the resource type {{resourceType}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The resource ARN is not supported for the resource-based policy
 attached to the resource type {{resourceType}}."
```

Risoluzione dell'errore

Alcuni ARN delle risorse non sono supportati nell'elemento `Resource` della policy basata sulle risorse quando la policy è collegata a un tipo di risorsa diverso. Ad esempio, gli AWS KMS ARN non sono supportati nell'`Resource`elemento per le policy dei bucket di Amazon S3. Specificare un ARN della risorsa supportato da un tipo di risorsa collegato alla policy basata sulle risorse.

Termini correlati

- Elementi delle policy JSON: Action

Errore: sid non supportato

Codice del problema: UNSUPPORTED_SID

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported Sid: Update the characters in the Sid element to use one of the following
 character types: [a-z, A-Z, 0-9]
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Update the characters in the Sid element to use one of the following
 character types: [a-z, A-Z, 0-9]"
```

Risoluzione dell'errore

L'elemento `Sid` supporta lettere maiuscole, lettere minuscole e numeri.

Termini correlati

- Elementi delle policy JSON IAM: Sid

Errore: carattere jolly non supportato nel principale

Codice del problema: UNSUPPORTED_WILDCARD_IN_PRINCIPAL

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported wildcard in principal: Wildcards (*, ?) are not supported with the
 principal key {{principal_key}}. Replace the wildcard with a valid principal value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Wildcards (*, ?) are not supported with the principal key
 {{principal_key}}. Replace the wildcard with a valid principal value."
```

Risoluzione dell'errore

La struttura dell'elemento `Principal` supporta l'utilizzo di una coppia chiave-valore. Il valore del principale specificato nella policy include un carattere jolly (*). Non è possibile includere un carattere jolly con la chiave del principale specificata. Ad esempio, quando specifichi gli utenti in un elemento `Principal`, non è possibile utilizzare un carattere jolly che indica "tutti gli utenti". Assegna un nome a uno o più utenti specifici. Allo stesso modo, quando si specifica una sessione con assunzione di ruolo, non è possibile utilizzare un carattere jolly (*) per indicare "tutte le sessioni". È necessario assegnare un nome a una sessione specifica. Non è possibile utilizzare un carattere jolly per associare parte di un nome o di un ARN.

Per risolvere questo risultato, rimuovi il carattere jolly e fornisci un principale più specifico.

Termini correlati

- AWS Elementi delle policy JSON: principale

Errore: parentesi mancante nella variabile

Codice del problema: MISSING_BRACE_IN_VARIABLE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing brace in variable: The policy variable is missing a closing curly brace. Add }
 after the variable text.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy variable is missing a closing curly brace. Add } after
 the variable text."
```

Risoluzione dell'errore

La struttura delle variabili delle policy supporta l'utilizzo di un prefisso $ seguito da una coppia di parentesi graffe ({ }). All'interno dei caratteri ${ }, includi il nome del valore ricavato dalla richiesta da utilizzare nella policy.

Per risolvere questo risultato, aggiungi la parentesi graffa mancante per assicurarti che sia presente il set completo di parentesi graffe di apertura e chiusura.

Termini correlati

- Elementi delle policy IAM: variabili

Errore: virgoletta mancante nella variabile

Codice del problema: MISSING_QUOTE_IN_VARIABLE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing quote in variable: The policy variable default value must begin and end with a
 single quote. Add the missing quote.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy variable default value must begin and end with a single
 quote. Add the missing quote."
```

Risoluzione dell'errore

Quando aggiungi una variabile alla policy, puoi specificare un valore di default per la variabile. Se una variabile non è presente, AWS utilizza il testo predefinito fornito.

Per aggiungere un valore di default a una variabile, racchiudi il valore di default tra virgolette singole (`' '`) e separa il testo della variabile e il valore di default con una virgola e uno spazio (`, `).

Ad esempio, se un principale è contrassegnato con `team=yellow`, possono accedere al bucket Amazon S3 `amzn-s3-demo-bucket` con il nome`amzn-s3-demo-bucket-yellow`. Una policy con questa risorsa potrebbe consentire ai membri del team di accedere alle proprie risorse, ma non a quelle di altri team. Per gli utenti senza tag dei team, puoi impostare un valore di default di`company-wide`. Questi utenti possono accedere solo al bucket `amzn-s3-demo-bucket-company-wide`, dove possono visualizzare informazioni generali, come le istruzioni per entrare a far parte di un team.

```
"Resource":"arn:aws:s3:::amzn-s3-demo-bucket-${aws:PrincipalTag/team, 'company-wide'}"
```

Termini correlati

- Elementi delle policy IAM: variabili

Errore: spazio non supportato nella variabile

Codice del problema: UNSUPPORTED_SPACE_IN_VARIABLE

Tipo di esito: ERROR

Ricerca dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported space in variable: A space is not supported within the policy variable
 text. Remove the space.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "A space is not supported within the policy variable text. Remove the
 space."
```

Risoluzione dell'errore

La struttura delle variabili delle policy supporta l'utilizzo di un prefisso $ seguito da una coppia di parentesi graffe ({ }). All'interno dei caratteri ${ }, includi il nome del valore ricavato dalla richiesta da utilizzare nella policy. Sebbene sia possibile includere uno spazio quando si specifica una variabile di default, non è possibile includere uno spazio nel nome della variabile.

Termini correlati

- Elementi delle policy IAM: variabili

Errore: variabile vuota

Codice del problema: EMPTY_VARIABLE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty variable: Empty policy variable. Remove the ${ } variable structure or provide a
 variable within the structure.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Empty policy variable. Remove the ${ } variable structure or provide
 a variable within the structure."
```

Risoluzione dell'errore

La struttura delle variabili delle policy supporta l'utilizzo di un prefisso $ seguito da una coppia di parentesi graffe ({ }). All'interno dei caratteri ${ }, includi il nome del valore ricavato dalla richiesta da utilizzare nella policy.

Termini correlati

- Elementi delle policy IAM: variabili

Errore: variabile non supportata nell'elemento

Codice del problema: VARIABLE_UNSUPPORTED_IN_ELEMENT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Variable unsupported in element: Policy variables are supported in the Resource and
 Condition elements. Remove the policy variable {{variable}} from this element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Policy variables are supported in the Resource and Condition
 elements. Remove the policy variable {{variable}} from this element."
```

Risoluzione dell'errore

Le variabili di policy possono essere utilizzate nell'elemento `Resource` e per confrontare stringhe nell'elemento `Condition`.

Termini correlati

- Elementi delle policy IAM: variabili

Errore: variabile non supportata nella versione

Codice del problema: VARIABLE_UNSUPPORTED_IN_VERSION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Variable unsupported in version: To include variables in your policy, use the policy
 version 2012-10-17 or later.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "To include variables in your policy, use the policy version
 2012-10-17 or later."
```

Risoluzione dell'errore

Per usare le variabili di policy, è necessario che l'elemento `Version` sia incluso in una istruzione e impostato su una versione che supporti le variabili di policy. Le variabili sono state introdotte a partire dalla versione `2012-10-17`. Le versioni precedenti del linguaggio di policy non supportano le variabili. Se non imposti la `Version` su `2012-10-17` o una versione successiva, le variabili come `${aws:username}` nella policy vengono trattate come stringhe letterali.

Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Una versione della policy viene creata quando si modifica una policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita.

Termini correlati

- Elementi delle policy IAM: variabili
- Elementi delle policy JSON IAM: Version

Errore: indirizzo IP privato

Codice del problema: PRIVATE_IP_ADDRESS

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Private IP address: aws:SourceIp works only for public IP address ranges. The values
 for condition key aws:SourceIp include only private IP addresses and will not have the
 desired effect. Update the value to include only public IP addresses.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "aws:SourceIp works only for public IP address ranges. The values
 for condition key aws:SourceIp include only private IP addresses and will not have the
 desired effect. Update the value to include only public IP addresses."
```

Risoluzione dell'errore

La chiave di condizione globale `aws:SourceIp` funziona solo per intervalli di indirizzi IP pubblici. Questo errore viene visualizzato quando la policy consente solo indirizzi IP privati. In questo caso, la condizione non corrisponderà mai.

- aws: chiave di condizione SourceIp globale
- Elementi della policy JSON IAM: Condition

Errore: privato NotIpAddress

Codice del problema: PRIVATE_NOT_IP_ADDRESS

Tipo di esito: ERROR

Ricerca dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Private NotIpAddress: The values for condition key aws:SourceIp include only private
 IP addresses and has no effect. aws:SourceIp works only for public IP address ranges.
 Update the value to include only public IP addresses.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The values for condition key aws:SourceIp include only private IP
 addresses and has no effect. aws:SourceIp works only for public IP address ranges.
 Update the value to include only public IP addresses."
```

Risoluzione dell'errore

La chiave di condizione globale `aws:SourceIp` funziona solo per intervalli di indirizzi IP pubblici. Questo errore viene visualizzato quando si utilizza l'operatore di condizione `NotIpAddress` e sono riportati solo gli indirizzi IP privati. In questo caso, la condizione corrispondere sempre ed è inefficace.

- aws: chiave di condizione SourceIp globale
- Elementi della policy JSON IAM: Condition

Errore: la dimensione della policy supera la quota della SCP

Codice del problema: POLICY_SIZE_EXCEEDS_SCP_QUOTA

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Policy size exceeds SCP quota: The {{policySize}} characters in the service control
 policy (SCP) exceed the {{policySizeQuota}} character maximum for SCPs. We recommend
 that you use multiple granular policies.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{policySize}} characters in the service control policy (SCP)
 exceed the {{policySizeQuota}} character maximum for SCPs. We recommend that you use
 multiple granular policies."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo del servizio (SCP) supportano la specificazione di valori negli `Action` elementi or. `NotAction` Tuttavia, questi valori possono includere caratteri jolly (*) solo alla fine della stringa. Ciò significa che puoi specificare `iam:Get*` ma non `iam:*role`.

Per specificare più azioni, AWS consiglia di elencarle singolarmente.

Termini correlati

- Quote per le organizzazioni AWS
- AWS Organizations politiche di controllo dei servizi

Errore: formato principale del servizio non valido

Codice del problema: INVALID_SERVICE_PRINCIPAL_FORMAT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid service principal format: The service principal does not match the expected
 format. Use the format {{expectedFormat}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service principal does not match the expected format. Use the
 format {{expectedFormat}}."
```

Risoluzione dell'errore

Il valore nella coppia chiave-valore della condizione deve corrispondere a un formato di principale di servizio definito.

Un'entità servizio è un identificatore che viene utilizzato per concedere autorizzazioni a un servizio. Puoi specificare un principale di servizio nell'elemento `Principal` o come valore per alcune chiavi di condizione globali e chiavi specifiche del servizio. Il principale del servizio è definito da ciascun servizio.

L'identificatore di un principale del servizio include il nome del servizio ed è solitamente nel formato seguente con tutte le lettere minuscole:

service-name.amazonaws.com

Alcune chiavi specifiche del servizio possono utilizzare un formato diverso per i principali di servizio. Ad esempio, la chiave di condizione `kms:ViaService` richiede il seguente formato per i principali del servizio con tutte le lettere minuscole:

service-name.AWS_region.amazonaws.com

Termini correlati

- Principali del servizio
- AWS chiavi di condizione globali
- Chiave di condizione `kms:ViaService`

Errore: chiave di tag mancante nella condizione

Codice del problema: MISSING_TAG_KEY_IN_CONDITION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing tag key in condition: The condition key {{conditionKeyName}} must include a tag
 key to control access based on tags. Use the format {{conditionKeyName}}tag-key and
 specify a key name for tag-key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{conditionKeyName}} must include a tag key to
 control access based on tags. Use the format {{conditionKeyName}}tag-key and specify a
 key name for tag-key."
```

Risoluzione dell'errore

Per controllare gli accessi in base ai tag, devi fornire informazioni sui tag nell'elemento condizione di una policy.

Ad esempio, per controllare l'accesso alle AWS risorse, includi la chiave di `aws:ResourceTag` condizione. Questa chiave richiede il formato `aws:ResourceTag/`*`tag-key`*. Per specificare la chiave di tag `owner` e il valore del tag `JaneDoe` In una condizione, utilizza il seguente formato:

```
"Condition": {
    "StringEquals": {"aws:ResourceTag/owner": "JaneDoe"}
}
```

Termini correlati

- Controllo degli accessi tramite tag
- Condizioni
- Chiavi della condizione globale
- AWS chiavi delle condizioni di servizio

Errore: formato vpc non valido

Codice del problema: INVALID_VPC_FORMAT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid vpc format: The VPC identifier in the condition key value is not valid. Use the
 prefix 'vpc-' followed by 8 or 17 alphanumeric characters.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The VPC identifier in the condition key value is not valid. Use the
 prefix 'vpc-' followed by 8 or 17 alphanumeric characters."
```

Risoluzione dell'errore

La chiave di condizione `aws:SourceVpc` deve utilizzare il prefisso `vpc-` seguito da 8 o 17 caratteri alfanumerici, ad esempio `vpc-11223344556677889` o `vpc-12345678`.

Termini correlati

- AWS chiavi di condizione globali: aws: SourceVpc

Errore: formato vpce non valido

Codice del problema: INVALID_VPCE_FORMAT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid vpce format: The VPCE identifier in the condition key value is not valid.  Use
 the prefix 'vpce-' followed by 8 or 17 alphanumeric characters.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The VPCE identifier in the condition key value is not valid.  Use
 the prefix 'vpce-' followed by 8 or 17 alphanumeric characters."
```

Risoluzione dell'errore

La chiave di condizione `aws:SourceVpce` deve utilizzare il prefisso `vpce-` seguito da 8 o 17 caratteri alfanumerici, ad esempio `vpce-11223344556677889` o `vpce-12345678`.

Termini correlati

- AWS chiavi di condizione globali: aws: SourceVpce

Errore: principale federato non supportato

Codice del problema: FEDERATED_PRINCIPAL_NOT_SUPPORTED

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Federated principal not supported: The policy type does not support a federated
 identity provider in the principal element. Use a supported principal.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy type does not support a federated identity provider in
 the principal element. Use a supported principal."
```

Risoluzione dell'errore

L'elemento `Principal` utilizza i principali federati per le policy di attendibilità collegate ai ruoli IAM per fornire l'accesso tramite la federazione delle identità. Le policy di identità e altre policy basate sulle risorse non supportano un provider di identità federato nell'elemento `Principal`. Ad esempio, non puoi utilizzare un principale SAML in una policy bucket Amazon S3. Modifica l'elemento `Principal` con un tipo di principale supportato.

Termini correlati

- Creazione di un ruolo per la federazione delle identità
- Elementi delle policy JSON: principale

Errore: operazione non supportata per la chiave di

Codice del problema: UNSUPPORTED_ACTION_FOR_CONDITION_KEY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported action for condition key: The following actions: {{actions}} are not
 supported by the condition key {{key}}. The condition will not be evaluated for these
 actions. We recommend that you move these actions to a different statement without
 this condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The following actions: {{actions}} are not supported by the
 condition key {{key}}. The condition will not be evaluated for these actions. We
 recommend that you move these actions to a different statement without this condition
 key."
```

Risoluzione dell'errore

Assicurati che la chiave di condizione sia nell'elemento `Condition` della dichiarazione di policy si applichi a tutte le operazioni nell'elemento `Action`. Per garantire che le operazioni specificate siano effettivamente consentite o rifiutate dalla policy, è necessario spostare le operazioni non supportate in una dichiarazione diversa senza la chiave di condizione.

> **Note**
>
> Se l'elemento `Action` ha operazioni con caratteri jolly, Sistema di analisi degli accessi IAM non le valuta per questo errore.

Termini correlati

- Elementi delle policy JSON: Action

Errore: operazione non supportata nella policy

Codice del problema: UNSUPPORTED_ACTION_IN_POLICY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported action in policy: The action {{action}} is not supported for the resource-
based policy attached to the resource type {{resourceType}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The action {{action}} is not supported for the resource-based policy
 attached to the resource type {{resourceType}}."
```

Risoluzione dell'errore

Alcune operazioni non sono supportate nell'elemento `Action` nella policy basata su risorse collegato a un tipo di risorsa diverso. Ad esempio, AWS Key Management Service le azioni non sono supportate nelle policy dei bucket di Amazon S3. Specificare un'operazione supportata dal tipo di risorsa collegato alla policy basata su risorse.

Termini correlati

- Elementi delle policy JSON: Action

Errore: ARN della risorsa non supportata nella policy

Codice del problema: UNSUPPORTED_RESOURCE_ARN_IN_POLICY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported resource ARN in policy: The resource ARN is not supported for the resource-
based policy attached to the resource type {{resourceType}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The resource ARN is not supported for the resource-based policy
 attached to the resource type {{resourceType}}."
```

Risoluzione dell'errore

Alcuni ARN delle risorse non sono supportati nell'elemento `Resource` della policy basata sulle risorse quando la policy è collegata a un tipo di risorsa diverso. Ad esempio, gli AWS KMS ARN non sono supportati nell'`Resource`elemento per le policy dei bucket di Amazon S3. Specificare un ARN della risorsa supportato da un tipo di risorsa collegato alla policy basata sulle risorse.

Termini correlati

- Elementi delle policy JSON: Action

Errore: chiave di condizione non supportata per il principale del servizio

Codice del problema: UNSUPPORTED_CONDITION_KEY_FOR_SERVICE_PRINCIPAL

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unsupported condition key for service principal: The following condition keys are not
 supported when used with the service principal: {{conditionKeys}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The following condition keys are not supported when used with the
 service principal: {{conditionKeys}}."
```

Risoluzione dell'errore

È possibile specificare Servizi AWS nell'`Principal`elemento di una politica basata sulle risorse utilizzando un'entità di servizio, che è un identificatore del servizio. Non è possibile utilizzare alcune chiavi di condizione con determinati principali del servizio. Ad esempio, non puoi utilizzare la chiave

di condizione `aws:PrincipalOrgID` con il principale del servizio `cloudfront.amazonaws.com`. È necessario rimuovere le chiavi di condizione che non si applicano al principale del servizio nell'elemento `Principal`.

Termini correlati

- Principali del servizio
- Elementi delle policy JSON: principale

Errore: errore di sintassi notprincipal della policy di attendibilità del ruolo

Codice del problema: ROLE_TRUST_POLICY_SYNTAX_ERROR_NOTPRINCIPAL

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Role trust policy syntax error notprincipal: Role trust policies do not support
 NotPrincipal. Update the policy to use a Principal element instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Role trust policies do not support NotPrincipal. Update the policy
 to use a Principal element instead."
```

Risoluzione dell'errore

Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo IAM. Le policy di attendibilità definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono assumere il ruolo. Le politiche di attendibilità dei ruoli non supportano `NotPrincipal`. Aggiornare la policy per utilizzare un elemento `Principal`.

Termini correlati

- Elementi delle policy JSON: principale
- Elementi della politica JSON: NotPrincipal

Errore: carattere jolly della policy di attendibilità del ruolo non supportato nel principale

Codice del problema: ROLE_TRUST_POLICY_UNSUPPORTED_WILDCARD_IN_PRINCIPAL

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Role trust policy unsupported wildcard in principal: "Principal:" "*" is not supported
 in the principal element of a role trust policy. Replace the wildcard with a valid
 principal value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": ""Principal:" "*" is not supported in the principal element of a role
 trust policy. Replace the wildcard with a valid principal value."
```

Risoluzione dell'errore

Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo IAM. Le policy di attendibilità definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono assumere il ruolo. L'elemento `Principal` della policy di attendibilità del ruolo non supporta `"Principal:" "*"`. Sostituisci il jolly con un valore principale valido.

Termini correlati

- Elementi delle policy JSON: principale

Error: errore di sintassi resource della policy di attendibilità del ruolo

Codice del problema: ROLE_TRUST_POLICY_SYNTAX_ERROR_RESOURCE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Role trust policy syntax error resource: Role trust policies apply to the role
 that they are attached to. You cannot specify a resource. Remove the Resource or
 NotResource element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Role trust policies apply to the role that they are attached to. You
 cannot specify a resource. Remove the Resource or NotResource element."
```

Risoluzione dell'errore

Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo IAM. Le policy di attendibilità definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono assumere il ruolo. Le politiche di attendibilità del ruolo si applicano al ruolo a cui sono associate. Non puoi specificare un elemento `Resource` o `NotResource` in una policy di attendibilità del ruolo. Rimozione dell'elemento `Resource` o `NotResource`.

- Elementi delle policy JSON: Resource
- Elementi della politica JSON: NotResource

Errore: il tipo non corrisponde all'intervallo IP

Codice del problema: TYPE_MISMATCH_IP_RANGE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch IP range: The condition operator {{operator}} is used with an invalid IP
 range value. Specify the IP range in standard CIDR format.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition operator {{operator}} is used with an invalid IP range
 value. Specify the IP range in standard CIDR format."
```

Risoluzione dell'errore

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione dell'indirizzo IP, in un formato CIDR.

Termini correlati

- Operatori di condizione indirizzo IP
- Elementi della policy JSON IAM: operatori di condizione

Errore: operazione mancante per la chiave di condizione

Codice del problema: MISSING_ACTION_FOR_CONDITION_KEY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing action for condition key: The {{actionName}} action must be in the action block
 to allow setting values for the condition key {{keyName}}. Add {{actionName}} to the
 action block.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{actionName}} action must be in the action block to allow
 setting values for the condition key {{keyName}}. Add {{actionName}} to the action
 block."
```

Risoluzione dell'errore

La chiave della condizione nell'elemento `Condition` della dichiarazione di policy non viene valutata a meno che l'operazione specificata non sia inclusa nell'elemento `Action`. Per garantire che le chiavi di condizione specificate siano effettivamente consentite o rifiutate dalla policy, aggiungi l'operazione all'elemento `Action`.

Termini correlati

- Elementi delle policy JSON: Action

Errore: sintassi del principale federato non valida nella policy di attendibilità dei ruoli

Codice del problema: INVALID_FEDERATED_PRINCIPAL_SYNTAX_IN_ROLE_TRUST_POLICY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid federated principal syntax in role trust policy: The principal value specifies
 a federated principal that does not match the expected format. Update the federated
 principal to a domain name or a SAML metadata ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The principal value specifies a federated principal that does not
 match the expected format. Update the federated principal to a domain name or a SAML
 metadata ARN."
```

Risoluzione dell'errore

Il valore principale specifica un pricipale federato che non corrisponde al formato previsto. Aggiorna il formato del principale federato con un nome di dominio valido o un ARN di metadati SAML.

Termini correlati

- Utenti federati e ruoli

Errore: operazione non corrispondente per il principale

Codice del problema: MISMATCHED_ACTION_FOR_PRINCIPAL

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Mismatched action for principal: The {{actionName}} action is invalid with the
 following principal(s): {{principalNames}}. Use a SAML provider principal with
```

```
 the sts:AssumeRoleWithSAML action or use an OIDC provider principal with the
 sts:AssumeRoleWithWebIdentity action. Ensure the provider is Federated if you use
 either of the two options.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{actionName}} action is invalid with the following
 principal(s): {{principalNames}}. Use a SAML provider principal with the
 sts:AssumeRoleWithSAML action or use an OIDC provider principal with the
 sts:AssumeRoleWithWebIdentity action. Ensure the provider is Federated if you use
 either of the two options."
```

Risoluzione dell'errore

L'operazione specificata nell'elemento `Action` della dichiarazione di policy non è valida con il principale specificato nell'elemento `Principal`. Ad esempio, non puoi utilizzare un principale provider SAML con l'operazione `sts:AssumeRoleWithWebIdentity`. È necessario utilizzare un provider principale SAML con l'operazione `sts:AssumeRoleWithSAML` oppure utilizzare un principale del fornitore OIDC con l'operazione `sts:AssumeRoleWithWebIdentity`.

Termini correlati

- AssumeRoleWithSAML
- AssumeRoleWithWebIdentity

Errore: operazione mancante per la policy di attendibilità dei ruoli ovunque

Codice del problema: MISSING_ACTION_FOR_ROLES_ANYWHERE_TRUST_POLICY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing action for roles anywhere trust policy: The rolesanywhere.amazonaws.com service
 principal requires the sts:AssumeRole, sts:SetSourceIdentity, and sts:TagSession
 permissions to assume a role. Add the missing permissions to the policy.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The rolesanywhere.amazonaws.com service principal requires the
 sts:AssumeRole, sts:SetSourceIdentity, and sts:TagSession permissions to assume a
 role. Add the missing permissions to the policy."
```

Risoluzione dell'errore

Affinché IAM Roles Anywhere sia in grado di assumere un ruolo e fornire credenziali AWS temporanee, il ruolo deve considerare attendibile il principale del servizio IAM Roles Anywhere. Il principale del servizio IAM Roles Anywhere richiede le autorizzazioni `sts:AssumeRole`, `sts:SetSourceIdentity` e `sts:TagSession` per assumere un ruolo. Se manca una delle autorizzazioni, va aggiunta alla policy.

Termini correlati

- Modello di fiducia in AWS Identity and Access Management Roles Anywhere

Errore: la dimensione della policy supera la quota della RCP

Codice del problema: POLICY_SIZE_EXCEEDS_RCP_QUOTA

Tipo di esito: ERROR

Alla ricerca dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Policy size exceeds RCP quota: The {{policySize}} characters in the resource control
 policy (RCP) exceed the {{policySizeQuota}} character maximum for RCPs. We recommend
 that you use multiple granular policies.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{policySize}} characters in the resource control policy (RCP)
 exceed the {{policySizeQuota}} character maximum for RCPs. We recommend that you use
 multiple granular policies."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo delle risorse (RCP) supportano la specificazione dei valori nell'elemento. `Action` Tuttavia, questi valori possono includere caratteri jolly (*) solo alla fine della stringa. Ciò significa che puoi specificare `s3:Get*` ma non `s3:*Object`.

Per specificare più azioni, AWS consiglia di elencarle singolarmente.

Termini correlati

- Quote per AWS Organizations
- AWS Organizations politiche di controllo delle risorse

Errore: principale dell'errore di sintassi RCP

Codice del problema: RCP_SYNTAX_ERROR_PRINCIPAL

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
RCP syntax error principal: The Principal element contents are not valid. RCPs only
 support specifying all principals ("*") in the Principal element. The NotPrincipal
 element is not supported for RCPs.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Principal element contents are not valid. RCPs only support
 specifying all principals ("*") in the Principal element. The NotPrincipal element is
 not supported for RCPs."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo delle risorse (RCP) supportano solo la specificazione di tutti i principali (» * «) nell'elemento. `Principal` L'elemento `NotPrincipal` non è supportato per le RCP.

Termini correlati

- Sintassi delle RCP

- Proprietà del principale

Errore: autorizzazione con errore di sintassi RCP

Codice del problema: RCP_SYNTAX_ERROR_ALLOW

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
RCP syntax error allow: RCPs only support specifying all principals ("*") in the
 Principal element, all resources ("*") in the Resource element, and no Condition
 element with an effect of Allow.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "RCPs only support specifying all principals ("*") in the Principal
 element, all resources ("*") in the Resource element, and no Condition element with an
 effect of Allow."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo delle risorse (RCP) supportano solo la specificazione di tutti i principali (» * «) nell'`Principal`elemento e di tutte le risorse (» * «) nell'elemento. `Resource` L'elemento `Condition` con un effetto di `Allow` non è supportato per le RCP.

Termini correlati

- Sintassi delle RCP
- Proprietà del principale
- Proprietà della risorsa

Errore: errore di sintassi RCP NotAction

Codice del problema: RCP_SYNTAX_ERROR_NOTACTION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
RCP syntax error NotAction: RCPs do not support the NotAction element. Update to use
 the Action element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "RCPs do not support the NotAction element. Update to use the Action
 element."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo delle risorse (RCP) non supportano l'elemento. `NotAction` Utilizza l'elemento `Action`.

Termini correlati

- Sintassi delle RCP
- Elementi delle policy JSON IAM: Action
- Elementi della policy JSON IAM: NotAction

Errore: errore di sintassi RCP action

Codice del problema: RCP_SYNTAX_ERROR_ACTION

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
RCP syntax error action: RCPs only support specifying select service prefixes in the
 Action element. Learn more here.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "RCPs only support specifying select service prefixes in the Action
 element. Learn more here."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo delle risorse (RCP) supportano solo la specificazione di determinati prefissi di servizio nell'elemento. `Action`

Termini correlati

- Sintassi delle RCP
- Elenco di quelli che supportano gli RCP Servizi AWS

Errore: account ARN mancante

Codice del problema: MISSING_ARN_ACCOUNT

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing ARN account: The resource {{resourceName}} in the arn is missing an account id.
 Please provide a 12 digit account id.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The resource {{resourceName}} in the arn is missing an account id.
 Please provide a 12 digit account id."
```

Risoluzione dell'errore

Include l'ID account nell'ARN della risorsa. Gli ID account sono numeri interi a 12 cifre. Per informazioni su come visualizzare l'ID del tuo account, vedi Trovare l'ID AWS del tuo account.

Termini correlati

- Risorse relative alle policy

- Identificatori account
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: valore della chiave kms non valido

Codice del problema: INVALID_KMS_KEY_VALUE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid kms key value: The {{key}} condition key value must be a valid KMS key ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{key}} condition key value must be a valid KMS key ARN."
```

Risoluzione dell'errore

Un AWS KMS key ARN (Amazon Resource Name) è un identificatore univoco e completo per una chiave KMS. Un ARN chiave include la regione e Account AWS l'ID della chiave. L'ARN di una chiave segue questo formato:

`arn:aws:kms:`*`region`*`:`*`account-id`*`:key/`*`key-id`*

Termini correlati

- Individuazione dell'ID e dell'ARN delle chiavi
- ARN chiave
- AWS chiavi di contesto relative alle condizioni globali

Errore: utilizzo della variabile troppo permissivo

Codice del problema: VARIABLE_USAGE_TOO_PERMISSIVE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Variable usage too permissive: Overly permissive use of policy variable for the {{key}}
 condition key. Use the policy variable preceded by 6 consecutive characters.
```

```
The policy variable is not allowed in the condition key {{key}}. We consider the key to
 be sensitive and policy variables can be evaluated as effective wildcards. Therefore
 policy variables are not allowed to be used with sensitive keys. Refer to https://
docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_condition-keys.html for the
 list of sensitive keys.
```

```
Overly permissive use of policy variable with aws:userID. Use the policy variable on
 the right side of colon or as the only character on the left side of a colon.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Overly permissive use of policy variable for the {{key}} condition
 key. Use the policy variable preceded by 6 consecutive characters."
```

```
"findingDetails": "The policy variable is not allowed in the condition key {{key}}.
 We consider the key to be sensitive and policy variables can be evaluated as
 effective wildcards. Therefore policy variables are not allowed to be used
 with sensitive keys. Refer to https://docs.aws.amazon.com/IAM/latest/UserGuide/
reference_policies_condition-keys.html for the list of sensitive keys."
```

```
"findingDetails": "Overly permissive use of policy variable with aws:userID. Use the
 policy variable on the right side of colon or as the only character on the left side
 of a colon."
```

Risoluzione dell'errore

Esistono tre possibili messaggi di esito relativi a questo errore.

- Per il primo messaggio di errore, modifica l'utilizzo della variabile di policy in modo che sia più specifico. Aggiungi almeno 6 caratteri consecutivi prima della variabile di policy per ridurre

l'ambito delle autorizzazioni. Ad esempio, invece di `${aws:username}`, puoi utilizzare `prefix-${aws:username}` o `myapp-${aws:username}`. Ciò garantisce che la variabile di policy non conceda un accesso eccessivamente ampio.

- Per il secondo messaggio di errore, rimuovi la variabile di policy dalla chiave di condizione specificata. Le variabili di policy possono fungere da jolly efficaci e non ne è consentito l'utilizzo con chiavi di condizione sensibili per motivi di sicurezza. In alternativa, utilizza valori statici specifici o valuta la possibilità di ristrutturare la policy per utilizzare chiavi di condizione non sensibili che supportano le variabili di policy.
- Per il terzo messaggio di errore, modifica l'utilizzo della variabile di policy `aws:userID` in modo che sia più restrittivo. Posiziona la variabile di policy sul lato destro dei due punti (dopo l'ID account) o usala come unico carattere alla sinistra dei due punti. Ad esempio, utilizza `AIDACKCEVSQ6C2EXAMPLE:${aws:userid}` o `${aws:userid}:*` anziché `${aws:userid}`.

Termini correlati

- Elementi della policy IAM: variabili e tag
- Elementi della policy IAM: condizione
- Condizioni con più chiavi o valori di contesto
- AWS chiavi di contesto delle condizioni globali

Errore: utilizzo di carattere jolly troppo permissivo

Codice del problema: WILDCARD_USAGE_TOO_PERMISSIVE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Wildcard usage too permissive: Overly permissive use of wildcard for the {{key}}
 condition key. Use the wildcard preceded by 6 consecutive characters.
```

```
The wildcard is not allowed in the condition key {{key}}. We consider the key to
 be sensitive and wildcards are not allowed to be used with sensitive keys. Refer
 to https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_condition-
keys.html for the list of sensitive keys.
```

```
Overly permissive use of wildcard with aws:userID. Use the wildcard on the right side
 of colon or as the only character on the left side of a colon.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Overly permissive use of wildcard for the {{key}} condition key. Use
 the wildcard preceded by 6 consecutive characters."
```

```
"findingDetails": "The wildcard is not allowed in the condition key {{key}}.
 We consider the key to be sensitive and wildcards are not allowed to be used
 with sensitive keys. Refer to https://docs.aws.amazon.com/IAM/latest/UserGuide/
reference_policies_condition-keys.html for the list of sensitive keys."
```

```
"findingDetails": "Overly permissive use of wildcard with aws:userID. Use the wildcard
 on the right side of colon or as the only character on the left side of a colon."
```

Risoluzione dell'errore

Esistono tre possibili messaggi di esito relativi a questo errore.

- Per il primo messaggio di errore, rendi più specifico l'utilizzo dei caratteri jolly aggiungendo almeno 6 caratteri consecutivi prima del carattere jolly. Ad esempio, invece di `*`, puoi utilizzare `prefix-*` o `prefix-*-suffix`. Ciò riduce l'ambito della condizione e segue il principio del privilegio minimo.
- Per il secondo messaggio di errore, rimuovi il carattere jolly dalla chiave di condizione specificata. I caratteri jolly non sono consentiti con le chiavi di condizione sensibili per motivi di sicurezza. Sostituisci i caratteri jolly con valori specifici che corrispondono allo schema di accesso desiderato oppure valuta la possibilità di utilizzare una chiave di condizione diversa, non sensibile, che supporti i caratteri jolly.
- Per il terzo messaggio di errore, modifica l'utilizzo del carattere jolly `aws:userID` in modo che sia più restrittivo. Posiziona il carattere jolly alla destra dei due punti (dopo l'ID account) o usalo come unico carattere alla sinistra dei due punti. Ad esempio, utilizza `AIDACKCEVSQ6C2EXAMPLE:*` o `*:*` anziché `*`.

Termini correlati

- Elementi della policy IAM: condizione
- Condizioni con più chiavi o valori di contesto
- AWS chiavi di contesto delle condizioni globali
- identificatori IAM

Errore: DynamoDB ha restituito la capacità consumata non valida

Codice di problema: DYNAMODB_INVALID_RETURN_CONSUMED_CAPACITY

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
DynamoDB invalid return consumed capacity: The value for
 dynamodb:ReturnConsumedCapacity is not valid. Update {{value}} to either TOTAL or
 NONE.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The value for dynamodb:ReturnConsumedCapacity is not valid. Update
 {{value}} to either TOTAL or NONE."
```

Risoluzione dell'errore

Aggiorna il valore della condizione con uno dei due valori forniti da DynamoDB: o. `TOTAL` `NONE` I valori fanno distinzione tra maiuscole e minuscole sotto `StringEquals` e`StringLike`, quindi una variante minuscola come non è valida. `total` L'`StringEqualsIgnoreCase`operatore accetta qualsiasi combinazione di maiuscole e minuscole. Una condizione che specifica un valore non valido non corrisponde mai a nessuna richiesta, quindi l'istruzione non ha effetto.

Errore: valori restituiti da DynamoDB non validi

Codice di problema: DYNAMODB_INVALID_RETURN_VALUES

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
DynamoDB invalid return values: The value for dynamodb:ReturnValues is not valid.
 Update {{value}} to one of NONE, ALL_OLD, UPDATED_OLD, ALL_NEW, or UPDATED_NEW.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The value for dynamodb:ReturnValues is not valid. Update {{value}}
 to one of NONE, ALL_OLD, UPDATED_OLD, ALL_NEW, or UPDATED_NEW."
```

Risoluzione dell'errore

Aggiorna il valore della condizione con uno dei cinque valori forniti da DynamoDB:`NONE`,, `ALL_OLD`UPDATED_OLD, `ALL_NEW` o. `UPDATED_NEW` I valori fanno distinzione tra maiuscole e minuscole in `StringEquals` e`StringLike`, quindi una variante minuscola come non è valida. `updated_new` L'`StringEqualsIgnoreCase`operatore accetta qualsiasi combinazione di maiuscole e minuscole. Una condizione che specifica un valore non valido non corrisponde mai a nessuna richiesta, quindi l'istruzione non ha effetto.

Errore: valore di selezione non valido per DynamoDB

Codice di problema: DYNAMODB_INVALID_SELECT_VALUE

Tipo di esito: ERROR

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
DynamoDB invalid select value: The value for dynamodb:Select is not valid. Update
 {{value}} to one of ALL_ATTRIBUTES, ALL_PROJECTED_ATTRIBUTES, SPECIFIC_ATTRIBUTES, or
 COUNT.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The value for dynamodb:Select is not valid. Update {{value}} to one
 of ALL_ATTRIBUTES, ALL_PROJECTED_ATTRIBUTES, SPECIFIC_ATTRIBUTES, or COUNT."
```

Risoluzione dell'errore

Aggiorna il valore della condizione con uno dei quattro valori forniti da DynamoDB:`ALL_ATTRIBUTES`,`ALL_PROJECTED_ATTRIBUTES`, `SPECIFIC_ATTRIBUTES` o. `COUNT` I valori fanno distinzione tra maiuscole e minuscole in `StringEquals` e`StringLike`, quindi una variante minuscola come non è valida. `specific_attributes` L'`StringEqualsIgnoreCase`operatore accetta qualsiasi combinazione di maiuscole e minuscole. Una condizione che specifica un valore non valido non corrisponde mai a nessuna richiesta, quindi l'istruzione non ha effetto.

Avviso generale: crea SLR con NotResource

Codice del problema: CREATE_SLR_WITH_NOT_RESOURCE

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with NotResource: Using the iam:CreateServiceLinkedRole action with
 NotResource can allow creation of unintended service-linked roles for multiple
 resources. We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the iam:CreateServiceLinkedRole action with NotResource can
 allow creation of unintended service-linked roles for multiple resources. We recommend
 that you specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consenta a un AWS servizio di eseguire azioni per conto dell'utente. L'uso di `iam:CreateServiceLinkedRole` in una policy con l'elemento `NotResource` può consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di specificare gli ARN consentiti nell'elemento `Resource`.

- CreateServiceLinkedRole operazione
- Elementi della policy JSON IAM: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso generale: crea una reflex con la stella in azione e NotResource

Codice del problema: CREATE_SLR_WITH_STAR_IN_ACTION_AND_NOT_RESOURCE

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with star in action and NotResource: Using an action with a wildcard(*) and
 NotResource can allow creation of unintended service-linked roles because it can allow
 iam:CreateServiceLinkedRole permissions on multiple resources. We recommend that you
 specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using an action with a wildcard(*) and NotResource can
 allow creation of unintended service-linked roles because it can allow
 iam:CreateServiceLinkedRole permissions on multiple resources. We recommend that you
 specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consenta a un AWS servizio di eseguire azioni per conto dell'utente. Le policy con un carattere jolly (*) nella `Action` che includono l'elemento `NotResource` possono consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di specificare gli ARN consentiti nell'elemento `Resource`.

- CreateServiceLinkedRole operazione
- Elementi della policy JSON IAM: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso generale: crea SLR con e NotAction NotResource

Codice del problema: CREATE_SLR_WITH_NOT_ACTION_AND_NOT_RESOURCE

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with NotAction and NotResource: Using NotAction with NotResource
 can allow creation of unintended service-linked roles because it allows
 iam:CreateServiceLinkedRole permissions on multiple resources. We recommend that you
 specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using NotAction with NotResource can allow creation of unintended
 service-linked roles because it allows iam:CreateServiceLinkedRole permissions on
 multiple resources. We recommend that you specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consenta a un AWS servizio di eseguire azioni per conto dell'utente. L'uso dell'elemento `NotAction` con l'elemento `NotResource` può consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di riscrivere la policy per consentire `iam:CreateServiceLinkedRole` su un elemento limitato di ARN nell'elemento `Resource`. È inoltre possibile aggiungere `iam:CreateServiceLinkedRole` all'elemento `NotAction`.

- CreateServiceLinkedRole operazione
- Elementi della policy JSON IAM: NotAction
- Elementi delle policy JSON IAM: Action
- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso generale: creazione di SLR con stella nella risorsa

Codice del problema: CREATE_SLR_WITH_STAR_IN_RESOURCE

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with star in resource: Using the iam:CreateServiceLinkedRole action with
 wildcards (*) in the resource can allow creation of unintended service-linked roles.
 We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the iam:CreateServiceLinkedRole action with wildcards (*) in
 the resource can allow creation of unintended service-linked roles. We recommend that
 you specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consenta a un AWS servizio di eseguire azioni per conto dell'utente. L'uso di `iam:CreateServiceLinkedRole` in una policy con un carattere jolly (*) l'elemento `Resource` può consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di specificare gli ARN consentiti nell'elemento `Resource`.

- CreateServiceLinkedRole operazione
- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo avviso generale

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Alcuni di questi casi d'uso riguardano utenti esperti all'interno del tuo account. Le seguenti politiche AWS gestite forniscono l'accesso agli utenti esperti e concedono le autorizzazioni per creare ruoli collegati ai servizi per qualsiasi servizio. AWS AWS consiglia di collegare le seguenti policy AWS gestite solo alle identità IAM considerate power user.

- PowerUserAccess
- AlexaForBusinessFullAccess
- AWSOrganizationsServiceTrustPolicy— Questa policy AWS gestita fornisce le autorizzazioni per l'utilizzo da parte del ruolo collegato al AWS Organizations servizio. Questo ruolo consente alle organizzazioni di creare ruoli aggiuntivi collegati ai servizi per altri servizi dell'organizzazione. AWS

Avviso generale: creazione di SLR con stella nell'operazione e nella risorsa

Codice del problema: CREATE_SLR_WITH_STAR_IN_ACTION_AND_RESOURCE

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with star in action and resource: Using wildcards (*) in the action and
 the resource can allow creation of unintended service-linked roles because it allows
 iam:CreateServiceLinkedRole permissions on all resources. We recommend that you
 specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using wildcards (*) in the action and the resource
 can allow creation of unintended service-linked roles because it allows
 iam:CreateServiceLinkedRole permissions on all resources. We recommend that you
 specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consenta a un AWS servizio di eseguire azioni per conto dell'utente. Le policy con un carattere jolly (*) negli elementi `Action` e `Resource` possono consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. Ciò consente di creare un ruolo collegato al servizio quando si specifica`"Action": "*"`,, `"Action": "iam:*"` o. `"Action": "iam:Create*"` AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento.

- CreateServiceLinkedRole operazione
- Elementi delle policy JSON IAM: Action
- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo avviso generale

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Alcuni di questi casi d'uso sono destinati agli amministratori del tuo account. Le seguenti politiche AWS gestite forniscono l'accesso come amministratore e concedono le autorizzazioni per creare ruoli collegati ai servizi per qualsiasi servizio. AWS AWS consiglia di allegare le seguenti politiche AWS gestite solo alle identità IAM considerate amministratori.

- AdministratorAccess
- IAMFullAccess

Avviso generale: crea una SLR con una stella nella risorsa e NotAction

Codice del problema: CREATE_SLR_WITH_STAR_IN_RESOURCE_AND_NOT_ACTION

Tipo di esito: GENERAL_WARNING

Ricerca dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with star in resource and NotAction: Using a resource with wildcards (*)
 and NotAction can allow creation of unintended service-linked roles because it allows
 iam:CreateServiceLinkedRole permissions on all resources. We recommend that you
 specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using a resource with wildcards (*) and NotAction can allow creation
 of unintended service-linked roles because it allows iam:CreateServiceLinkedRole
 permissions on all resources. We recommend that you specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consenta a un AWS servizio di eseguire azioni per conto dell'utente. L'uso dell'elemento `NotAction` in una policy con un carattere jolly (*) nell'elemento `Resource` può consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di specificare gli ARN consentiti nell'elemento `Resource`. È inoltre possibile aggiungere `iam:CreateServiceLinkedRole` all'elemento `NotAction`.

- CreateServiceLinkedRole operazione

- Elementi della policy JSON IAM: NotAction
- Elementi delle policy JSON IAM: Action
- Elementi delle policy JSON IAM: Resource

Avviso generale: chiave di condizione globale obsoleta

Codice del problema: DEPRECATED_GLOBAL_CONDITION_KEY

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Deprecated global condition key: We recommend that you update aws:ARN to use the newer
 condition key aws:PrincipalArn.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "We recommend that you update aws:ARN to use the newer condition key
 aws:PrincipalArn."
```

Risoluzione dell'avviso generale

La policy include una chiave di condizione globale obsoleta. Aggiorna la chiave di condizione nella coppia chiave-valore della condizione per utilizzare una chiave di condizione globale supportata.

- Chiavi della condizione globale

Avviso generale: valore data non valido

Codice del problema: INVALID_DATE_VALUE

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid date value: The date {{date}} might not resolve as expected. We recommend that
 you use the YYYY-MM-DD format.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The date {{date}} might not resolve as expected. We recommend that
 you use the YYYY-MM-DD format."
```

Risoluzione dell'avviso generale

Il tempo Unix Epoch descrive un punto nel tempo trascorso dal 1 gennaio 1970, meno i secondi intercalari. L'epoca potrebbe non corrispondere all'ora esatta prevista. AWS consiglia di utilizzare lo standard W3C per i formati di data e ora. Ad esempio, è possibile specificare una data completa, ad esempio YYYY-MM-DD (1997-07-16), oppure aggiungere l'ora al secondo, ad esempio (1997-07-16 + 01:00). YYYY-MM-DDThh:mm:ssTZD T19:20:30

- Formati di data e ora W3C
- Elementi delle policy JSON IAM: Version
- aws: chiave di condizione globale CurrentTime

Avviso generale: riferimento al ruolo non valido

Codice del problema: INVALID_ROLE_REFERENCE

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid role reference: The Principal element includes the IAM role ID {{roleid}}. We
 recommend that you use a role ARN instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Principal element includes the IAM role ID {{roleid}}. We
 recommend that you use a role ARN instead."
```

Risoluzione dell'avviso generale

AWS consiglia di specificare l'Amazon Resource Name (ARN) per un ruolo IAM anziché il relativo ID principale. Quando IAM salva la policy, trasformerà l'ARN nell'ID principale per il ruolo esistente. AWS include una precauzione di sicurezza. Se qualcuno elimina e crea nuovamente il ruolo, avrà un nuovo ID e la policy non corrisponderà all'ID del nuovo ruolo.

- Specifica di un principale: ruoli IAM
- ARN IAM
- ID univoci IAM

Avviso generale: riferimento utente non valido

Codice del problema: INVALID_USER_REFERENCE

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Invalid user reference: The Principal element includes the IAM user ID {{userid}}. We
 recommend that you use a user ARN instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Principal element includes the IAM user ID {{userid}}. We
 recommend that you use a user ARN instead."
```

Risoluzione dell'avviso generale

AWS consiglia di specificare l'Amazon Resource Name (ARN) per un utente IAM anziché il suo ID principale. Quando IAM salva la policy, trasformerà l'ARN nell'ID principale per l'utente esistente. AWS include una precauzione di sicurezza. Se qualcuno elimina e crea nuovamente l'utente, avrà un nuovo ID e la policy non corrisponderà all'ID del nuovo utente.

- Specifica di un principale: utenti IAM
- ARN IAM

- ID univoci IAM

Avviso generale: versione mancante

Codice del problema: MISSING_VERSION

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing version: We recommend that you specify the Version element to help you with
 debugging permission issues.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "We recommend that you specify the Version element to help you with
 debugging permission issues."
```

Risoluzione dell'avviso generale

AWS consiglia di includere il `Version` parametro opzionale nella policy. Se non includi un elemento Versione, per impostazione predefinita il valore viene impostato su `2012-10-17`, ma le funzionalità più recenti, come le variabili di policy, non funzioneranno con la policy. Ad esempio, le variabili tipo `${aws:username}` non saranno riconosciute come variabili e verranno trattate come stringhe letterali nella policy.

- Elementi delle policy JSON IAM: Version

Avviso generale: sid univoci consigliati

Codice del problema: UNIQUE_SIDS_RECOMMENDED

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Unique Sids recommended: We recommend that you use statement IDs that are unique to
 your policy. Update the Sid value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "We recommend that you use statement IDs that are unique to your
 policy. Update the Sid value."
```

Risoluzione dell'avviso generale

AWS consiglia di utilizzare ID di dichiarazione univoci. L'elemento `Sid` (ID istruzione) consente di immettere un identificatore opzionale fornito per l'istruzione della policy. Puoi assegnare un valore ID di istruzione a ogni istruzione in una matrice di istruzioni utilizzando l'elemento `SID`.

Termini correlati

- Elementi delle policy JSON IAM: Sid

Avviso generale: carattere jolly senza operatore like

Codice del problema: WILDCARD_WITHOUT_LIKE_OPERATOR

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Wildcard without like operator: Your condition value includes a * or ? character. If
 you meant to use a wildcard (*, ?), update the condition operator to include Like.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Your condition value includes a * or ? character. If you meant to
 use a wildcard (*, ?), update the condition operator to include Like."
```

Risoluzione dell'avviso generale

La struttura dell'elemento `Condition` richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore. Quando specifichi un valore di condizione che utilizza un carattere jolly (*,?) , devi utilizzare la versione `Like`dell'operatore di condizione. Ad esempio, anziché l'operatore di condizione stringa `StringEquals`, utilizza `StringLike`.

```
"Condition": {"StringLike": {"aws:PrincipalTag/job-category": "admin-*"}}
```

- Elementi della policy JSON IAM: operatori di condizione
- Elementi della policy JSON IAM: Condition

AWS politiche gestite con questo avviso generale

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Le seguenti politiche AWS gestite includono caratteri jolly nel loro valore di condizione senza un operatore di condizione che `Like` includa il pattern matching. Quando si utilizza la policy AWS gestita come riferimento per creare una policy gestita dal cliente, si AWS consiglia di utilizzare un operatore di condizione che supporti il pattern matching con caratteri jolly (*,?) , ad esempio. `StringLike`

- AWSGlueConsoleSageMakerNotebookFullAccess

Avviso generale: la dimensione della policy supera la quota delle policy di identità

Codice del problema: POLICY_SIZE_EXCEEDS_IDENTITY_POLICY_QUOTA

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Policy size exceeds identity policy quota: The {{policySize}} characters in the
 identity policy, excluding whitespace, exceed the {{policySizeQuota}} character
 maximum for inline and managed policies. We recommend that you use multiple granular
 policies.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{policySize}} characters in the identity policy, excluding
 whitespace, exceed the {{policySizeQuota}} character maximum for inline and managed
 policies. We recommend that you use multiple granular policies."
```

Risoluzione dell'avviso generale

Puoi allegare fino a 20 policy gestite a un'identità IAM (utente, gruppo di utenti o ruolo). Tuttavia, le dimensioni di ciascuna policy gestita non possono superare la quota di default i 6.144 caratteri. IAM non calcola gli spazi vuoti per determinare le dimensioni di una policy rispetto a tali limiti. Le quote, note anche come limiti in AWS, sono i valori massimi per le risorse, le azioni e gli elementi del tuo AWS account.

Inoltre, puoi aggiungere a un'identità IAM tutte le policy in linea desiderate. Tuttavia, la dimensione della somma di tutte le policy in linea per identità non può superare la quota specificata.

Se la policy è maggiore della quota, è possibile organizzare la policy in più istruzioni e raggruppare le istruzioni in più policy.

Termini correlati

- IAM e quote di AWS STS caratteri
- Istruzioni e policy multiple
- Policy gestite dal cliente IAM
- Panoramica delle policy JSON
- Sintassi della policy JSON IAM

AWS politiche gestite con questo avviso generale

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Le seguenti politiche AWS gestite concedono le autorizzazioni alle azioni su molti AWS servizi e superano la dimensione massima delle politiche. Quando si utilizza la policy gestita da AWS come riferimento per creare la policy gestita, è necessario suddividerla in più policy.

- ReadOnlyAccess
- AWSSupportServiceRolePolicy

Avviso generale: la dimensione della policy supera la quota delle policy delle risorse

Codice del problema: POLICY_SIZE_EXCEEDS_RESOURCE_POLICY_QUOTA

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Policy size exceeds resource policy quota: The {{policySize}} characters in the
 resource policy exceed the {{policySizeQuota}} character maximum for resource
 policies. We recommend that you use multiple granular policies.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{policySize}} characters in the resource policy exceed the
 {{policySizeQuota}} character maximum for resource policies. We recommend that you use
 multiple granular policies."
```

Risoluzione dell'avviso generale

Resource-based le policy sono documenti di policy JSON che alleghi a una risorsa, ad esempio un bucket Amazon S3. Queste policy concedono all'entità principale specificata l'autorizzazione per eseguire operazioni specifiche sulla risorsa e definiscono le condizioni in cui ciò si applica. La dimensione delle policy basate sulle risorse non può superare la quota impostata per quella risorsa. Le quote, note anche come limiti in AWS, sono i valori massimi per le risorse, le azioni e gli elementi del tuo account. AWS

Se la policy è maggiore della quota, è possibile organizzare la policy in più istruzioni e raggruppare le istruzioni in più policy.

Termini correlati

- Resource-based politiche
- policy del bucket di Amazon S3
- Istruzioni e policy multiple
- Panoramica delle policy JSON
- Sintassi della policy JSON IAM

Avviso generale: mancata corrispondenza del tipo

Codice del problema: TYPE_MISMATCH

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch: Use the operator type {{allowed}} instead of operator {{operator}} for
 the condition key {{key}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Use the operator type {{allowed}} instead of operator {{operator}}
 for the condition key {{key}}."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione supportato.

Ad esempio, la chiave di condizione globale di `aws:MultiFactorAuthPresent` richiede un operatore di condizione con il tipo di dati `Boolean`. Se specifichi una data o un numero intero, il tipo di dati non corrisponderà.

Termini correlati

- Chiavi della condizione globale
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: booleano con mancata corrispondenza del tipo

Codice del problema: TYPE_MISMATCH_BOOLEAN

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch Boolean: Add a valid Boolean value (true or false) for the condition
 operator {{operator}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a valid Boolean value (true or false) for the condition operator
 {{operator}}."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare un tipo di dati dell'operatore condizione booleano, ad esempio `true` o `false`.

Ad esempio, la chiave di condizione globale di `aws:MultiFactorAuthPresent` richiede un operatore di condizione con il tipo di dati `Boolean`. Se specifichi una data o un numero intero, il tipo di dati non corrisponderà.

Termini correlati

- Operatori di condizione booleani
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: data della mancata corrispondenza del tipo

Codice del problema: TYPE_MISMATCH_DATE

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch date: The date condition operator is used with an invalid value. Specify
 a valid date using YYYY-MM-DD or other ISO 8601 date/time format.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The date condition operator is used with an invalid value. Specify a
 valid date using YYYY-MM-DD or other ISO 8601 date/time format."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione data, in un formato data ora `YYYY-MM-DD` o altro ISO 8601.

Termini correlati

- Operatori di condizione data
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: numero della mancata corrispondenza del tipo

Codice del problema: TYPE_MISMATCH_NUMBER

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch number: Add a valid numeric value for the condition operator
 {{operator}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a valid numeric value for the condition operator {{operator}}."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione numerico.

Termini correlati

- Operatori di condizione numerici
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: stringa della mancata corrispondenza del tipo

Codice del problema: TYPE_MISMATCH_STRING

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch string: Add a valid base64-encoded string value for the condition
 operator {{operator}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a valid base64-encoded string value for the condition operator
 {{operator}}."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione stringa.

Termini correlati

- Operatori di condizione stringa
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: si consigliano repository e ramo github specifici

Codice del problema: SPECIFIC_GITHUB_REPO_AND_BRANCH_RECOMMENDED

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Specific github repo and branch recommended: Using a wildcard (*) in
 token.actions.githubusercontent.com:sub can allow requests from more sources than
 you intended. Specify the value of token.actions.githubusercontent.com:sub with the
```

```
repository and branch name. If the subject claim does not include the branch, specify
it in a token.actions.githubusercontent.com:ref condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using a wildcard (*) in token.actions.githubusercontent.com:sub
 can allow requests from more sources than you intended. Specify the value
 of token.actions.githubusercontent.com:sub with the repository and branch
 name. If the subject claim does not include the branch, specify it in a
 token.actions.githubusercontent.com:ref condition key."
```

Risoluzione dell'avviso generale

Se si utilizza GitHub come IdP OIDC, la procedura migliore è limitare le entità che possono assumere il ruolo associato all'IdP IAM. Quando includi una `Condition` dichiarazione in una politica di fiducia dei ruoli, puoi limitare il ruolo a un' GitHub organizzazione, un repository o una filiale specifici. Puoi utilizzare la chiave della condizione `token.actions.githubusercontent.com:sub` per limitare l'accesso. Ti consigliamo di limitare la condizione a un insieme specifico di repository o rami. Se utilizzi una wildcard (`*`) in`token.actions.githubusercontent.com:sub`, GitHub le azioni provenienti da organizzazioni o repository al di fuori del tuo controllo possono assumere ruoli associati all' GitHub IdP IAM nel tuo account. AWS

Specifica il repository in. `token.actions.githubusercontent.com:sub` È possibile specificare il ramo con lo stesso valore o in una chiave di `token.actions.githubusercontent.com:ref` condizione separata. Se l'affermazione oggetto non include il ramo, ad esempio quando nomina un GitHub ambiente, specifica il ramo in`token.actions.githubusercontent.com:ref`.

Termini correlati

- Configurazione di un ruolo per il provider di GitHub identità OIDC

Avviso generale: la dimensione della policy supera la quota delle policy di attendibilità del ruolo

Codice del problema: POLICY_SIZE_EXCEEDS_ROLE_TRUST_POLICY_QUOTA

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Policy size exceeds role trust policy quota: The characters in the role trust policy,
 excluding whitespace, exceed the character maximum. We recommend that you request a
 role trust policy length quota increase using Service Quotas and AWS Support Center.
 If the quotas have already been increased, then you can ignore this warning.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The characters in the role trust policy, excluding whitespace,
 exceed the character maximum. We recommend that you request a role trust policy length
 quota increase using Service Quotas and AWS Support Center. If the quotas have already
 been increased, then you can ignore this warning."
```

Risoluzione dell'avviso generale

IAM e AWS STS dispongono di quote che limitano la dimensione delle politiche di fiducia dei ruoli. I caratteri nella policy di attendibilità del ruolo, esclusi gli spazi bianchi, superano il numero massimo di caratteri. È consigliabile richiedere un aumento della quota della policy di attendibilità del ruolo utilizzando Service Quotas o AWS Support Center Console.

Termini correlati

- IAM e AWS STS quote, requisiti relativi ai nomi e limiti di caratteri

Avviso generale: a RCP manca la chiave della condizione principale correlata

Codice del problema: RCP_MISSING_RELATED_PRINCIPAL_CONDITION_KEY

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
RCP missing related principal condition key: RCPs impact IAM roles, users, and AWS
 service principals. To prevent unintended impact to services acting on your behalf
 using a service principal, an additional statement should be added to the Condition
 block "BoolIfExists": { "aws:PrincipalIsAWSService": "false"} whenever a principal key
 {{conditionKeyName}} is used.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "RCPs impact IAM roles, users, and AWS service principals. To prevent
 unintended impact to services acting on your behalf using a service principal,
 an additional statement should be added to the Condition block "BoolIfExists":
 { "aws:PrincipalIsAWSService": "false"} whenever a principal key {{conditionKeyName}}
 is used."
```

Risoluzione dell'avviso generale

AWS Organizations le politiche di controllo delle risorse (RCP) possono influire sui ruoli, sugli utenti e sui responsabili IAM. Servizio AWS Per evitare un impatto involontario sui servizi che agiscono per tuo conto utilizzando un principale di servizio, aggiungi la seguente istruzione al tuo elemento `Condition`.

```
"BoolIfExists": { "aws:PrincipalIsAWSService": "false"}
```

Termini correlati

- Sintassi delle RCP
- Proprietà del principale

Avviso generale: a RCP manca la chiave di condizione del principale del servizio correlata

Codice del problema: RCP_MISSING_RELATED_SERVICE_PRINCIPAL_CONDITION_KEY

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
RCP missing related service principal condition key: RCPs impact IAM roles, users,
 and AWS service principals. To prevent unintended impact to your principals,
 an additional statement should be added to the Condition block "BoolIfExists":
 { "aws:PrincipalIsAWSService": "true"} whenever the key {{conditionKeyName}} is used.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "RCPs impact IAM roles, users, and AWS service principals. To prevent
 unintended impact to your principals, an additional statement should be added to the
 Condition block "BoolIfExists": { "aws:PrincipalIsAWSService": "true"} whenever the
 key {{conditionKeyName}} is used."
```

Risoluzione dell'avviso generale

AWS Organizations le politiche di controllo delle risorse (RCP) possono influire sui ruoli, sugli utenti e sui responsabili IAM. Servizio AWS Per evitare un impatto involontario sui principali, aggiungi la seguente istruzione al tuo elemento `Condition`:

```
"BoolIfExists": { "aws:PrincipalIsAWSService": "true"}
```

Termini correlati

- Sintassi delle RCP
- Proprietà del principale

Avviso generale: a RCP manca il controllo null della chiave di condizione del servizio

Codice del problema: RCP_MISSING_SERVICE_CONDITION_KEY_NULL_CHECK

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
RCP missing service condition key null check: The specified service may have a service
 integration that does not require the use of the {{conditionKeyName}} condition
 key. To prevent unintended impact to services acting on your behalf using a service
 principal, an additional statement should be added to the Condition block "Null":
 { "aws:SourceAccount": "false"} or "Null": { "aws:SourceArn": "false"} whenever the
 key {{conditionKeyName}} is used.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The specified service may have a service integration that does
 not require the use of the {{conditionKeyName}} condition key. To prevent unintended
 impact to services acting on your behalf using a service principal, an additional
 statement should be added to the Condition block "Null": { "aws:SourceAccount":
 "false"} or "Null": { "aws:SourceArn": "false"} whenever the key {{conditionKeyName}}
 is used."
```

Risoluzione dell'avviso generale

AWS Organizations le politiche di controllo delle risorse (RCP) possono influire sui ruoli, sugli utenti e sui responsabili IAM. Servizio AWS Per evitare un impatto involontario sui servizi che agiscono per tuo conto utilizzando un principale di servizio, aggiungi le seguenti istruzioni al tuo elemento `Condition` ogni volta che viene utilizzata la chiave specificata:

```
"Null": { "aws:SourceAccount": "false"}
```

or

```
"Null": { "aws:SourceArn": "false"}
```

Termini correlati

- Sintassi delle RCP
- Proprietà del principale

Avviso generale: utilizza la chiave di condizione solo con i servizi supportati

Codice del problema: USE_CONDITION_KEY_ONLY_WITH_SUPPORTED_SERVICES

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Use condition key only with supported services: The condition key {{key}} works only
 with specific AWS services and must be scoped to supported services in your policies.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{key}} works only with specific AWS services and
 must be scoped to supported services in your policies.
```

Risoluzione dell'avviso generale

AWS Consulta la documentazione per identificare quali Servizi AWS supportano questa chiave di condizione. Se alcuni servizi della tua policy non supportano la chiave di condizione, modifica la tua politica in modo che la chiave di condizione includa solo i servizi Servizi AWS che la supportano.

Termini correlati

- `aws:VpceAccount`
- `aws:VpceOrgID`
- `aws:VpceOrgPaths`
- Azioni, risorse e chiavi di condizione per Servizi AWS

Avviso generale: operazione di sostituzione con attributi in DynamoDB

Codice di problema: DYNAMODB_REPLACE_OPERATION_WITH_ATTRIBUTES

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
DynamoDB replace operation with attributes: The {{action}} action replaces the entire
 item, so dynamodb:Attributes cannot restrict access at the attribute level. That
 condition only applies to UpdateItem; use dynamodb:UpdateItem for attribute-level
 access control.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{action}} action replaces the entire item, so
 dynamodb:Attributes cannot restrict access at the attribute level. That condition only
 applies to UpdateItem; use dynamodb:UpdateItem for attribute-level access control."
```

Risoluzione dell'avviso generale

La `dynamodb:Attributes` condizione non limita `dynamodb:PutItem` o`dynamodb:BatchWriteItem`. Queste azioni sostituiscono l'intero elemento, quindi un chiamante può eliminare un attributo limitato semplicemente escludendolo dall'elemento che scrive. Per controllare quali attributi un chiamante può modificare, concedi `dynamodb:UpdateItem` invece e mantieni la `dynamodb:Attributes` condizione contenuta in tale dichiarazione. Se non intendevi controllare a livello di attributi in questo caso, rimuovi la condizione. `dynamodb:Attributes`

Avviso generale: la scansione DynamoDB ignora i tasti principali

Codice di problema: DYNAMODB_SCAN_BYPASSES_LEADING_KEYS

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
DynamoDB scan bypasses leading keys: The dynamodb:Scan operation returns all items
 in the table, bypassing the dynamodb:LeadingKeys restriction. Consider removing
 dynamodb:Scan.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The dynamodb:Scan operation returns all items in the table,
 bypassing the dynamodb:LeadingKeys restriction. Consider removing dynamodb:Scan."
```

Risoluzione dell'avviso generale

`dynamodb:Scan`legge l'intera tabella, quindi la `dynamodb:LeadingKeys` condizione in questa istruzione non limita gli elementi che un chiamante può leggere. `dynamodb:Scan`Rimuovilo o concedilo in una dichiarazione separata su cui non si basa`dynamodb:LeadingKeys`.

Avviso generale: condizione di intervento errata di DynamoDB

Codice di problema: DYNAMODB_WRONG_CONDITION_FOR_ACTION

Tipo di esito: GENERAL_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
DynamoDB wrong condition for action: The condition key {{conditionKey}} does
 not apply to the {{action}} action. Use dynamodb:Select with read actions and
 dynamodb:ReturnValues with write actions.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{conditionKey}} does not apply to the {{action}}
 action. Use dynamodb:Select with read actions and dynamodb:ReturnValues with write
 actions."
```

Risoluzione dell'avviso generale

Questa istruzione viene utilizzata `dynamodb:Select` con un'azione di scrittura o `dynamodb:ReturnValues` con un'azione di lettura. DynamoDB fornisce `dynamodb:Select` solo azioni di lettura e `dynamodb:ReturnValues` solo azioni di scrittura, quindi la condizione qui non ha mai effetto e non protegge nulla. Associa ogni chiave a un'azione di supporto o rimuovila. Questo controllo si applica sia alle dichiarazioni che `Allow` alle `Deny` dichiarazioni.

Avviso di sicurezza: chiave di condizione non attendibile

Codice del problema: UNTRUSTWORTHY_CONDITION_KEY

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Untrustworthy condition key: The {{key}} condition key is not recommended for access
 control as it can be spoofed/manipulated by the caller.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{key}} condition key is not recommended for access control as
 it can be spoofed/manipulated by the caller."
```

Risoluzione dell'avviso di sicurezza

Non utilizzare questa chiave di condizione per il controllo degli accessi. Il chiamante può potenzialmente manipolare o falsificare il valore della chiave, determinando un rischio per la sicurezza.

Termini correlati

- AWS condizioni globali, chiavi di contesto

Avviso di sicurezza: consenti con NotPrincipal

Codice del problema: ALLOW_WITH_NOT_PRINCIPAL

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Allow with NotPrincipal: Using Allow with NotPrincipal can be overly permissive. We
 recommend that you use Principal instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using Allow with NotPrincipal can be overly permissive. We recommend
 that you use Principal instead."
```

Risoluzione dell'avviso di sicurezza

L'uso di `"Effect": "Allow"` con `NotPrincipal` può essere eccessivamente permissivo. Ad esempio, questo può concedere autorizzazioni a committenti anonimi. AWS consiglia di specificare i principali a cui è necessario accedere utilizzando l'elemento. `Principal` In alternativa, è possibile consentire un accesso ampio e quindi aggiungere un'altra istruzione che utilizza l'elemento `NotPrincipal` con `“Effect”: “Deny”`.

- AWS Elementi delle policy JSON: principale
- AWS Elementi della politica JSON: NotPrincipal

Avviso di sicurezza: ForAllValues con chiave a valore singolo

Codice del problema: FORALLVALUES_WITH_SINGLE_VALUED_KEY

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
ForAllValues with single valued key: Using ForAllValues qualifier with the single-
valued condition key {{key}} can be overly permissive. We recommend that you remove
 ForAllValues:.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using ForAllValues qualifier with the single-valued condition key
 {{key}} can be overly permissive. We recommend that you remove ForAllValues:."
```

Risoluzione dell'avviso di sicurezza

AWS consiglia di utilizzare l'`ForAllValues`unico con condizioni multivalore. L'operatore impostato `ForAllValues` verifica se il valore di ogni membro del set di richieste è un sottoinsieme del set di chiavi di condizione. La condizione restituisce true se ogni valore delle chiavi nella richiesta corrisponde ad almeno un valore nella policy. Restituisce true anche se non ci sono chiavi nella richiesta o se i valori delle chiavi si riducono a un set di dati nullo, ad esempio una stringa vuota.

Per sapere se una condizione supporta un valore singolo o più valori, rivedi la pagina Operazioni, risorse e chiavi di condizione per il servizio. Le chiavi di condizione con il prefisso del tipo di dati `Array0f` sono chiavi di condizione multivalore. Ad esempio, Amazon SES supporta chiavi con valori singoli (`String`) e il tipo di dati multivalore `Array0fString`.

- Chiavi di contesto multivalore

Avviso di sicurezza: passa il ruolo con NotResource

Codice del problema: PASS_ROLE_WITH_NOT_RESOURCE

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Pass role with NotResource: Using the iam:PassRole action with NotResource can be
 overly permissive because it can allow iam:PassRole permissions on multiple resources.
 We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the iam:PassRole action with NotResource can be overly
 permissive because it can allow iam:PassRole permissions on multiple resources. We
 recommend that you specify resource ARNs instead."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). L'utilizzo `iam:PassRole` di una policy con l'`NotResource`elemento può consentire ai responsabili di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- sono: PassedToService
- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso di sicurezza: passa il ruolo con stella in azione e NotResource

Codice del problema: PASS_ROLE_WITH_STAR_IN_ACTION_AND_NOT_RESOURCE

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Pass role with star in action and NotResource: Using an action with a wildcard (*) and
 NotResource can be overly permissive because it can allow iam:PassRole permissions on
 multiple resources. We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using an action with a wildcard (*) and NotResource can be overly
 permissive because it can allow iam:PassRole permissions on multiple resources. We
 recommend that you specify resource ARNs instead."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). Le policy con un carattere jolly (*) nel campo `Action` e che includono l'`NotResource`elemento possono consentire ai principali di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- sono: PassedToService
- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso di sicurezza: passaggio del ruolo con e NotAction NotResource

Codice del problema: PASS_ROLE_WITH_NOT_ACTION_AND_NOT_RESOURCE

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Pass role with NotAction and NotResource: Using NotAction with NotResource can
 be overly permissive because it can allow iam:PassRole permissions on multiple
 resources.. We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using NotAction with NotResource can be overly permissive because
 it can allow iam:PassRole permissions on multiple resources.. We recommend that you
 specify resource ARNs instead."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). L'uso dell'`NotAction`elemento e l'elenco di alcune risorse nell'`NotResource`elemento possono consentire ai responsabili di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- sono: PassedToService
- Elementi della policy IAM JSON: NotAction
- Elementi delle policy JSON IAM: Action
- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso di sicurezza: invio del ruolo con stella in risorsa

Codice del problema: PASS_ROLE_WITH_STAR_IN_RESOURCE

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Pass role with star in resource: Using the iam:PassRole action with wildcards (*)
 in the resource can be overly permissive because it allows iam:PassRole permissions
 on multiple resources. We recommend that you specify resource ARNs or add the
 iam:PassedToService condition key to your statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the iam:PassRole action with wildcards (*) in the resource can
 be overly permissive because it allows iam:PassRole permissions on multiple resources.
 We recommend that you specify resource ARNs or add the iam:PassedToService condition
 key to your statement."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). Le policy che lo consentono `iam:PassRole` e che includono un carattere jolly (*) nell'`Resource`elemento possono consentire ai principali di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

Alcuni AWS servizi includono il rispettivo namespace nel nome del ruolo. Questo controllo delle policy tiene conto di queste convenzioni durante l'analisi della policy per generare i risultati. Ad esempio, il seguente ARN della risorsa potrebbe non generare un risultato:

```
arn:aws:iam::*:role/Service*
```

- Invio di un ruolo a un servizio
- Sono: PassedToService
- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo avviso di sicurezza

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Uno di questi casi d'uso riguarda gli amministratori all'interno del tuo account. Le seguenti politiche AWS gestite forniscono l'accesso come amministratore e concedono le autorizzazioni per passare qualsiasi ruolo IAM a qualsiasi servizio. AWS consiglia di allegare le seguenti politiche AWS gestite solo alle identità IAM considerate amministratori.

- AdministratorAccess-Amplify

Le seguenti politiche AWS gestite includono le autorizzazioni a utilizzare un `iam:PassRole` carattere jolly (*) nella risorsa e si trovano su un percorso di deprecazione. ??? Per ognuna di queste politiche, abbiamo aggiornato le linee guida sulle autorizzazioni, ad esempio consigliando una nuova politica AWS gestita che supporti il caso d'uso. Per visualizzare le alternative a queste policy, consulta per guide relative a ogni servizio.

- AWSElasticBeanstalkFullAccess
- AWSElasticBeanstalkService
- AWSLambdaFullAccess
- AWSLambdaReadOnlyAccess
- AWSOpsWorksFullAccess
- AWSOpsWorksRole
- AWSDataPipelineRole
- AmazonDynamoDBFullAccesswithDataPipeline
- AmazonElasticMapReduceFullAccess
- AmazonDynamoDBFullAccesswithDataPipeline
- AmazonEC2ContainerServiceFullAccess

Le seguenti politiche AWS gestite forniscono autorizzazioni solo per i ruoli collegati ai servizi, che consentono ai AWS servizi di eseguire azioni per conto dell'utente. Non puoi collegare queste policy alle identità IAM.

- AWSServiceRoleForAmazonEKSNodegroup

Avviso di sicurezza: invio del ruolo con stella in azione e risorsa

Codice del problema: PASS_ROLE_WITH_STAR_IN_ACTION_AND_RESOURCE

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Pass role with star in action and resource: Using wildcards (*) in the action and the
 resource can be overly permissive because it allows iam:PassRole permissions on all
```

```
resources. We recommend that you specify resource ARNs or add the iam:PassedToService
condition key to your statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using wildcards (*) in the action and the resource can be overly
 permissive because it allows iam:PassRole permissions on all resources. We recommend
 that you specify resource ARNs or add the iam:PassedToService condition key to your
 statement."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). Le policy con un carattere jolly (*) negli `Resource` elementi `Action` e possono consentire ai principali di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- sono: PassedToService
- Elementi delle policy JSON IAM: Action
- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo avviso di sicurezza

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Alcuni di questi casi d'uso sono destinati agli amministratori del tuo account. Le seguenti politiche AWS gestite forniscono l'accesso come amministratore e concedono le autorizzazioni per passare qualsiasi ruolo IAM a qualsiasi servizio. AWS AWS consiglia di allegare le seguenti politiche AWS gestite solo alle identità IAM considerate amministratori.

- AdministratorAccess
- IAMFullAccess

Avviso di sicurezza: assegna il ruolo con una stella nella risorsa e NotAction

Codice del problema: PASS_ROLE_WITH_STAR_IN_RESOURCE_AND_NOT_ACTION

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Pass role with star in resource and NotAction: Using a resource with wildcards (*) and
 NotAction can be overly permissive because it allows iam:PassRole permissions on all
 resources. We recommend that you specify resource ARNs or add the iam:PassedToService
 condition key to your statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using a resource with wildcards (*) and NotAction can be overly
 permissive because it allows iam:PassRole permissions on all resources. We recommend
 that you specify resource ARNs or add the iam:PassedToService condition key to your
 statement."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). L'utilizzo dell'`NotAction`elemento in una policy con un carattere jolly (*) nell'`Resource`elemento può consentire ai principali di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- sono: PassedToService
- Elementi della policy IAM JSON: NotAction
- Elementi delle policy JSON IAM: Action
- Elementi delle policy JSON IAM: Resource

Avviso di sicurezza: chiavi di condizione abbinate mancanti

Codice del problema: MISSING_PAIRED_CONDITION_KEYS

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing paired condition keys: Using the condition key {{conditionKeyName}}
 can be overly permissive without also using the following condition keys:
 {{recommendedKeys}}. Condition keys like this one are more secure when paired with
 a related key. We recommend that you add the related condition keys to the same
 condition block.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the condition key {{conditionKeyName}} can be overly
 permissive without also using the following condition keys: {{recommendedKeys}}.
 Condition keys like this one are more secure when paired with a related key. We
 recommend that you add the related condition keys to the same condition block."
```

Risoluzione dell'avviso di sicurezza

Alcune chiavi di condizione sono più sicure se abbinate ad altre chiavi di condizione correlate. AWS consiglia di includere le chiavi di condizione correlate nello stesso blocco di condizione della chiave di condizione esistente. Ciò rende più sicure le autorizzazioni concesse tramite la policy.

Ad esempio, è possibile utilizzare la chiave di condizione `aws:VpcSourceIp` per confrontare l'indirizzo IP da cui è stata effettuata una richiesta con l'indirizzo IP specificato nella policy. AWS consiglia di aggiungere la chiave di condizione `aws:SourceVPC` correlata. Controlla se la richiesta proviene dal VPC specificato nella policy e l'indirizzo IP specificato.

Termini correlati

- Chiave della condizione globale di `aws:VpcSourceIp`
- Chiave della condizione globale di `aws:SourceVPC`
- Chiavi della condizione globale
- Elemento condizione

- Panoramica delle policy JSON

Avviso di sicurezza: Rifiuta con chiave di condizione tag non supportata per il servizio

Codice del problema: DENY_WITH_UNSUPPORTED_TAG_CONDITION_KEY_FOR_SERVICE

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Deny with unsupported tag condition key for service: Using the effect Deny with the
 tag condition key {{conditionKeyName}} and actions for services with the following
 prefixes can be overly permissive: {{serviceNames}}. Actions for the listed services
 are not denied by this statement. We recommend that you move these actions to a
 different statement without this condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the effect Deny with the tag condition key
 {{conditionKeyName}} and actions for services with the following prefixes can be
 overly permissive: {{serviceNames}}. Actions for the listed services are not denied
 by this statement. We recommend that you move these actions to a different statement
 without this condition key."
```

Risoluzione dell'avviso di sicurezza

L'utilizzo di chiavi di condizione dei tag non supportate nell'`Condition`elemento di una policy con `"Effect": "Deny"` può essere eccessivamente permissivo, poiché la condizione viene ignorata per quel servizio. AWS consiglia di rimuovere le azioni del servizio che non supportano la chiave di condizione e di creare un'altra dichiarazione per negare l'accesso a risorse specifiche per tali azioni.

Se utilizzi la chiave di condizione `aws:ResourceTag` e non è supportata da un'operazione di servizio, la chiave non viene inclusa nel contesto della richiesta. In questo caso, la condizione nell'istruzione `Deny` restituisce sempre `false` e l'operazione non viene mai negata. Ciò accade anche se la risorsa è taggata correttamente.

Quando un servizio supporta la chiave di condizione `aws:ResourceTag`, è possibile utilizzare i tag per controllare l'accesso alle risorse del servizio. Questo è noto come controllo degli accessi basato

su attributi (ABAC). I servizi che non supportano queste chiavi richiedono il controllo dell'accesso alle risorse tramite il controllo degli accessi basato su risorse (RBAC).

> **Note**
>
> Alcuni servizi consentono il supporto per la chiave di condizione `aws:ResourceTag` per un sottoinsieme di risorse e operazioni. Sistema di analisi degli accessi IAM restituisce risultati per le operazioni di servizio non supportate. Ad esempio, Amazon S3 supporta `aws:ResourceTag` per un sottoinsieme delle relative risorse. Per visualizzare tutti i tipi di risorse disponibili in Amazon S3 che supportano la chiave di condizione `aws:ResourceTag`, consulta Tipi di risorse definiti da Amazon S3 in Service Authorization Reference.

Ad esempio, supponiamo che tu desideri rifiutare l'accesso per rimuovere tag da risorse specifiche che sono taggate con la coppia chiave-valore `status=Confidential`. Supponiamo inoltre che AWS Lambda consenta di etichettare e rimuovere i tag `aws:ResourceTag` dalle risorse, ma non supporti la chiave condizionale. Per negare le azioni di eliminazione per AWS App Mesh e AWS Backup se questo tag è presente, usate la chiave di `aws:ResourceTag` condizione. Per Lambda, utilizza una convenzione di denominazione delle risorse che include il prefisso `"Confidential"`. Quindi includi un'istruzione separata che impedisca l'eliminazione delle risorse con tale convenzione di denominazione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyDeleteSupported",
            "Effect": "Deny",
            "Action": [
                "appmesh:DeleteMesh",
                "backup:DeleteBackupPlan"
                ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/status": "Confidential"
                }
            }
        },
        {
```

```
            "Sid": "DenyDeleteUnsupported",
            "Effect": "Deny",
            "Action": "lambda:DeleteFunction",
            "Resource": "arn:aws:lambda:*:123456789012:function:status-Confidential*"
        }
    ]
}
```

> **Warning**
>
> Non utilizzate la IfExists versione... dell'operatore di condizione come soluzione alternativa per questo risultato. Questo significa "Rifiuta l'operazione se la chiave è presente nel contesto della richiesta e i valori corrispondono. Altrimenti, rifiuta l'operazione". Nell'esempio precedente, inclusa l'operazione `lambda:DeleteFunction` nell'istruzione `DenyDeleteSupported` con l'operatore `StringEqualsIfExists` rifiuta sempre sempre l'operazione. Per tale operazione, la chiave non è presente nel contesto e ogni tentativo di eliminare tale tipo di risorsa viene negato, indipendentemente dal fatto che la risorsa sia taggata o meno.

Termini correlati

- Chiavi della condizione globale
- Confronto tra ABAC e RBAC
- Elementi della policy JSON IAM: operatori di condizione
- Elemento condizione
- Panoramica delle policy JSON

Avviso di sicurezza: negazione NotAction con tag non supportato (chiave di condizione per l'assistenza)

Codice del problema:
DENY_NOTACTION_WITH_UNSUPPORTED_TAG_CONDITION_KEY_FOR_SERVICE

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Deny NotAction with unsupported tag condition key for service: Using the effect Deny
 with NotAction and the tag condition key {{conditionKeyName}} can be overly permissive
 because some service actions are not denied by this statement. This is because the
 condition key doesn't apply to some service actions. We recommend that you use Action
 instead of NotAction.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the effect Deny with NotAction and the tag condition key
 {{conditionKeyName}} can be overly permissive because some service actions are not
 denied by this statement. This is because the condition key doesn't apply to some
 service actions. We recommend that you use Action instead of NotAction."
```

Risoluzione dell'avviso di sicurezza

L'uso delle chiavi di condizione dei tag nell'elemento `Condition` di una policy con l'elemento `NotAction` e `"Effect": "Deny"` può essere eccessivamente permissivo. La condizione viene ignorata per le azioni di servizio che non supportano la chiave di condizione. AWS consiglia di riscrivere la logica per negare un elenco di azioni.

Se utilizzi la chiave di condizione `aws:ResourceTag` con `NotAction`, tutte le operazioni di servizio nuove o esistenti che non supportano la chiave non vengono rifiutate. AWS consiglia di elencare esplicitamente le operazioni che si desidera negare. Sistema di analisi degli accessi IAM restituisce una ricerca separata per le operazioni elencate che non supportano la chiave di condizione `aws:ResourceTag`. Per ulteriori informazioni, consulta Avviso di sicurezza: Rifiuta con chiave di condizione tag non supportata per il servizio.

Quando un servizio supporta la chiave di condizione `aws:ResourceTag`, è possibile utilizzare i tag per controllare l'accesso alle risorse del servizio. Questo è noto come controllo degli accessi basato su attributi (ABAC). I servizi che non supportano queste chiavi richiedono il controllo dell'accesso alle risorse tramite il controllo degli accessi basato su risorse (RBAC).

Termini correlati

- Chiavi della condizione globale
- Confronto tra ABAC e RBAC
- Elementi della policy JSON IAM: operatori di condizione

- Elemento condizione
- Panoramica delle policy JSON

Avviso di sicurezza: limita l'accesso al principale del servizio

Codice del problema: RESTRICT_ACCESS_TO_SERVICE_PRINCIPAL

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Restrict access to service principal: Granting access to a service principal without
 specifying a source is overly permissive. Use aws:SourceArn, aws:SourceAccount,
 aws:SourceOrgID, or aws:SourceOrgPaths condition key to grant fine-grained access.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Granting access to a service principal without specifying a source
 is overly permissive. Use aws:SourceArn, aws:SourceAccount, aws:SourceOrgID, or
 aws:SourceOrgPaths condition key to grant fine-grained access."
```

Risoluzione dell'avviso di sicurezza

È possibile specificare Servizi AWS nell'`Principal`elemento di una politica basata sulle risorse utilizzando un'entità di servizio, che è un identificatore del servizio. Quando concedi l'accesso a un principale del servizio per agire per conto tuo, limita l'accesso. È possibile evitare policy eccessivamente permissive utilizzando le chiavi`aws:SourceArn`,`aws:SourceAccount`, o di `aws:SourceOrgPaths` condizione per limitare l'accesso a una fonte specifica`aws:SourceOrgID`, ad esempio l'ARN di una risorsa specifica, l'ID dell'organizzazione o i percorsi dell'organizzazione. Account AWS La limitazione dell'accesso consente di prevenire un problema di sicurezza chiamato problema del "confused deputy".

Termini correlati

- Servizio AWS principali
- AWS chiavi di condizione globali: aws: SourceAccount

- AWS chiavi di condizione globali: aws: SourceArn
- AWS chiavi di condizione globali: aws: SourceOrgId
- AWS chiavi di condizione globali: aws: SourceOrgPaths
- Problema del "confused deputy"

Avviso di sicurezza: chiavi di condizione mancante per il principale oidc

Codice del problema: MISSING_CONDITION_KEY_FOR_OIDC_PRINCIPAL

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing condition key for oidc principal: Using an Open ID Connect principal without a
 condition can be overly permissive. Add condition keys with a prefix that matches your
 federated OIDC principals to ensure that only the intended identity provider assumes
 the role.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using an Open ID Connect principal without a condition can be
 overly permissive. Add condition keys with a prefix that matches your federated OIDC
 principals to ensure that only the intended identity provider assumes the role."
```

Risoluzione dell'avviso di sicurezza

L'utilizzo di un principale Open ID Connect senza una condizione può essere eccessivamente permissivo. Aggiungi chiavi di condizione con un prefisso che corrisponda ai principali OIDC federati per assicurarti che solo il provider di identità previsto assuma il ruolo.

Termini correlati

- Creazione di un ruolo per la federazione di identità Web oppure OpenID Connect (console)

Avviso di sicurezza: chiavi di condizione repository github mancanti

Codice del problema: MISSING_GITHUB_REPO_CONDITION_KEY

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Missing github repo condition key: Granting a federated GitHub principal permissions
 without a condition key can allow more sources to assume the role than you intended.
 Add the token.actions.githubusercontent.com:sub condition key and specify the branch
 and repository name in the value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Granting a federated GitHub principal permissions without a
 condition key can allow more sources to assume the role than you intended. Add the
 token.actions.githubusercontent.com:sub condition key and specify the branch and
 repository name in the value."
```

Risoluzione dell'avviso di sicurezza

Se si utilizza GitHub come IdP OIDC, la procedura migliore è limitare le entità che possono assumere il ruolo associato all'IdP IAM. Quando includi una `Condition` dichiarazione in una politica di fiducia dei ruoli, puoi limitare il ruolo a un' GitHub organizzazione, un repository o una filiale specifici. Puoi utilizzare la chiave della condizione `token.actions.githubusercontent.com:sub` per limitare l'accesso. Ti consigliamo di limitare la condizione a un insieme specifico di repository o rami. Se non includi questa condizione, GitHub le azioni provenienti da organizzazioni o repository al di fuori del tuo controllo possono assumere ruoli associati all' GitHub IdP IAM nel tuo account. AWS

Termini correlati

- Configurazione di un ruolo per GitHub il provider di identità OIDC

Avviso di sicurezza: operatore simile a una stringa con chiavi di condizione ARN

Codice del problema: STRING_LIKE_OPERATOR_WITH_ARN_CONDITION_KEYS

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
String like operator with ARN condition keys: Use the operator type {{allowed}} instead
 of operator {{operator}} for the condition key {{key}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Use the operator type {{allowed}} instead of operator {{operator}}
 for the condition key {{key}}."
```

Risoluzione dell'avviso di sicurezza

AWS consiglia di utilizzare gli operatori ARN anziché gli operatori di stringa quando si confrontano gli ARN per garantire una corretta restrizione dell'accesso in base ai valori delle condizioni ARN. Aggiorna l'operatore `StringLike` con l'operatore `ArnLike` del tuo elemento `Condition` ogni volta che viene utilizzata la chiave specificata.

Queste politiche AWS gestite sono eccezioni a questo avviso di sicurezza:

- AmazonSecurityLakeAdministrator
- AWSCodePipeline_FullAccess
- AWSCodePipeline_ReadOnlyAccess
- S3UnlockBucketPolicy
- SQSUnlockQueuePolicy

Termini correlati

- Operatori di condizione del nome della risorsa Amazon (ARN)
- Operatori di condizione stringa
- AWS politiche gestite

Avviso di sicurezza: attributi DynamoDB senza selezione

Codice di emissione: DYNAMODB_ATTRIBUTES_WITHOUT_SELECT

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
DynamoDB attributes without select: Restricting dynamodb:Attributes without also
 setting dynamodb:Select to SPECIFIC_ATTRIBUTES allows all attributes to be returned
 on read requests that omit a projection expression. We recommend that you also set
 dynamodb:Select to SPECIFIC_ATTRIBUTES.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Restricting dynamodb:Attributes without also setting dynamodb:Select
 to SPECIFIC_ATTRIBUTES allows all attributes to be returned on read requests that
 omit a projection expression. We recommend that you also set dynamodb:Select to
 SPECIFIC_ATTRIBUTES."
```

Risoluzione dell'avviso di sicurezza

Aggiungete una `StringEquals` condizione che viene `dynamodb:Select` impostata `SPECIFIC_ATTRIBUTES` nella stessa istruzione della `dynamodb:Attributes` condizione. DynamoDB valuta `dynamodb:Attributes` solo le richieste che specificano gli attributi da restituire. Senza questa condizione, una richiesta di lettura che omette un'espressione di proiezione restituisce l'intero elemento.

Termini correlati

- Utilizzo delle condizioni delle policy IAM per un controllo granulare degli accessi
- Elementi della policy JSON IAM: operatori di condizione

Avviso di sicurezza: prefisso S3 in contesto negativo

Codice di problema: S3_PREFIX_IN_NEGATIVE_CONTEXT

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
S3 prefix in negative context: Using the s3:prefix condition key in a Deny statement or
 with a negated string operator does not block listing the objects under the matching
```

```
prefixes. A request that specifies a shorter prefix does not match the condition and
can still list those objects. To restrict listing to specific prefixes, use s3:prefix
with StringLike in an Allow statement, or use a Deny statement with StringNotLike.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the s3:prefix condition key in a Deny statement or with
 a negated string operator does not block listing the objects under the matching
 prefixes. A request that specifies a shorter prefix does not match the condition and
 can still list those objects. To restrict listing to specific prefixes, use s3:prefix
 with StringLike in an Allow statement, or use a Deny statement with StringNotLike."
```

Risoluzione dell'avviso di sicurezza

Per limitare l'elenco a prefissi specifici, utilizzate uno dei seguenti modelli:

- Consenti con StringLike: concedi `s3:ListBucket` con la `StringLike` condizione di `s3:prefix` elencare i prefissi che desideri autorizzare. Le richieste che forniscono un prefisso diverso non sono coperte da questo Consenti e non possono elencare tali oggetti.
- Rifiuta con StringNotLike: nega `s3:ListBucket` con la `StringNotLike` condizione di `s3:prefix` elencare i prefissi che si desidera consentire. Una richiesta che omette il prefisso o fornisce un prefisso più breve non corrisponde al set consentito e viene rifiutata.

Un `Deny` with `StringLike` o un `Allow` with `StringNotLike` non limita efficacemente l'accesso perché un chiamante può bypassare la condizione specificando un prefisso più breve (ad esempio, `priv` anziché`private/`) che non corrisponde al valore della condizione.

Termini correlati

- Esempi di condizioni chiave di Amazon S3
- Elementi della policy JSON IAM: operatori di condizione
- Elementi della policy IAM: condizione

Avviso di sicurezza: ForAnyValue con tipo di reclamo da parte del pubblico

Codice del problema: FORANYVALUE_WITH_AUDIENCE_CLAIM_TYPE

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
ForAnyValue with audience claim type: Using ForAnyValue qualifier with the single-
valued condition key {{key}} can be overly permissive. We recommend that you remove
 ForAnyValue:.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using ForAnyValue qualifier with the single-valued condition key
 {{key}} can be overly permissive. We recommend that you remove ForAnyValue:."
```

Risoluzione dell'avviso di sicurezza

AWS consiglia di non utilizzare l'operatore `ForAnyValue` set con chiavi di condizione a valore singolo. Utilizza gli operatori di insieme solo con le chiavi della condizione multivalore. Rimuovi l'operatore di insieme `ForAnyValue`.

Termini correlati

- Single-valued rispetto alle chiavi di contesto multivalore
- Single-valued esempi di politiche chiave di contesto

Avviso di sicurezza: attributi DynamoDB senza valori restituiti

Codice di problema: DYNAMODB_ATTRIBUTES_WITHOUT_RETURN_VALUES

Tipo di esito: SECURITY_WARNING

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
DynamoDB attributes without return values: Restricting dynamodb:Attributes without
 limiting dynamodb:ReturnValues allows write operations to return the entire item,
 which can expose attributes outside the allowed list. We recommend that you restrict
 dynamodb:ReturnValues to NONE, UPDATED_OLD, or UPDATED_NEW.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Restricting dynamodb:Attributes without limiting
 dynamodb:ReturnValues allows write operations to return the entire item, which
 can expose attributes outside the allowed list. We recommend that you restrict
 dynamodb:ReturnValues to NONE, UPDATED_OLD, or UPDATED_NEW."
```

Risoluzione dell'avviso di sicurezza

Aggiungi una condizione che limiti `dynamodb:ReturnValues` a `NONEUPDATED_OLD`, o `UPDATED_NEW` nella stessa dichiarazione della `dynamodb:Attributes` condizione. Senza questo limite, una scrittura che imposta `dynamodb:ReturnValues` `ALL_OLD` o `ALL_NEW` restituisce l'intero elemento, inclusi gli attributi diversi da quelli consentiti.

Suggerimento: operazione array vuota

Codice del problema: EMPTY_ARRAY_ACTION

Tipo di esito: SUGGESTION

Ricerca dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty array action: This statement includes no actions and does not affect the policy.
 Specify actions.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This statement includes no actions and does not affect the policy.
 Specify actions."
```

Risoluzione del suggerimento

Le istruzioni devono includere un elemento `Action` o `NotAction` che include un insieme di azioni. Quando l'elemento è vuoto, l'istruzione della policy non fornisce autorizzazioni. Specifica le operazioni nell'elemento `Action`.

- Elementi delle policy JSON IAM: Action

Suggerimento: condizione array vuota

Codice del problema: EMPTY_ARRAY_CONDITION

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty array condition: There are no values for the condition key {{key}} and it does
 not affect the policy. Specify conditions.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "There are no values for the condition key {{key}} and it does not
 affect the policy. Specify conditions."
```

Risoluzione del suggerimento

La struttura dell'elemento `Condition` facoltativa richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore. Quando il valore della condizione è vuoto, la condizione restituisce `true` e l'istruzione della policy non fornisce autorizzazioni. Specifica un valore di condizione.

- Elementi della policy JSON IAM: Condition

Suggerimento: condizione di matrice vuota ForAllValues

Codice del problema: EMPTY_ARRAY_CONDITION_FORALLVALUES

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty array condition ForAllValues: The ForAllValues prefix with an empty condition key
 matches only if the key {{key}} is missing from the request context. To determine if
 the request context is empty, we recommend that you use the Null condition operator
 with the value of true instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The ForAllValues prefix with an empty condition key matches only
 if the key {{key}} is missing from the request context. To determine if the request
 context is empty, we recommend that you use the Null condition operator with the value
 of true instead."
```

Risoluzione del suggerimento

La struttura dell'elemento `Condition` richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore. L'operatore impostato `ForAllValues` verifica se il valore di ogni membro del set di richieste è un sottoinsieme del set di chiavi di condizione.

Quando si utilizza `ForAllValues` con una chiave di condizione vuota, la condizione corrisponde solo se non ci sono chiavi nella richiesta. AWS consiglia, se si desidera verificare se un contesto di richiesta è vuoto, di utilizzare invece l'operatore di condizione `Null`.

- Chiavi di contesto multivalore
- Operatore di condizione null
- Elementi della policy JSON IAM: Condition

Suggerimento: condizione di matrice vuota ForAnyValue

Codice del problema: EMPTY_ARRAY_CONDITION_FORANYVALUE

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty array condition ForAnyValue: The ForAnyValue prefix with an empty condition key
 {{key}} never matches the request context and it does not affect the policy. Specify
 conditions.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The ForAnyValue prefix with an empty condition key {{key}} never
 matches the request context and it does not affect the policy. Specify conditions."
```

Risoluzione del suggerimento

La struttura dell'elemento `Condition` richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore. L'operatore impostato `ForAnyValues` verifica se almeno un membro del set di valori di richiesta è corrispondente ad almeno un membro del set di valori delle chiavi di condizione.

Quando utilizzi `ForAnyValues` con una chiave di condizione vuota, la condizione non corrisponde mai. Ciò significa che la dichiarazione non ha alcun effetto sulla politica. AWS consiglia di riscrivere la condizione.

- Chiavi di contesto multivalore
- Elementi della policy JSON IAM: Condition

Suggerimento: condizione di matrice vuota IfExists

Codice del problema: EMPTY_ARRAY_CONDITION_IFEXISTS

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty array condition IfExists: The IfExists suffix with an empty condition key matches
 only if the key {{key}} is missing from the request context. To determine if the
 request context is empty, we recommend that you use the Null condition operator with
 the value of true instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The IfExists suffix with an empty condition key matches only if the
 key {{key}} is missing from the request context. To determine if the request context
 is empty, we recommend that you use the Null condition operator with the value of true
 instead."
```

Risoluzione del suggerimento

Il suffisso `...IfExists` modifica un operatore di condizione. Ciò significa che se la chiave di policy è presente nel contesto della richiesta, la chiave deve essere elaborata come specificato nella policy. Se la chiave non è presente, l'elemento della condizione viene valutato come true (VERO).

Quando si utilizza `...IfExists` con una chiave di condizione vuota, la condizione corrisponde solo se non ci sono chiavi nella richiesta. AWS consiglia, se si desidera verificare se un contesto di richiesta è vuoto, di utilizzare invece l'operatore di condizione `Null`.

- ... IfExists operatori di condizione
- Elementi della policy JSON IAM: Condition

Suggerimento: principale array vuoto

Codice del problema: EMPTY_ARRAY_PRINCIPAL

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty array principal: This statement includes no principals and does not affect the
 policy. Specify principals.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This statement includes no principals and does not affect the
 policy. Specify principals."
```

Risoluzione del suggerimento

È necessario utilizzare l'`NotPrincipal`elemento `Principal` or nelle politiche di fiducia per i ruoli IAM e nelle politiche basate sulle risorse. Resource-based le policy sono policy che puoi incorporare direttamente in una risorsa.

Quando si fornisce un array vuoto nell'`Principal`elemento di un'istruzione, l'istruzione non ha alcun effetto sulla politica. AWS consiglia di specificare i principali che devono avere accesso alla risorsa.

- Elementi delle policy JSON IAM: Principal
- Elementi della policy JSON IAM: NotPrincipal

Suggerimento: risorsa array vuota

Codice del problema: EMPTY_ARRAY_RESOURCE

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty array resource: This statement includes no resources and does not affect the
 policy. Specify resources.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This statement includes no resources and does not affect the policy.
 Specify resources."
```

Risoluzione del suggerimento

Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`.

Quando si fornisce un array vuoto nell'elemento risorsa di un'istruzione, l'istruzione non ha alcun effetto sulla policy. AWS consiglia di specificare Amazon Resource Names (ARN) per le risorse.

- Elementi delle policy JSON IAM: Resource
- Elementi della policy IAM JSON: NotResource

Suggerimento: condizione oggetto vuota

Codice del problema: EMPTY_OBJECT_CONDITION

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty object condition: This condition block is empty and it does not affect the
 policy. Specify conditions.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This condition block is empty and it does not affect the policy.
 Specify conditions."
```

Risoluzione del suggerimento

La struttura dell'elemento `Condition` richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore.

Quando si specifica un oggetto vuoto nell'elemento di condizione di un'istruzione, l'istruzione non ha alcun effetto sulla policy. Rimuovi l'elemento facoltativo o specifica le condizioni.

- Elementi della policy JSON IAM: Condition

Suggerimento: principale oggetto vuoto

Codice del problema: EMPTY_OBJECT_PRINCIPAL

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty object principal: This statement includes no principals and does not affect the
 policy. Specify principals.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This statement includes no principals and does not affect the
 policy. Specify principals."
```

Risoluzione del suggerimento

È necessario utilizzare l'`NotPrincipal`elemento `Principal` or nelle politiche di fiducia per i ruoli IAM e nelle politiche basate sulle risorse. Resource-based le policy sono policy che puoi incorporare direttamente in una risorsa.

Quando si fornisce un oggetto vuoto nell'`Principal`elemento di un'istruzione, l'istruzione non ha alcun effetto sulla politica. AWS consiglia di specificare i principali che devono avere accesso alla risorsa.

- Elementi delle policy JSON IAM: Principal
- Elementi della policy JSON IAM: NotPrincipal

Suggerimento: valore sid vuoto

Codice del problema: EMPTY_SID_VALUE

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Empty Sid value: Add a value to the empty string in the Sid element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a value to the empty string in the Sid element."
```

Risoluzione del suggerimento

L'elemento `Sid` (ID istruzione) facoltativo consente di immettere un identificatore fornito per l'istruzione della policy. Puoi assegnare un valore `Sid` a ogni istruzione in un array di istruzioni. Se scegli di utilizzare l'elemento `Sid`, devi fornire un valore di stringa.

Termini correlati

- Elementi delle policy JSON IAM: Sid

Suggerimento: equivalente a null false

Codice del problema: EQUIVALENT_TO_NULL_FALSE

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Equivalent to null false: We recommend replacing the key {{key}} in the condition block
 of {{operator}} with {{{recommendedKey}}: false} to ensure better enforcement of the
 condition.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "We recommend replacing the key {{key}} in the condition block of
 {{operator}} with {{{recommendedKey}}: false} to ensure better enforcement of the
 condition."
```

Risoluzione del suggerimento

Sostituisci la chiave della condizione corrente con la chiave consigliata impostata su `false`. Questa modifica migliora la chiarezza delle policy e garantisce una valutazione delle condizioni più affidabile. Aggiorna il blocco delle condizioni per utilizzare `{recommendedKey}: false` al posto dell'attuale combinazione chiave-operatore.

Termini correlati

- Elementi della policy IAM: condizione
- Condizioni con più chiavi o valori di contesto
- AWS chiavi di contesto delle condizioni globali

Suggerimento: equivalente a null true

Codice del problema: EQUIVALENT_TO_NULL_TRUE

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Equivalent to null true: We recommend replacing the key {{key}} in the condition block
 of {{operator}} with {{{recommendedKey}}: true} to ensure better enforcement of the
 condition.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "We recommend replacing the key {{key}} in the condition block of
 {{operator}} with {{{recommendedKey}}: true} to ensure better enforcement of the
 condition."
```

Risoluzione del suggerimento

Sostituisci la chiave della condizione corrente con la chiave consigliata impostata su `true`. Questa modifica migliora la chiarezza delle policy e garantisce una valutazione delle condizioni più affidabile. Aggiorna il blocco delle condizioni per utilizzare `{recommendedKey}: true` al posto dell'attuale combinazione chiave-operatore.

Termini correlati

- Elementi della policy IAM: condizione
- Condizioni con più chiavi o valori di contesto
- AWS chiavi di contesto delle condizioni globali

Suggerimento: migliora l'intervallo IP

Codice del problema: IMPROVE_IP_RANGE

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Improve IP range: The non-zero bits in the IP address after the masked bits are
 ignored. Replace address with {{addr}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The non-zero bits in the IP address after the masked bits are
 ignored. Replace address with {{addr}}."
```

Risoluzione del suggerimento

Le condizioni dell'indirizzo IP devono essere nel formato CIDR standard, ad esempio 203.0.113. 0/24 o 2001: :/64. DB8:1234:5678 Quando si includono bit diversi da zero dopo i bit mascherati, questi non vengono considerati per la condizione. AWS consiglia di utilizzare il nuovo indirizzo incluso nel messaggio.

- Operatori di condizione indirizzo IP
- Elementi della policy JSON IAM: Condition

Suggerimento: null con qualificatore

Codice del problema: NULL_WITH_QUALIFIER

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Null with qualifier: Avoid using the Null condition operator with the ForAllValues or
 ForAnyValue qualifiers because they always return a true or false respectively.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Avoid using the Null condition operator with the ForAllValues or
 ForAnyValue qualifiers because they always return a true or false respectively."
```

Risoluzione del suggerimento

Nell'elemento `Condition` è possibile creare espressioni in cui utilizzare operatori condizionali ("uguale a", "minore di" e così via) per confrontare le chiavi e i valori della policy rispetto alle chiavi e

ai valori del contesto della richiesta. Per le richieste che includono più valori per una singola chiave di condizione, è necessario utilizzare gli operatori su set `ForAllValues` o `ForAnyValue`.

Quando si utilizza l'operatore di condizione `Null` con `ForAllValues`, l'istruzione restituisce sempre `true`. Quando si utilizza l'operatore di `Null` condizione con`ForAnyValue`, l'istruzione viene sempre `false` restituita. AWS consiglia di utilizzare l'operatore di `StringLike` condizione con questi operatori impostati.

Termini correlati

- Chiavi di contesto multivalore
- Operatore di condizione null
- Elemento condizione

Suggerimento: sottoinsieme di indirizzi IP privati

Codice del problema: PRIVATE_IP_ADDRESS_SUBSET

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Private IP address subset: The values for condition key aws:SourceIp include a mix
 of private and public IP addresses. The private addresses will not have the desired
 effect. aws:SourceIp works only for public IP address ranges. To define permissions
 for private IP ranges, use aws:VpcSourceIp.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The values for condition key aws:SourceIp include a mix of private
 and public IP addresses. The private addresses will not have the desired effect.
 aws:SourceIp works only for public IP address ranges. To define permissions for
 private IP ranges, use aws:VpcSourceIp."
```

Risoluzione del suggerimento

La chiave di condizione globale `aws:SourceIp` funziona solo per intervalli di indirizzi IP pubblici.

Se il tuo elemento `Condition` include un mix di indirizzi IP privati e pubblici, l'istruzione potrebbe non avere l'effetto desiderato. Non puoi specificare indirizzi IP privati utilizzando `aws:VpcSourceIP`.

> **Note**
>
> La chiave di condizione globale `aws:VpcSourceIP` corrisponde solo se la richiesta proviene dall'indirizzo IP specificato e passa attraverso un endpoint VPC.

- aws: chiave di condizione SourceIp globale
- aws: chiave di condizione VpcSourceIp globale
- Operatori di condizione indirizzo IP
- Elementi della policy JSON IAM: Condition

Suggerimento: sottoinsieme privato NotIpAddress

Codice del problema: PRIVATE_NOT_IP_ADDRESS_SUBSET

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Private NotIpAddress subset: The values for condition key aws:SourceIp include a mix of
 private and public IP addresses. The private addresses have no effect. aws:SourceIp
 works only for public IP address ranges. To define permissions for private IP ranges,
 use aws:VpcSourceIp.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The values for condition key aws:SourceIp include a mix of private
 and public IP addresses. The private addresses have no effect. aws:SourceIp works
 only for public IP address ranges. To define permissions for private IP ranges, use
 aws:VpcSourceIp."
```

Risoluzione del suggerimento

La chiave di condizione globale `aws:SourceIp` funziona solo per intervalli di indirizzi IP pubblici.

Se il tuo elemento `Condition` include l'operatore di condizione `NotIpAddress` e un mix di indirizzi IP privati e pubblici, l'istruzione potrebbe non avere l'effetto desiderato. Ogni indirizzo IP pubblico non specificato nella policy corrisponderà. Nessun indirizzo IP privato corrisponderà. Per ottenere questo effetto, puoi usare `NotIpAddress` con `aws:VpcSourceIP` e specificare gli indirizzi IP privati che non devono corrispondere.

- aws: chiave di condizione SourceIp globale
- aws: chiave di condizione VpcSourceIp globale
- Operatori di condizione indirizzo IP
- Elementi della policy JSON IAM: Condition

Suggerimento: azione ridondante

Codice del problema: REDUNDANT_ACTION

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Redundant action: The {{redundantActionCount}} action(s) are redundant because they
 provide similar permissions. Update the policy to remove the redundant action such as:
 {{redundantAction}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{redundantActionCount}} action(s) are redundant because they
 provide similar permissions. Update the policy to remove the redundant action such as:
 {{redundantAction}}."
```

Risoluzione del suggerimento

Quando si utilizzano i caratteri jolly (*) nell'`Action`elemento, è possibile includere autorizzazioni ridondanti. AWS consiglia di rivedere la politica e di includere solo le autorizzazioni necessarie. In questo modo è possibile rimuovere le operazioni ridondanti.

Ad esempio, le operazioni riportate di seguito includono due volte l'operazione `iam:GetCredentialReport`.

```
"Action": [
        "iam:Get*",
        "iam:List*",
        "iam:GetCredentialReport"
    ],
```

In questo esempio, le autorizzazioni sono definite per ogni operazione IAM che inizia con `Get` o `List`. Quando IAM aggiunge ulteriori operazioni get o list, questa policy le consentirà. Potresti voler consentire tutte queste azioni di sola lettura. L'operazione `iam:GetCredentialReport` è già inclusa come parte di `iam:Get*`. Per rimuovere le autorizzazioni duplicate, puoi rimuovere `iam:GetCredentialReport`.

Quando tutti i contenuti di un'operazione sono ridondanti, viene visualizzato un risultato per questo controllo delle policy. In questo esempio, se l'elemento includeva `iam:*CredentialReport`, non è considerato ridondante. Ciò include `iam:GetCredentialReport`, che è ridondante, e `iam:GenerateCredentialReport`, che non lo è. La rimozione di `iam:Get*` o `iam:*CredentialReport` modificherebbe le autorizzazioni della policy.

- Elementi delle policy JSON IAM: Action

AWS politiche gestite con questo suggerimento

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Le operazioni ridondanti non influenzano le autorizzazioni concesse dalla policy. Quando si utilizza una policy AWS gestita come riferimento per creare una policy gestita dal cliente, AWS consiglia di rimuovere le azioni ridondanti dalla policy.

Suggerimento: valore condizione ridondante num

Codice del problema: REDUNDANT_CONDITION_VALUE_NUM

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Redundant condition value num: Multiple values in {{operator}} are redundant. Replace
 with the {{greatest/least}} single value for {{key}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Multiple values in {{operator}} are redundant. Replace with the
 {{greatest/least}} single value for {{key}}."
```

Risoluzione del suggerimento

Quando si utilizzano operatori di condizioni numeriche per valori simili in una chiave di condizione, è possibile creare una sovrapposizione che si traduce in autorizzazioni ridondanti.

Ad esempio, il seguente elemento `Condition` include più condizioni `aws:MultiFactorAuthAge` che hanno una sovrapposizione di età di 1200 secondi.

```
"Condition": {
        "NumericLessThan": {
          "aws:MultiFactorAuthAge": [
            "2700",
            "3600"
          ]
        }
      }
```

In questo esempio, le autorizzazioni vengono definite se l'autenticazione a più fattori (MFA) è stata completata meno di 3600 secondi (1 ora) fa. È possibile rimuovere il valore `2700` ridondante.

- Operatori di condizione numerici
- Elementi della policy JSON IAM: Condition

Suggerimento: risorsa ridondante

Codice del problema: REDUNDANT_RESOURCE

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Redundant resource: The {{redundantResourceCount}} resource ARN(s) are redundant
 because they reference the same resource. Review the use of wildcards (*)
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{redundantResourceCount}} resource ARN(s) are redundant because
 they reference the same resource. Review the use of wildcards (*)"
```

Risoluzione del suggerimento

Quando usi i caratteri jolly (*) in Amazon Resource Name (ARN), puoi creare autorizzazioni per le risorse ridondanti.

Ad esempio, il seguente elemento `Resource` include più ARN con autorizzazioni ridondanti.

```
"Resource": [
            "arn:aws:iam::111122223333:role/jane-admin",
            "arn:aws:iam::111122223333:role/jane-s3only",
            "arn:aws:iam::111122223333:role/jane*"
        ],
```

In questo esempio, le autorizzazioni sono definite per qualsiasi ruolo con un nome che inizia con `jane`. È possibile rimuovere gli ARN `jane-admin` e `jane-s3only` ridondanti senza modificare le autorizzazioni risultanti. Questo rende la policy dinamica. Definirà le autorizzazioni per tutti i ruoli futuri che iniziano con `jane`. Se l'intenzione della policy è consentire l'accesso a un numero statico di ruoli, rimuovere l'ultimo ARN ed elencare solo gli ARN da definire.

- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo suggerimento

AWS le policy gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi d'uso generali AWS .

Le operazioni ridondanti non influenzano le autorizzazioni concesse dalla policy. Quando si utilizza una policy AWS gestita come riferimento per creare una policy gestita dal cliente, AWS consiglia di rimuovere le risorse ridondanti dalla policy.

Suggerimento: istruzione ridondante

Codice del problema: REDUNDANT_STATEMENT

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Redundant statement: The statements are redundant because they provide identical
 permissions. Update the policy to remove the redundant statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The statements are redundant because they provide identical
 permissions. Update the policy to remove the redundant statement."
```

Risoluzione del suggerimento

L'elemento `Statement` è l'elemento principale per una policy. Questa elemento è obbligatorio. L'elemento `Statement` può contenere una singola istruzione o una matrice di singole istruzioni.

Quando si include la stessa istruzione più di una volta in una policy lunga, le istruzioni sono ridondanti. È possibile rimuovere una delle istruzioni senza influire sulle autorizzazioni concesse dalla policy. Quando un utente modifica una policy, potrebbe modificare una delle istruzioni senza aggiornare il duplicato. Ciò potrebbe comportare un numero di autorizzazioni maggiore del previsto.

- Elementi delle policy IAM JSON: istruzione

Suggerimento: carattere jolly nel nome del servizio

Codice del problema: WILDCARD_IN_SERVICE_NAME

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Wildcard in service name: Avoid using wildcards (*, ?) in the service name because it
 might grant unintended access to other AWS services with similar names.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Avoid using wildcards (*, ?) in the service name because it might
 grant unintended access to other AWS services with similar names."
```

Risoluzione del suggerimento

Quando si include il nome di un AWS servizio in una policy, si AWS consiglia di non includere caratteri jolly (*,?). Ciò potrebbe aggiungere autorizzazioni per servizi futuri non previsti. Ad esempio, ci sono più di una dozzina di AWS servizi con la parola `*code*` nel nome.

```
"Resource": "arn:aws:*code*::111122223333:*"
```

- Elementi delle policy JSON IAM: Resource

Suggerimento: consenti con chiave di condizione tag non supportata per il servizio

Codice del problema: ALLOW_WITH_UNSUPPORTED_TAG_CONDITION_KEY_FOR_SERVICE

Tipo di esito: SUGGESTION

Ricerca dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Allow with unsupported tag condition key for service: Using the effect Allow with the
 tag condition key {{conditionKeyName}} and actions for services with the following
 prefixes does not affect the policy: {{serviceNames}}. Actions for the listed service
 are not allowed by this statement. We recommend that you move these actions to a
 different statement without this condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the effect Allow with the tag condition key
 {{conditionKeyName}} and actions for services with the following prefixes does not
```

```
affect the policy: {{serviceNames}}. Actions for the listed service are not allowed
by this statement. We recommend that you move these actions to a different statement
without this condition key."
```

Risoluzione del suggerimento

L'utilizzo di chiavi di condizione dei tag non supportate nell'`Condition`elemento di una policy con non `"Effect": "Allow"` influisce sulle autorizzazioni concesse dalla policy, poiché la condizione viene ignorata per quell'azione di servizio. AWS consiglia di rimuovere le azioni per i servizi che non supportano la chiave di condizione e di creare un'altra dichiarazione per consentire l'accesso a risorse specifiche in quel servizio.

Se utilizzi la chiave di condizione `aws:ResourceTag` e non è supportata da un'operazione di servizio, la chiave non viene inclusa nel contesto della richiesta. In questo caso, la condizione nell'istruzione `Allow` restituisce sempre `false` e l'operazione non viene mai rifiutata. Ciò accade anche se la risorsa è taggata correttamente.

Quando un servizio supporta la chiave di condizione `aws:ResourceTag`, è possibile utilizzare i tag per controllare l'accesso alle risorse del servizio. Questo è noto come controllo degli accessi basato su attributi (ABAC). I servizi che non supportano queste chiavi richiedono il controllo dell'accesso alle risorse tramite il controllo degli accessi basato su risorse (RBAC).

> **Note**
>
> Alcuni servizi consentono il supporto per la chiave di condizione `aws:ResourceTag` per un sottoinsieme di risorse e operazioni. Sistema di analisi degli accessi IAM restituisce risultati per le operazioni di servizio non supportate. Ad esempio, Amazon S3 supporta `aws:ResourceTag` per un sottoinsieme delle relative risorse. Per visualizzare tutti i tipi di risorse disponibili in Amazon S3 che supportano la chiave di condizione `aws:ResourceTag`, consulta Tipi di risorse definiti da Amazon S3 in Service Authorization Reference.

Ad esempio, supponiamo che tu desideri consentire ai membri del team di visualizzare i dettagli per le risorse specifiche che sono taggate con la coppia chiave-valore `team=BumbleBee`. Si supponga inoltre che AWS Lambda consenta di etichettare le risorse, ma non supporti la chiave `aws:ResourceTag` condizionale. Per consentire le azioni di visualizzazione per AWS App Mesh e AWS Backup se questo tag è presente, utilizzate la chiave di `aws:ResourceTag` condizione. Per Lambda, utilizza una convenzione di denominazione delle risorse che include il nome del team come

prefisso. Quindi includi un'istruzione separata che consenta la visualizzazione delle risorse con tale convenzione di denominazione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewSupported",
            "Effect": "Allow",
            "Action": [
                "appmesh:DescribeMesh",
                "backup:GetBackupPlan"
                ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/team": "BumbleBee"
                }
            }
        },
        {
            "Sid": "AllowViewUnsupported",
            "Effect": "Allow",
            "Action": "lambda:GetFunction",
            "Resource": "arn:aws:lambda:*:123456789012:function:team-BumbleBee*"
        }
    ]
}
```

Warning

Non utilizzare la `Not` versione dell'operatore di condizione con `"Effect": "Allow"` come soluzione alternativa per questo risultato. Questi operatori di condizione forniscono la corrispondenza negata. Ciò significa che dopo che la condizione è stata valutata, il risultato viene negato. Nell'esempio precedente, che include l'operazione `lambda:GetFunction` nell'istruzione `AllowViewSupported` con l'operatore `StringNotEquals` consente sempre l'operazione, indipendentemente dal fatto che la risorsa sia taggata o meno.
Non utilizzate la `IfExists` versione... dell'operatore di condizione come soluzione alternativa per questo risultato. Questo significa "Consenti l'operazione se la chiave è presente nel contesto della richiesta e i valori corrispondono. Altrimenti, autorizza l'operazione." Nell'esempio precedente, inclusa l'operazione `lambda:GetFunction` nell'istruzione

`AllowViewSupported` con l'operatore `StringEqualsIfExists` consente sempre l'operazione. Per tale operazione, la chiave non è presente nel contesto e ogni tentativo di visualizzare tale tipo di risorsa viene negato, indipendentemente dal fatto che la risorsa sia taggata o meno.

Termini correlati

- Chiavi della condizione globale
- Elementi della policy JSON IAM: operatori di condizione
- Elemento condizione
- Panoramica delle policy JSON

Suggerimento: consentire l'utilizzo di una chiave NotAction condizionale con tag non supportata per l'assistenza

Codice del problema:
ALLOW_NOTACTION_WITH_UNSUPPORTED_TAG_CONDITION_KEY_FOR_SERVICE

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Allow NotAction with unsupported tag condition key for service: Using the effect Allow
 with NotAction and the tag condition key {{conditionKeyName}} allows only service
 actions that support the condition key. The condition key doesn't apply to some
 service actions. We recommend that you use Action instead of NotAction.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the effect Allow with NotAction and the tag condition key
 {{conditionKeyName}} allows only service actions that support the condition key. The
 condition key doesn't apply to some service actions. We recommend that you use Action
 instead of NotAction."
```

Risoluzione del suggerimento

L'uso delle chiavi di condizione dei tag non supportate nell'elemento `Condition` di una policy con l'elemento `NotAction` e `"Effect": "Allow"` non influisce sulle autorizzazioni concesse dai policy. La condizione viene ignorata per le azioni di servizio che non supportano la chiave di condizione. AWS consiglia di riscrivere la logica per consentire un elenco di azioni.

Se utilizzi la chiave di condizione `aws:ResourceTag` con `NotAction`, tutte le operazioni di servizio nuove o esistenti che non supportano la chiave non vengono rifiutate. AWS consiglia di elencare esplicitamente le operazioni che si desidera consentire. Sistema di analisi degli accessi AWS IAM restituisce una ricerca separata per le operazioni elencate che non supportano la chiave di condizione `aws:ResourceTag`. Per ulteriori informazioni, consulta Suggerimento: consenti con chiave di condizione tag non supportata per il servizio.

Quando un servizio supporta la chiave di condizione `aws:ResourceTag`, è possibile utilizzare i tag per controllare l'accesso alle risorse del servizio. Questo è noto come controllo degli accessi basato su attributi (ABAC). I servizi che non supportano queste chiavi richiedono il controllo dell'accesso alle risorse tramite il controllo degli accessi basato su risorse (RBAC).

Termini correlati

- Chiavi della condizione globale
- Confronto tra ABAC e RBAC
- Elementi della policy JSON IAM: operatori di condizione
- Elemento condizione
- Panoramica delle policy JSON

Suggerimento: chiave di condizione consigliata per il principale del servizio

Codice del problema: RECOMMENDED_CONDITION_KEY_FOR_SERVICE_PRINCIPAL

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Recommended condition key for service principal: To restrict access to the service
 principal {{servicePrincipalPrefix}} operating on your behalf, we recommend
```

```
aws:SourceArn, aws:SourceAccount, aws:SourceOrgID, or aws:SourceOrgPaths instead of
{{key}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "To restrict access to the service principal
 {{servicePrincipalPrefix}} operating on your behalf, we recommend aws:SourceArn,
 aws:SourceAccount, aws:SourceOrgID, or aws:SourceOrgPaths instead of {{key}}."
```

Risoluzione del suggerimento

È possibile specificare Servizi AWS nell'`Principal`elemento di una politica basata sulle risorse utilizzando un'entità di servizio, che è un identificatore del servizio. Quando si concede l'accesso ai principali del servizio, è consigliabile utilizzare le chiavi di condizione `aws:SourceArn`, `aws:SourceAccount`, `aws:SourceOrgID` o `aws:SourceOrgPaths` anziché altre chiavi di condizione, come `aws:Referer`. Questo aiuta a prevenire un problema di sicurezza chiamato problema del "confused deputy".

Termini correlati

- Servizio AWS presidi
- AWS chiavi di condizione globali: aws: SourceAccount
- AWS chiavi di condizione globali: aws: SourceArn
- AWS chiavi di condizione globali: aws: SourceOrgId
- AWS chiavi di condizione globali: aws: SourceOrgPaths
- Problema del "confused deputy"

Suggerimento: chiave di condizione irrilevante nella policy

Codice del problema: IRRELEVANT_CONDITION_KEY_IN_POLICY

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Irrelevant condition key in policy: The condition key {{condition-key}} is not relevant
 for the {{resource-type}} policy.  Use this key in an identity-based policy to govern
 access to this resource.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{condition-key}} is not relevant for the
 {{resource-type}} policy.  Use this key in an identity-based policy to govern access
 to this resource."
```

Risoluzione del suggerimento

Alcune chiavi di condizione non sono rilevanti per le policy basate sulle risorse. Ad esempio, la chiave di condizione `s3:ResourceAccount` non è rilevante per la polocy basata sulle risorse collegata a un tipo di risorsa bucket Amazon S3 o a un punto di accesso Amazon S3.

Puoi utilizzare la chiave di condizione nella policy basata sulle identità per controllare l'accesso alla risorsa.

Termini correlati

- Identity-based politiche e politiche basate sulle risorse

Suggerimento: chiave ridondante a causa della presenza di un carattere jolly

Codice del problema: REDUNDANT_KEY_DUE_TO_WILDCARD_IN_CONDITION

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Redundant key due to wildcard in condition: The key {{key}} in the condition block of
 {{operator}} is redundant because it is always matched. Remove this key to simplify
 the condition.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The key {{key}} in the condition block of {{operator}} is redundant
 because it is always matched. Remove this key to simplify the condition."
```

Risoluzione del suggerimento

Rimuovi la chiave di condizione ridondante dalla policy. La chiave viene sempre corrisposta a causa dello schema dei caratteri jolly, il che la rende superflua. Semplifica il blocco delle condizioni rimuovendo questa chiave pur mantenendo le stesse autorizzazioni effettive.

Termini correlati

- Elementi della policy IAM: condizione
- Condizioni con più chiavi o valori di contesto
- AWS chiavi di contesto delle condizioni globali

Suggerimento: principale ridondante nella policy di attendibilità del ruolo

Codice del problema: REDUNDANT_PRINCIPAL_IN_ROLE_TRUST_POLICY

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Redundant principal in role trust policy: The assumed-role principal
 {{redundant_principal}} is redundant with its parent role {{parent_role}}. Remove the
 assumed-role principal.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The assumed-role principal {{redundant_principal}} is redundant with
 its parent role {{parent_role}}. Remove the assumed-role principal."
```

Risoluzione del suggerimento

Se si specifica sia un principale con ruolo assunto che il suo ruolo padre nell'elemento `Principal` di una policy, non consente o nega autorizzazioni diverse. Ad esempio, è ridondante se si specifica l'elemento `Principal` utilizzando il seguente formato:

```
"Principal": {
            "AWS": [
            "arn:aws:iam::AWS-account-ID:role/rolename",
            "arn:aws:iam::AWS-account-ID:assumed-role/rolename/rolesessionname"
        ]
```

Si consiglia di rimuovere il principale del ruolo assunto.

Termini correlati

- Principali della sessione come ruolo

Suggerimento: istruzione ridondante a causa della presenza di un carattere jolly

Codice del problema: REDUNDANT_STATEMENT_DUE_TO_WILDCARD_IN_CONDITION

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Redundant statement due to wildcard in condition: The key {{key}} in the condition
 block of {{operator}} does not match any values. Remove this key to simplify the
 condition.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The key {{key}} in the condition block of {{operator}} does not
 match any values. Remove this key to simplify the condition."
```

Risoluzione del suggerimento

Rimuovi la chiave di condizione che non corrisponde a nessun valore. Questa chiave crea una condizione irraggiungibile che non sarà mai soddisfatta, rendendola ridondante. Pulisci la tua policy rimuovendo questa chiave per migliorare la leggibilità e le prestazioni.

Termini correlati

- Elementi della policy IAM: condizione
- Condizioni con più chiavi o valori di contesto
- AWS chiavi di contesto delle condizioni globali

Suggerimento: conferma il tipo di attestazione del pubblico

Codice del problema: CONFIRM_AUDIENCE_CLAIM_TYPE

Tipo di esito: SUGGESTION

Individuazione dei dettagli

Nel Console di gestione AWS, il risultato di questo controllo include il seguente messaggio:

```
Confirm audience claim type: The "{{key}}" ({{audienceType}}) claim key identifies
 the recipients that the JSON web token is intended for. Because this claim is single-
valued, do not use a qualifier.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The "{{key}}" ({{audienceType}}) claim key identifies the recipients
 that the JSON web token is intended for. Because this claim is single-valued, do not
 use a qualifier."
```

Risoluzione del suggerimento

La chiave di attestazione `aud` (destinatario) è un identificatore univoco per l'app rilasciato durante la registrazione dell'app con IdP. Identifica i destinatari del token Web JSON. Le attestazioni del pubblico possono essere multivalore o a valore singolo. Se l'attestazione è multivalore, utilizza un'operatore di condizione `ForAllValues` o `ForAnyValue`. Se l'attestazione ha un valore singolo, non utilizzare un operatore di condizione.

Termini correlati

- Creazione di un ruolo per la federazione di identità Web oppure OpenID Connect (console)
- Chiavi di contesto multivalore
- Single-valued rispetto alle chiavi di condizione multivalore

Convalidare le policy utilizzando i controlli delle policy personalizzate di Sistema di analisi degli accessi IAM

Puoi utilizzare i controlli delle policy personalizzati per verificare la presenza di nuovi accessi in base ai tuoi standard di sicurezza. Viene addebitato un costo per ogni controllo di un nuovo accesso. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi IAM.

Convalida delle policy con controlli delle policy personalizzati (console)

Come passaggio facoltativo, è possibile eseguire un controllo delle policy personalizzato durante la modifica di una policy nell'editor di policy JSON nella console IAM. Puoi verificare se la policy aggiornata concede un nuovo accesso rispetto alla versione esistente.

Per verificare la presenza di nuovi accessi durante la modifica delle policy JSON IAM

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Nell'elenco delle policy, seleziona il nome della policy che desideri modificare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Seleziona la scheda Autorizzazioni e scegli Modifica.
5. Scegli l'opzione JSON e aggiorna la tua policy.
6. Nel riquadro di convalida delle policy sotto la policy, scegli la scheda Verifica nuovi accessi e seleziona Verifica policy. Se le autorizzazioni modificate concedono un nuovo accesso, l'istruzione verrà evidenziata nel riquadro di convalida della policy.
7. Se non intendi concedere un nuovo accesso, aggiorna le istruzioni di policy e scegli Verifica policy finché non viene rilevato alcun nuovo accesso.

> **Note**
>
> Viene addebitato un costo per ogni controllo di un nuovo accesso. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

8. Scegli Next (Successivo).
9. Nella pagina Verifica e salva, esamina il campo Autorizzazioni definite in questa policy, quindi scegli Salva modifiche.

Convalida delle politiche con controlli personalizzati (AWS CLI o API)

Puoi eseguire controlli delle policy personalizzate di IAM Access Analyzer dall'API IAM Access Analyzer AWS CLI o dall'API IAM Access Analyzer.

Per eseguire controlli delle policy personalizzati di Sistema di analisi degli accessi IAM (AWS CLI)

- Per verificare se è consentito un nuovo accesso per una policy aggiornata rispetto alla policy esistente, esegui il seguente comando: check-no-new-access
- Per verificare se l'accesso specificato non è consentito da una policy, esegui il comando seguente: check-access-not-granted
- Per verificare se una policy delle risorse può concedere l'accesso pubblico a un tipo di risorsa specificato, esegui il comando seguente: check-no-public-access

Per eseguire controlli delle policy personalizzati di Sistema di analisi degli accessi IAM (API)

- Per verificare se è consentito un nuovo accesso per una policy aggiornata rispetto alla policy esistente, utilizza l'operazione API CheckNoNewAccess.
- Per verificare se l'accesso specificato non è consentito da una policy, utilizza l'operazione API CheckAccessNotGranted.
- Per verificare se una policy delle risorse può concedere l'accesso pubblico a un tipo di risorsa specificato, utilizza l'operazione API CheckNoPublicAccess.

Generazione di policy per Sistema di analisi degli accessi IAM

In qualità di amministratore o sviluppatore, puoi concedere autorizzazioni a entità IAM (utenti o ruoli) che vanno oltre quanto richiesto. IAM fornisce diverse opzioni che consentono di perfezionare le autorizzazioni concesse.

Un'opzione consiste nel generare una policy IAM basata sull'attività di accesso per un'entità. IAM Access Analyzer esamina AWS CloudTrail i tuoi log e genera un modello di policy che contiene le autorizzazioni utilizzate dall'entità nell'intervallo di date specificato. È possibile utilizzare il modello per creare una policy con autorizzazioni granulari che concedono solo le autorizzazioni necessarie per supportare il caso d'uso specifico.

Argomenti

- Come funziona la generazione di policy

- Informazioni sul servizio e sul livello di azione
- Da sapere sulla generazione di policy
- Autorizzazioni richieste per generare una policy
- Genera una policy basata sull' CloudTrail attività (console)
- Genera una policy utilizzando AWS CloudTrail dati in un altro account
- Genera una policy basata sull' CloudTrail attività (AWS CLI)
- Genera una politica basata sull' CloudTrail attività (AWS API)
- Servizi di generazione di policy per Sistema di analisi degli accessi IAM

Come funziona la generazione di policy

IAM Access Analyzer analizza CloudTrail gli eventi per identificare le azioni e i servizi che sono stati utilizzati da un'entità IAM (utente o ruolo). Viene quindi generata una policy IAM basata su tale attività. È possibile perfezionare le autorizzazioni di un'entità quando si sostituisce una policy di autorizzazioni generali associata all'entità con la policy generata. Di seguito si riporta una panoramica di alto livello del processo di generazione della policy.

- Configurato per la generazione di modelli di policy: specifichi un periodo di tempo massimo di 90 giorni per consentire a IAM Access Analyzer di analizzare la cronologia degli eventi. AWS CloudTrail È necessario specificare un ruolo del servizio esistente o crearne uno nuovo. Il ruolo di servizio consente a IAM Access Analyzer di accedere alla CloudTrail cronologia e alle informazioni relative all'ultimo accesso al servizio per identificare i servizi e le azioni utilizzati. È necessario specificare il CloudTrail percorso che registra gli eventi per l'account prima di poter generare una policy. Per ulteriori informazioni sulle quote di IAM Access Analyzer per CloudTrail i dati, consulta le quote di IAM Access Analyzer.
- Genera policy: IAM Access Analyzer genera una policy basata sull'attività di accesso nei tuoi eventi. CloudTrail
- Esaminare e personalizzare la policy – Dopo la generazione della policy, è possibile esaminare i servizi e le azioni utilizzati dall'entità durante l'intervallo di date specificato. È possibile personalizzare ulteriormente la policy, aggiungendo o rimuovendo autorizzazioni, specificando risorse e aggiungendo condizioni al modello di policy.
- Creare e allegare policy – È possibile salvare la policy generata creando una policy gestita. È possibile allegare la policy creata all'utente o al ruolo la cui attività è stata utilizzata per generare la policy.

Informazioni sul servizio e sul livello di azione

Quando Sistema di analisi degli accessi AWS IAM genera una policy IAM, vengono restituite informazioni che consentono di personalizzare ulteriormente la policy. Quando viene generata una policy, è possibile restituire due categorie di informazioni:

- Policy con informazioni a livello di azione: per alcuni AWS servizi, come Amazon EC2, IAM Access Analyzer è in grado di identificare le azioni presenti negli CloudTrail eventi ed elenca le azioni utilizzate nella policy che genera. Per un elenco dei servizi supportati, consulta Servizi di generazione di policy per Sistema di analisi degli accessi IAM. Per alcuni servizi, Sistema di analisi degli accessi IAM richiede l'aggiunta azioni per i servizi alla policy generata.
- Policy con informazioni sui livelli di servizio – Sistema di analisi degli accessi IAM utilizza le informazioni relative all'ultimo accesso per creare un modello di policy con tutti i servizi utilizzati di recente. Quando utilizzi il Console di gestione AWS, ti chiediamo di esaminare i servizi e aggiungere azioni per completare la policy.

Per un elenco delle azioni in ciascun servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi nel riferimento all'autorizzazione del servizio.

Da sapere sulla generazione di policy

Prima di generare una policy, esaminare i dettagli importanti riportati di seguito.

- Abilita un CloudTrail percorso: devi avere un CloudTrail percorso abilitato affinché il tuo account possa generare una politica basata sull'attività di accesso. Quando crei un CloudTrail percorso, CloudTrail invia gli eventi relativi al percorso a un bucket Amazon S3 da te specificato. Per informazioni su come creare un CloudTrail percorso, consulta Creazione di un percorso per il tuo AWS account nella Guida per l'AWS CloudTrail utente.
- Eventi dati non disponibili: Sistema di analisi degli accessi IAM non identifica l'attività a livello di azione per eventi di dati, ad esempio eventi di dati di Amazon S3, nelle policy generate.
- PassRole— L'`iam:PassRole`azione non viene tracciata CloudTrail e non è inclusa nelle politiche generate.
- Ridurre il tempo di generazione della policy – Per generare più rapidamente una policy, ridurre l'intervallo di date specificato durante la configurazione per la generazione delle policy.
- Utilizzo CloudTrail per il controllo: non utilizzare la generazione delle policy per scopi di controllo; utilizzatela invece. CloudTrail Per ulteriori informazioni sull'utilizzo CloudTrail, consulta Registrazione delle chiamate IAM e AWS STS API con. AWS CloudTrail

- Azioni negate: la generazione delle policy esamina tutti CloudTrail gli eventi, comprese le azioni negate.
- Una console IAM di policy – È possibile generare una policy alla volta nella console IAM.
- Console IAM per la disponibilità di policy generate – È possibile esaminare una policy generata nella console IAM per un massimo di 7 giorni dopo la sua generazione. Dopo 7 giorni, è necessario generare una nuova policy.
- Quote di generazione di policy: per ulteriori informazioni sulle quote di generazione delle policy di Sistema di analisi degli accessi IAM, consulta Quote di Sistema di analisi degli accessi IAM.
- Si applicano le tariffe standard di Amazon S3: quando utilizzi la funzionalità di generazione delle policy, IAM Access Analyzer esamina CloudTrail i log nel bucket S3. Non sono previsti costi di storage aggiuntivi per accedere ai CloudTrail log per la generazione delle policy. AWS addebita le tariffe standard di Amazon S3 per le richieste e il trasferimento dei dati dei CloudTrail log archiviati nel bucket S3.
- AWS Control Tower supporto — La generazione di policy non supporta l'utilizzo di AWS CloudTrail percorsi AWS Control Tower creati durante la generazione delle policy, per i seguenti motivi:
  - I CloudTrail dati dell'organizzazione vengono registrati in un altro account, l'account AWS Control Tower Log Archive.
  - Le autorizzazioni per il bucket S3 in cui sono archiviati questi log non possono essere riconfigurate a causa delle restrizioni sul bucket di registrazione S3 impostate dalle policy di controllo dei servizi di AWS Control Tower.

Autorizzazioni richieste per generare una policy

Le autorizzazioni necessarie per generare una policy per la prima volta differiscono da quelle necessarie per generare una policy per usi successivi. Per ulteriori informazioni, consulta Nozioni di base su AWS Identity and Access Management Access Analyzer.

First-time configurazione

Quando si genera una policy per la prima volta, è necessario scegliere un ruolo del servizio esistente appropriato nell'account o crearne uno nuovo. Il ruolo di servizio fornisce a IAM Access Analyzer l'accesso CloudTrail e l'ultimo accesso al servizio nel tuo account. Solo gli amministratori devono disporre delle autorizzazioni necessarie per creare e configurare i ruoli. Pertanto, è consigliabile che un amministratore crei il ruolo del servizio durante la prima configurazione. Per saperne di più sulle autorizzazioni necessarie per creare ruoli di servizio, vedi Creazione di un ruolo per delegare le autorizzazioni a un servizio. AWS

Autorizzazioni richieste per il ruolo del servizio

Quando si crea un ruolo del servizio, si configurano due policy per il ruolo. Si allega una policy di autorizzazioni IAM al ruolo che specifica le operazioni che il ruolo può eseguire. È inoltre possibile allegare una policy di attendibilità del ruolo al ruolo che specifica l'entità che può utilizzare il ruolo.

La prima policy di esempio mostra la policy di autorizzazioni per il ruolo del servizio necessario per generare una policy. Nella seconda policy di esempio viene illustrata la policy di attendibilità del ruolo necessaria per il ruolo del servizio. Puoi utilizzare queste policy per aiutarti a creare un ruolo di servizio quando usi l' AWS API o AWS CLI per generare una policy. Quando si utilizza la console IAM per creare un ruolo del servizio come parte del processo di generazione della policy, vengono generate automaticamente queste policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "cloudtrail:GetTrail",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetServiceLastAccessedDetails",
                "iam:GenerateServiceLastAccessedDetails"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
```

```
    ]
}
```

La policy di esempio seguente mostra la policy di attendibilità del ruolo con le autorizzazioni che consentono a Sistema di analisi degli accessi IAM di assumere il ruolo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "access-analyzer.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Usi successivi

Per generare policy in Console di gestione AWS, un utente IAM deve disporre di una policy di autorizzazioni che gli consenta di passare il ruolo di servizio utilizzato per la generazione delle policy a IAM Access Analyzer. `iam:PassRole`è in genere accompagnata da un codice `iam:GetRole` che consente all'utente di ottenere i dettagli del ruolo da passare. In questo esempio, l'utente può passare solo i ruoli esistenti nell'account specificato con nomi che iniziano con `AccessAnalyzerMonitorServiceRole*`. Per saperne di più sul passaggio dei ruoli IAM ai AWS servizi, vedi Concedere a un utente le autorizzazioni per passare un ruolo a un AWS servizio.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Sid": "AllowUserToPassRole",
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/service-role/
AccessAnalyzerMonitorServiceRole*"
    }
  ]
}
```

È inoltre necessario disporre delle seguenti autorizzazioni IAM Access Analyzer per generare policy nell' Console di gestione AWS AWS API o AWS CLI come mostrato nella seguente dichiarazione sulla politica.

```
{
  "Sid": "AllowUserToGeneratePolicy",
  "Effect": "Allow",
  "Action": [
    "access-analyzer:CancelPolicyGeneration",
    "access-analyzer:GetGeneratedPolicy",
    "access-analyzer:ListPolicyGenerations",
    "access-analyzer:StartPolicyGeneration"
  ],
  "Resource": "*"
}
```

Per i primi utilizzi e per quelli successivi

Quando si utilizza Console di gestione AWS per generare una policy, è necessario disporre dell'`cloudtrail:ListTrails`autorizzazione per elencare i CloudTrail percorsi nel proprio account, come mostrato nella seguente dichiarazione sulla politica.

```
{
  "Sid": "AllowUserToListTrails",
  "Effect": "Allow",
  "Action": [
    "cloudtrail:ListTrails"
  ],
  "Resource": "*"
```

```
}
```

Genera una policy basata sull' CloudTrail attività (console)

È possibile generare una policy per un utente IAM o un ruolo.

Fase 1: Generazione di una policy basata sull' CloudTrail attività

Nella procedura seguente viene illustrato come generare una policy per un ruolo utilizzando il Console di gestione AWS.

Generare una policy per un ruolo IAM

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione sulla sinistra, scegliere Roles (Ruoli).

   > **Note**
   >
   > I passaggi per generare una policy basata sull'attività di un utente IAM sono quasi identici. A tale scopo, scegliere Users (Utenti) anziché Roles (Ruoli).

3. Nell'elenco dei ruoli dell'account, scegliere il nome del ruolo di cui si desidera utilizzare l'attività per generare una policy.
4. Nella scheda Autorizzazioni, nella sezione Genera policy based on CloudTrail events, scegli Genera policy.
5. Nella pagina Genera policy, specifica il periodo di tempo in cui desideri che IAM Access Analyzer analizzi i tuoi CloudTrail eventi alla ricerca delle azioni intraprese con il ruolo. Puoi scegliere un intervallo massimo di 90 giorni. Ti consigliamo di scegliere il periodo di tempo più breve possibile per ridurre il tempo di generazione della polizza.
6. Nella sezione di CloudTrail accesso, scegli un ruolo esistente adatto o crea un nuovo ruolo se non esiste un ruolo adatto. Il ruolo fornisce a IAM Access Analyzer le autorizzazioni per accedere ai tuoi CloudTrail dati per tuo conto per esaminare l'attività di accesso per identificare i servizi e le azioni che sono stati utilizzati. Consulta Autorizzazioni richieste per generare una policy per ulteriori informazioni sulle autorizzazioni necessarie per questo ruolo.
7. Nella sezione CloudTrail Trail to be analyzed (Percorso da analizzare), specificare il percorso CloudTrail che registra gli eventi per l'account.

Se scegli un CloudTrail percorso che memorizza i log in un altro account, viene visualizzata una finestra informativa sull'accesso tra account. Cross-accountl'accesso richiede una configurazione aggiuntiva. Per ulteriori informazioni, consulta Choose a role for cross-account access più avanti in questo argomento.

8. Scegliere Generate policy (Genera policy).
9. Mentre è in corso la generazione della policy, l'utente viene rimandato alla pagina Roles (Ruoli) Summary (Riepilogo) nella scheda Permissions (Autorizzazioni). Attendere che lo stato nella sezione Policy request details (Dettagli richiesta policy) mostri Success (Operazione riuscita), quindi scegliere View generated policy (Visualizza policy generata). È possibile visualizzare la policy generata per un massimo di 7 giorni. Se si genera un'altra policy, la policy esistente viene sostituita con quella nuova generata.

Passaggio 2: Esaminare le autorizzazioni e aggiungere azioni per i servizi utilizzati

Esaminare i servizi e le azioni che Sistema di analisi degli accessi IAM ha identificato come utilizzati dal ruolo. È possibile aggiungere azioni per tutti i servizi utilizzati nel modello di policy generata.

1. Leggere le seguenti sezioni:
   - Nella pagina Review permissions (Esamina le autorizzazioni), controllare l'elenco delle azioni incluse nella policy generata. Nell'elenco vengono visualizzati i servizi e le operazioni che Sistema di analisi degli accessi IAM ha identificato come utilizzati dal ruolo nell'intervallo di date specificato.
   - La sezione Services used (Servizi utilizzati) mostra i servizi aggiuntivi che Sistema di analisi degli accessi IAM ha identificato come utilizzati dal ruolo nell'intervallo di date specificato. Le informazioni sulle azioni utilizzate potrebbero non essere disponibili per i servizi elencati in questa sezione. Utilizzare i menu per ciascun servizio elencato per scegliere manualmente le azioni che si desidera includere nella policy.
2. Dopo avere terminato di aggiungere le azioni, scegliere Next (Avanti).

Passaggio 3: Personalizzare ulteriormente la policy generata

È possibile personalizzare ulteriormente la policy aggiungendo o rimuovendo autorizzazioni o specificando risorse.

Per personalizzare la policy generata

1. Aggiornare il modello della policy. Il modello di policy contiene i segnaposto ARN delle risorse per le azioni che supportano le autorizzazioni a livello di risorsa, come mostrato nell'immagine seguente. Resource-level le autorizzazioni si riferiscono alla capacità di specificare su quali risorse gli utenti sono autorizzati a eseguire azioni. Si consiglia di utilizzare gli ARN per specificare le singole risorse nella policy per le azioni che supportano le autorizzazioni a livello di risorsa. È possibile sostituire i segnaposto ARN della risorsa con ARN della risorsa validi per il caso d'uso.

   Se un'operazione non supporta le autorizzazioni a livello di risorsa, bisogna utilizzare il carattere jolly (*) per specificare che tutte le risorse possono essere interessate dall'operazione. Per sapere quali AWS servizi supportano le autorizzazioni a livello di risorsa, consulta AWS i servizi che funzionano con IAM. Per un elenco delle operazioni in ciascun servizio e per sapere quali operazioni supportano le autorizzazioni a livello di risorsa, consultare Actions, Resources, and Condition Keys for Services AWS (Operazioni, risorse e chiavi di condizione per i servizi).



2. (Facoltativo) Aggiungere, modificare o rimuovere le istruzioni della policy JSON nel modello. Per ulteriori informazioni sulla scrittura di policy JSON, consulta Creazione di policy IAM (console).
3. Al termine della personalizzazione del modello della policy, sono disponibili le seguenti opzioni:

   - (Facoltativo) È possibile copiare la JSON nel modello da utilizzare separatamente all'esterno della pagina Generated policy (Policy generata). Ad esempio, se si desidera utilizzare la JSON

per creare una policy in un account diverso. Se la policy nel modello supera il limite di 6.144 caratteri per le policy JSON, viene suddivisa in più policy.

- Scegliere Next (Avanti) per riesaminare e creare una policy gestita nello stesso account.

Passaggio 4: Esaminare e creare una policy gestita

Se si dispone delle autorizzazioni per creare e allegare policy IAM, è possibile creare una policy gestita dalla policy generata. È quindi possibile allegare la policy a un utente o a un ruolo nel proprio account.

Per rivedere e creare una policy

1. Nella pagina Review and create managed policy (Rivedi e crea una policy gestita) digitare i valori per Name (Nome) e Description (Descrizione) (facoltativa) per la policy che si sta creando.
2. (Facoltativo) Nella sezione Summary (Riepilogo) è possibile esaminare le autorizzazioni che verranno incluse nella policy.
3. (Facoltativo) Aggiungere metadati alla policy collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag con IAM, consulta Tagging delle risorse IAM.
4. Al termine, effettuare una delle seguenti operazioni:
   - È possibile allegare la nuova policy direttamente al ruolo utilizzato per generare la policy. Per fare ciò, nella parte inferiore della pagina, seleziona la casella di controllo accanto a Allega politica a. ***YourRoleName*** Quindi scegliere Create and attach policy (Crea e allega policy).
   - In caso contrario, selezionare Create policy (Crea policy). È possibile trovare la policy creata nell'elenco di policy nel riquadro di navigazione Policies (Policy) della console IAM.
5. È possibile allegare la policy creata a un'entità nel proprio account. Dopo aver collegato la policy, è possibile rimuovere tutte le altre policy di carattere troppo generale che potrebbero essere collegate all'entità. Per sapere come collegare una policy gestita, consulta Aggiunta di autorizzazioni di identità IAM (console).

Genera una policy utilizzando AWS CloudTrail dati in un altro account

Potresti creare CloudTrail percorsi che archiviano i dati in account centrali per semplificare le attività di governo. Ad esempio, puoi utilizzarlo AWS Organizations per creare un percorso che registri tutti gli eventi relativi Account AWS a tutti i membri dell'organizzazione. Il percorso appartiene a un account centrale.

Se desideri generare una policy per un utente o un ruolo in un account diverso dall'account in cui sono archiviati i dati di CloudTrail registro, devi concedere l'accesso a più account. Per fare ciò, hai bisogno sia di un ruolo che di una policy bucket che concedano le autorizzazioni di IAM Access Analyzer ai tuoi log. CloudTrail Per ulteriori informazioni sulla creazione di percorsi dell'organizzazione, consulta Creazione di un percorso per un'organizzazione.

In questo esempio, supponiamo di voler generare una policy per un utente o un ruolo nell'account A. Il CloudTrail percorso nell'account A memorizza CloudTrail i log in un bucket nell'account B. Prima di poter generare una policy, devi apportare i seguenti aggiornamenti:

1. Scegli un ruolo esistente o crea un nuovo ruolo di servizio che conceda a IAM Access Analyzer l'accesso al bucket nell'account B (dove sono CloudTrail archiviati i tuoi log).
2. Verifica la tua policy di proprietà degli oggetti del bucket Amazon S3 e di autorizzazioni del bucket nell'account B per consentire a Sistema di analisi degli accessi IAM di accedere agli oggetti nel bucket.

Fase 1: Scelta o creazione di un ruolo per l'accesso tra account

- Nella schermata Genera policy, l'opzione Utilizza un ruolo esistente è preselezionata se nel tuo account esiste già un ruolo con le autorizzazioni richieste. In caso contrario, scegli Crea e utilizza un nuovo ruolo di servizio. Il nuovo ruolo viene utilizzato per concedere a IAM Access Analyzer l'accesso ai tuoi CloudTrail log nell'account B.

Fase 2: verifica o aggiornamento della configurazione del bucket Amazon S3 nell'account B

1. Accedi Console di gestione AWS e apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Nell'elenco dei bucket, scegli il nome del bucket in cui sono archiviati i log dei CloudTrail percorsi.
3. Scegli la scheda Permissions (Autorizzazioni) e individua la sezione Object Ownership (Proprietà dell'oggetto).

   Utilizza le impostazioni di proprietà degli oggetti del bucket Amazon S3 per controllare la proprietà dei nuovi oggetti che vengono caricati nei tuoi bucket. Per impostazione predefinita, quando altri oggetti Account AWS caricano nel tuo bucket, l'account di caricamento possiede gli oggetti. Per generare una policy, il proprietario del bucket deve possedere tutti gli oggetti all'interno del bucket. A seconda del caso d'uso dell'ACL, potrebbe essere necessario modificare

l'impostazione Object Ownership (Proprietà dell'oggetto) del bucket. Imposta Object Ownership (Proprietà dell'oggetto) su una delle seguenti opzioni.

- Bucket owner enforced (Proprietario del bucket applicato) (opzione consigliata)
- Bucket owner preferred (Proprietario del bucket preferito)

> **Important**
>
> Per generare correttamente una policy, gli oggetti del bucket devono essere di proprietà del proprietario del bucket. Se scegli di utilizzare Bucket owner preferred (Proprietario del bucket preferito), puoi generare una policy solo per il periodo di tempo successivo alla modifica della proprietà dell'oggetto.

Per ulteriori informazioni sulla proprietà degli oggetti in Amazon S3, consulta la sezione Controllo della proprietà degli oggetti e disabilitazione delle ACL per il bucket nella Guida per l'utente di Amazon S3.

4. Aggiungi le autorizzazioni alla tua policy del bucket Amazon S3 nell'account B per consentire l'accesso al ruolo nell'account A.

   La policy di esempio seguente consente `ListBucket` e `GetObject` per il bucket denominato `amzn-s3-demo-bucket`. Consente l'accesso se il ruolo che accede al bucket appartiene a un account dell'organizzazione e ha un nome che inizia con `AccessAnalyzerMonitorServiceRole`. L'uso di `aws:PrincipalArn` come `Condition` nell'elemento `Resource` assicura che il ruolo possa accedere all'attività per l'account solo se appartiene all'account A. Puoi sostituire `amzn-s3-demo-bucket` con il nome del bucket, `optional-prefix` con un prefisso facoltativo per il bucket e `organization-id` con l'ID dell'organizzazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyGenerationBucketPolicy",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
```

```
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/optional-prefix/AWSLogs/organization-id/
${aws:PrincipalAccount}/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "organization-id"
        },
        "ArnLike": {
          "aws:PrincipalArn": "arn:aws:iam::${aws:PrincipalAccount}:role/service-
role/AccessAnalyzerMonitorServiceRole*"
        }
      }
    }
  ]
}
```

5. Se crittografi i log utilizzando AWS KMS, aggiorna la policy relativa alle AWS KMS chiavi nell'account in cui memorizzi i CloudTrail log per concedere a IAM Access Analyzer l'accesso all'uso della chiave, come mostrato nel seguente esempio di policy. Sostituisci `CROSS_ACCOUNT_ORG_TRAIL_FULL_ARN` con l'ARN per il tuo percorso e `organization-id` con l'ID dell'organizzazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:aws:cloudtrail:arn":
 "CROSS_ACCOUNT_ORG_TRAIL_FULL_ARN",
          "aws:PrincipalOrgID": "organization-id"
```

```
        },
        "StringLike": {
          "kms:ViaService": [
            "access-analyzer.*.amazonaws.com",
            "s3.*.amazonaws.com"
            ]
        },
        "ArnLike": {
          "aws:PrincipalArn": "arn:aws:iam::${aws:PrincipalAccount}:role/service-
role/AccessAnalyzerMonitorServiceRole*"
        }
      }
     }
    ]
}
```

Genera una policy basata sull' CloudTrail attività (AWS CLI)

È possibile utilizzare i seguenti comandi per generare una policy utilizzando AWS CLI.

Per generare una policy

- aws accessanalyzer start-policy-generation

Per visualizzare una policy generata

- aws accessanalyzer get-generated-policy

Per annullare una richiesta di generazione di policy

- aws accessanalyzer cancel-policy-generation

Per visualizzare un elenco di richieste di generazione di policy

- aws accessanalyzer list-policy-generations

Genera una politica basata sull' CloudTrail attività (AWS API)

È possibile utilizzare le seguenti operazioni per generare una policy utilizzando l' AWS API.

Per generare una policy

- StartPolicyGeneration

Per visualizzare una policy generata

- GetGeneratedPolicy

Per annullare una richiesta di generazione di policy

- CancelPolicyGeneration

Per visualizzare un elenco di richieste di generazione di policy

- ListPolicyGenerations

Servizi di generazione di policy per Sistema di analisi degli accessi IAM

La tabella seguente elenca AWS i servizi per i quali IAM Access Analyzer genera policy con informazioni a livello di azione. Per un elenco delle azioni in ogni servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi nel Service Authorization Reference.

AWS fornisce informazioni sulla generazione delle policy in formato JSON per semplificare l'automazione dei flussi di lavoro di gestione delle policy. Con le informazioni di riferimento del servizio, è possibile accedere alle informazioni sulla generazione delle policy da file leggibili Servizi AWS da computer. Per ulteriori informazioni, consulta Simplified Servizio AWS information for programmatic access nella documentazione di riferimento sull'autorizzazione dei servizi.

Servizio	Prefisso del servizio
AWS Identity and Access Management Access Analyzer	access-analyzer
Gestione dell'account AWS	account
AWS Certificate Manager	acm
Flussi di lavoro gestiti da Amazon per Apache Airflow	airflow

Servizio	Prefisso del servizio
AWS Amplify	amplify
AWS Amplify Generatore di interfacce utente	amplifyuibuilder
Amazon AppIntegrations	app-integrations
AWS AppConfig	appconfig
Amazon AppFlow	appflow
AWS Application Cost Profiler	application-cost-profiler
Informazioni approfondite sulle CloudWatch applicazioni Amazon	applicationinsights
AWS App Mesh	appmesh
WorkSpaces Applicazioni Amazon	appstream
AWS AppSync	appsync
Amazon Managed Service per Prometheus	aps
Amazon Athena	atena
AWS Audit Manager	auditmanager
AWS Auto Scaling	autoscaling
Marketplace AWS	aws-marketplace
AWS Backup	backup
AWS Batch	batch
Amazon Braket	braket
Budget AWS	budgets

Servizio	Prefisso del servizio
AWS Cloud9	cloud9
AWS CloudFormation	cloudformation
Amazon CloudFront	cloudfront
AWS CloudHSM	cloudhsm
Amazon CloudSearch	cloudsearch
AWS CloudTrail	cloudtrail
Amazon CloudWatch	cloudwatch
AWS CodeArtifact	codeartifact
AWS CodeDeploy	codedeploy
Amazon CodeGuru Profiler	codeguru-profiler
CodeGuru Revisore Amazon	codeguru-reviewer
AWS CodePipeline	codepipeline
AWS CodeStar	codestar
Notifiche AWS CodeStar	codestar-notifications
Amazon Cognito Identity	cognito-identity
Pool di utenti Amazon Cognito	cognito-idp
Amazon Cognito Sync	cognito-sync
Amazon Comprehend Medical	comprehendmedical
AWS Compute Optimizer	compute-optimizer

Servizio	Prefisso del servizio
AWS Config	config
Cliente Amazon Connect	connect
AWS Cost and Usage Report	cur
AWS Glue DataBrew	databrew
AWS Data Exchange	dataexchange
AWS Data Pipeline	datapipeline
DynamoDB Accelerator	dax
AWS Device Farm	devicefarm
Amazon DevOps Guru	devops-guru
AWS Direct Connect	directconnect
Amazon Data Lifecycle Manager	dlm
AWS Database Migration Service	dms
Cluster elastici Amazon DocumentDB	docdb-elastic
Amazon DynamoDB	dynamodb
Amazon Elastic Block Store	ebs
Amazon Elastic Compute Cloud	ec2
Amazon Elastic Container Registry	ecr
Amazon Elastic Container Registry Public	ecr-public
Amazon Elastic Container Service	ecs
Amazon Elastic Kubernetes Service	eks

Servizio	Prefisso del servizio
Amazon ElastiCache	elasticache
AWS Elastic Beanstalk	elasticbeanstalk
Amazon Elastic File System	elasticfilesystem
Elastic Load Balancing	elasticloadbalancing
Amazon EMR su EKS (Containers EMR)	emr-containers
Amazon EMR Serverless	emr-serverless
OpenSearch Servizio Amazon	es
Amazon EventBridge	eventi
Amazon FinSpace	finspace
Amazon Data Firehose	firehose
AWS Fault Injection Service	fis
AWS Firewall Manager	fms
Amazon Fraud Detector	frauddetector
Amazon FSx	fsx
GameLift Server Amazon	gamelift
Servizio di posizione Amazon	geo
Amazon Glacier	glacier
Grafana gestito da Amazon	grafana
AWS IoT Greengrass	greengrass

Servizio	Prefisso del servizio
AWS Ground Station	groundstation
Amazon GuardDuty	guardduty
AWS HealthLake	healthlake
Amazon Honeycode	honeycode
AWS Identity and Access Management	iam
AWS Archivio di identità	identitystore
EC2 Image Builder	imagebuilder
Amazon Inspector Classic	inspector
Amazon Inspector	inspector2
AWS IoT	iot
AWS IoT Analytics	iotanalytics
AWS IoT Core Device Advisor	iotdeviceadvisor
AWS IoT Events	iotevents
AWS IoT Fleet Hub	iotfleethub
AWS IoT SiteWise	iotsitewise
AWS IoT TwinMaker	iottwinmaker
Wireless AWS IoT	iotwireless
Amazon Interactive Video Service	ivs
Amazon Interactive Video Service Chat	ivschat
Amazon Managed Streaming per Apache Kafka	kafka

Servizio	Prefisso del servizio
Amazon Managed Streaming per Kafka Connect	kafkaconnect
Amazon Kendra	kendra
Amazon Kinesis	kinesis
Amazon Kinesis Analytics V2	kinesisanalytics
AWS Key Management Service	kms
AWS Lambda	lambda
Amazon Lex V2	lex
AWS License Manager Gestore di abbonamenti Linux	license-manager-linux-subscriptions
Amazon Lightsail	lightsail
CloudWatch Registri Amazon	log
Amazon Lookout per le apparecchiature	lookoutequipment
Amazon Lookout per le metriche	lookoutmetrics
Modernizzazione del mainframe AWS	m2
Blockchain gestita da Amazon	managedblockchain
AWS Elemental MediaConnect	mediaconnect
AWS Elemental MediaConvert	mediaconvert
AWS Elemental MediaLive	medialive
AWS Elemental MediaTailor	mediatailor
Amazon MemoryDB	memorydb

Servizio	Prefisso del servizio
AWS Transform MGN	mgn
AWS Migration Hub	mgh
Suggerimenti AWS sulla strategia di Migration Hub	migration hub-strategy
Amazon Pinpoint	mobiletargeting
Amazon MQ	mq
AWS Network Manager	networkmanager
Amazon Nimble Studio	nimble
AWS HealthOmics	omics
AWS OpsWorks	opsworks
AWS OpsWorks CM	opsworks-cm
AWS Outposts	outposts
AWS Organizations	organizations
AWS Approfondimenti sulle prestazioni	pi
EventBridge Tubi Amazon	pipes
Amazon Polly	polly
Profili dei clienti Amazon Connect	profile
Amazon QLDB	qldb
AWS Resource Access Manager	ram
AWS Cestino di riciclaggio	rbin

Servizio	Prefisso del servizio
Amazon Relational Database Service	rds
Amazon Redshift	redshift
API dati di Amazon Redshift	redshift-data
AWS Migration Hub Refactor Spaces	refactor-spaces
Amazon Rekognition	rekognition
AWS Resilience Hub	resiliencehub
Esploratore di risorse AWS	resource-explorer-2
AWS Resource Groups	resource-groups
AWS RoboMaker	robomaker
AWS Identity and Access Management Ruoli ovunque	rolesanywhere
Amazon Route 53	route53
Controlli di ripristino Amazon Route 53	route53-recovery-control-config
Preparazione al ripristino di Amazon Route 53	route53-recovery-readiness
Amazon Route 53 Resolver	route53resolver
AWS CloudWatch RUM	rum
Amazon Simple Storage Service	s3
Amazon S3 su Outposts	s3-outposts

Servizio	Prefisso del servizio
Funzionalità geospaziali di Amazon SageMaker AI	sagemaker-geospatial
Savings Plans	savingsplans
EventBridge Schemi Amazon	schemas
Amazon SimpleDB	sdb
AWS Secrets Manager	secretsmanager
AWS Security Hub CSPM	securityhub
Amazon Security Lake	securitylake
AWS Serverless Application Repository	serverlessrepo
AWS Service Catalog	servicecatalog
AWS Cloud Map	servicediscovery
Service Quotas	servicequotas
Amazon Simple Email Service	ses
AWS Shield	shield
AWS Signer	signer
AWS SimSpace Weaver	simspaceweaver
AWS Server Migration Service	sms
Servizio di SMS e messaggi vocali Amazon Pinpoint	sms-voice
AWS Snowball Edge	snowball
Amazon Simple Queue Service	sqs

Servizio	Prefisso del servizio
AWS Systems Manager	ssm
Strumento di gestione degli incidenti AWS Systems Manager	ssm-incidents
AWS Systems Manager per SAP	ssm-sap
AWS Step Functions	states
Servizio di token di sicurezza AWS	sts
Amazon Simple Workflow Service	swf
Amazon CloudWatch Synthetics	synthetics
AWS Resource Groups Tagging API	tag
Amazon Textract	textract
Amazon Timestream	timestream
AWS Costruttore di reti di telecomunicazioni	tnb
Amazon Transcribe	transcribe
AWS Transfer Family	transfer
Amazon Translate	translate
Connect Customer Voice ID	voiceid
Amazon VPC Lattice	vpc-lattice
AWS WAFV2	wafv2
AWS Well-Architected Tool	wellarchitected
La saggezza dei clienti di Amazon Connect	wisdom
Amazon WorkLink	worklink

Servizio	Prefisso del servizio
Amazon WorkSpaces	workspace
AWS X-Ray	xray

Informazioni sulle azioni per la generazione di policy

La tabella seguente riporta le azioni per cui Sistema di analisi degli accessi IAM genera policy con informazioni a livello di operazione.

> **Important**
>
> L'operazione `iam:UpdateAccountName` sarà dichiarata obsoleta il 22 aprile 2026. Dopo il 22 aprile 2026, l'accesso all'aggiornamento del nome dell'account sarà controllato solo dall'autorizzazione `account:PutAccountName`. Ti consigliamo vivamente di aggiornare tutte le policy di controllo dei servizi che controllano gli aggiornamenti dei nomi degli account di modo che utilizzino l'autorizzazione `account:PutAccountName`.

Prefisso del servizio	Azioni
access-analyzer	analizzatore di accesso: ApplyArchiveRule
	analizzatore di accesso: CancelPolicyGeneration
	analizzatore di accesso: CheckAccessNotGranted
	analizzatore di accesso: CheckNoNewAccess
	analizzatore di accesso: CheckNoPublicAccess
	analizzatore di accesso: CreateAccessPreview
	analizzatore di accesso: CreateAnalyzer
	analizzatore di accesso: CreateArchiveRule

Prefisso del servizio	Azioni
	analizzatore di accesso: DeleteAnalyzer
	analizzatore di accesso: DeleteArchiveRule
	analizzatore di accesso: GenerateFindingRecommendation
	analizzatore di accesso: GetAccessPreview
	analizzatore di accesso: GetAnalyzedResource
	analizzatore di accesso: GetAnalyzer
	analizzatore di accesso: GetArchiveRule
	analizzatore di accesso: GetFinding
	analizzatore di accesso: GetFindingRecommendation
	analizzatore di accesso: GetFindingsStatistics
	analizzatore di accesso: GetGeneratedPolicy
	analizzatore di accesso: ListAccessPreviewFindings
	analizzatore di accesso: ListAccessPreviews
	analizzatore di accesso: ListAnalyzedResources
	analizzatore di accesso: ListAnalyzers
	analizzatore di accesso: ListArchiveRules
	analizzatore di accesso: ListFindings
	analizzatore di accesso: ListPolicyGenerations
	analizzatore di accesso: StartPolicyGeneration
	analizzatore di accesso: StartResourceScan
	analizzatore di accesso: UpdateAnalyzer

Prefisso del servizio	Azioni
	analizzatore di accesso: UpdateArchiveRule analizzatore di accesso: UpdateFindings analizzatore di accesso: ValidatePolicy
account	conto: AcceptPrimaryEmailUpdate conto: DeleteAlternateContact conto: DisableRegion conto: EnableRegion conto: GetAccountInformation conto: GetAlternateContact conto: GetContactInformation conto: GetGovCloudAccountInformation conto: GetPrimaryEmail conto: GetRegionOptStatus conto: ListRegions conto: PutAccountName conto: PutAlternateContact conto: PutContactInformation conto: StartPrimaryEmailUpdate

Prefisso del servizio	Azioni
acm	macchina fotografica: CreateAcmeEndpoint acm: DeleteCertificate acm: DescribeCertificate acm: ExportCertificate acm: GetAccountConfiguration acm: GetCertificate acm: ImportCertificate acm: ListAcmeEndpoints acm: ListCertificates acm: PutAccountConfiguration acm: RenewCertificate acm: RequestCertificate acm: ResendValidationEmail acm: SearchCertificates acm: UpdateCertificateOptions

Prefisso del servizio	Azioni
airflow	flusso d'aria: CreateCliToken flusso d'aria: CreateEnvironment flusso d'aria: CreateWebLoginToken flusso d'aria: DeleteEnvironment flusso d'aria: GetEnvironment flusso d'aria: ListEnvironments flusso d'aria: PublishMetrics flusso d'aria: UpdateEnvironment

Prefisso del servizio	Azioni
amplify	amplificare: CreateApp
	amplificare: CreateBackendEnvironment
	amplificare: CreateBranch
	amplificare: CreateDeployment
	amplificare: CreateDomainAssociation
	amplificare: CreateWebHook
	amplificare: DeleteApp
	amplificare: DeleteBackendEnvironment
	amplificare: DeleteBranch
	amplificare: DeleteDomainAssociation
	amplificare: DeleteJob
	amplificare: DeleteWebHook
	amplificare: GenerateAccessLogs
	amplificare: GetApp
	amplificare: GetArtifactUrl
	amplificare: GetBackendEnvironment
	amplificare: GetBranch
	amplificare: GetDomainAssociation
	amplificare: GetJob
	amplificare: GetWebHook
	amplificare: ListApps

Prefisso del servizio	Azioni
	amplificare: ListArtifacts amplificare: ListBackendEnvironments amplificare: ListBranches amplificare: ListDomainAssociations amplificare: ListJobs amplificare: ListWebHooks amplificare: StartDeployment amplificare: StartJob amplificare: StopJob amplificare: UpdateApp amplificare: UpdateBranch amplificare: UpdateDomainAssociation amplificare: UpdateWebHook

Prefisso del servizio	Azioni
amplifyuibuilder	amplifyuibuilder: CreateComponent amplificyuibuilder: CreateForm amplificyuibuilder: CreateTheme amplificyuibuilder: DeleteComponent amplificyuibuilder: DeleteForm amplificyuibuilder: DeleteTheme amplificyuibuilder: ExportComponents amplificyuibuilder: ExportThemes amplificyuibuilder: GetCodegenJob amplificyuibuilder: ListCodegenJobs amplificyuibuilder: ListComponents amplificyuibuilder: ListForms amplificyuibuilder: ListThemes amplificyuibuilder: ResetMetadataFlag amplificyuibuilder: StartCodegenJob amplificyuibuilder: UpdateComponent amplificyuibuilder: UpdateForm amplificyuibuilder: UpdateTheme

Prefisso del servizio	Azioni
app-integrations	integrazioni con app: CreateApplication
	integrazioni con app: CreateDataIntegration
	integrazioni con app: CreateDataIntegrationAssociation
	integrazioni con app: CreateEventIntegration
	integrazioni con app: DeleteApplication
	integrazioni con app: DeleteDataIntegration
	integrazioni con app: DeleteEventIntegration
	integrazioni con app: GetApplication
	integrazioni con app: GetDataIntegration
	integrazioni con app: GetEventIntegration
	integrazioni con app: ListApplicationAssociations
	integrazioni con app: ListApplications
	integrazioni con app: ListDataIntegrationAssociations
	integrazioni con app: ListDataIntegrations
	integrazioni con app: ListEventIntegrationAssociations
	integrazioni con app: ListEventIntegrations
	integrazioni con app: UpdateApplication
	integrazioni con app: UpdateDataIntegration
	integrazioni con app: UpdateDataIntegrationAssociation
	integrazioni con app: UpdateEventIntegration

Prefisso del servizio	Azioni
appconfig	appconfig: CreateApplication app config: CreateConfigurationProfile app config: CreateDeploymentStrategy app config: CreateEnvironment app config: CreateExtension app config: CreateExtensionAssociation app config: CreateHostedConfigurationVersion app config: DeleteApplication app config: DeleteConfigurationProfile app config: DeleteDeploymentStrategy app config: DeleteEnvironment app config: DeleteExtension app config: DeleteExtensionAssociation app config: DeleteHostedConfigurationVersion app config: GetAccountSettings app config: GetApplication app config: GetConfiguration app config: GetConfigurationProfile app config: GetDeployment app config: GetDeploymentStrategy app config: GetEnvironment

Prefisso del servizio	Azioni
	app config: GetExtension app config: GetExtensionAssociation app config: GetHostedConfigurationVersion app config: ListApplications app config: ListConfigurationProfiles app config: ListDeploymentStrategies app config: ListDeployments app config: ListEnvironments app config: ListExtensionAssociations app config: ListExtensions app config: ListHostedConfigurationVersions app config: StartDeployment app config: StopDeployment app config: UpdateAccountSettings app config: UpdateApplication app config: UpdateConfigurationProfile app config: UpdateDeploymentStrategy app config: UpdateEnvironment app config: UpdateExtension app config: UpdateExtensionAssociation app config: ValidateConfiguration

Prefisso del servizio	Azioni
appflow	flusso di app: CancelFlowExecutions
	flusso di app: CreateConnectorProfile
	flusso di app: CreateFlow
	flusso di app: DeleteConnectorProfile
	flusso di app: DeleteFlow
	flusso di app: DescribeConnector
	flusso di app: DescribeConnectorEntity
	flusso di app: DescribeConnectorProfiles
	flusso di app: DescribeConnectors
	flusso di app: DescribeFlow
	flusso di app: DescribeFlowExecutionRecords
	flusso di app: ListConnectorEntities
	flusso di app: ListConnectors
	flusso di app: ListFlows
	flusso di app: RegisterConnector
	flusso di app: ResetConnectorMetadataCache
	flusso di app: StartFlow
	flusso di app: StopFlow
	flusso di app: UnRegisterConnector
	flusso di app: UpdateConnectorProfile
	flusso di app: UpdateConnectorRegistration

Prefisso del servizio	Azioni
	flusso di app: UpdateFlow

Prefisso del servizio	Azioni
applicationinsights	approfondimenti sulle applicazioni: AddWorkload approfondimenti sulle applicazioni: CreateApplication approfondimenti sulle applicazioni: CreateComponent approfondimenti sulle applicazioni: CreateLogPattern approfondimenti sulle applicazioni: DeleteApplication approfondimenti sulle applicazioni: DeleteComponent approfondimenti sulle applicazioni: DeleteLogPattern approfondimenti sulle applicazioni: DescribeApplication approfondimenti sulle applicazioni: DescribeComponent approfondimenti sulle applicazioni: DescribeComponentConfiguration approfondimenti sulle applicazioni: DescribeComponentConfigurationRecommendation approfondimenti sulle applicazioni: DescribeLogPattern approfondimenti sulle applicazioni: DescribeObservation approfondimenti sulle applicazioni: DescribeProblem approfondimenti sulle applicazioni: DescribeProblemObservations approfondimenti sulle applicazioni: DescribeWorkload approfondimenti sulle applicazioni: ListApplications approfondimenti sulle applicazioni: ListComponents approfondimenti sulle applicazioni: ListConfigurationHistory approfondimenti sulle applicazioni: ListLogPatternSets

Prefisso del servizio	Azioni
	approfondimenti sulle applicazioni: ListLogPatterns approfondimenti sulle applicazioni: ListProblems approfondimenti sulle applicazioni: ListWorkloads approfondimenti sulle applicazioni: RemoveWorkload approfondimenti sulle applicazioni: UpdateApplication approfondimenti sulle applicazioni: UpdateComponent approfondimenti sulle applicazioni: UpdateComponentConfiguration approfondimenti sulle applicazioni: UpdateLogPattern approfondimenti sulle applicazioni: UpdateWorkload

Prefisso del servizio	Azioni
appmesh	app mesh: CreateGatewayRoute
	app mesh: CreateMesh
	app mesh: CreateRoute
	app mesh: CreateVirtualGateway
	app mesh: CreateVirtualNode
	app mesh: CreateVirtualRouter
	app mesh: CreateVirtualService
	app mesh: DeleteGatewayRoute
	app mesh: DeleteMesh
	app mesh: DeleteRoute
	app mesh: DeleteVirtualGateway
	app mesh: DeleteVirtualNode
	app mesh: DeleteVirtualRouter
	app mesh: DeleteVirtualService
	app mesh: DescribeGatewayRoute
	app mesh: DescribeMesh
	app mesh: DescribeRoute
	app mesh: DescribeVirtualGateway
	app mesh: DescribeVirtualNode
	app mesh: DescribeVirtualRouter
	app mesh: DescribeVirtualService

Prefisso del servizio	Azioni
	app mesh: ListGatewayRoutes
	app mesh: ListMeshes
	app mesh: ListRoutes
	app mesh: ListVirtualGateways
	app mesh: ListVirtualNodes
	app mesh: ListVirtualRouters
	app mesh: ListVirtualServices
	app mesh: StreamAggregatedResources
	app mesh: UpdateGatewayRoute
	app mesh: UpdateMesh
	app mesh: UpdateRoute
	app mesh: UpdateVirtualGateway
	app mesh: UpdateVirtualNode
	app mesh: UpdateVirtualRouter
	app mesh: UpdateVirtualService

Prefisso del servizio	Azioni
appstream	appstream: AssociateAppBlockBuilderAppBlock appstream: AssociateApplicationFleet appstream: AssociateApplicationToEntitlement appstream: AssociateFleet appstream: AssociateSoftwareToImageBuilder appstream: BatchAssociateUserStack appstream: BatchDisassociateUserStack appstream: CopyImage appstream: CreateAppBlock appstream: CreateAppBlockBuilder appstream: CreateAppBlockBuilderStreamingURL appstream: CreateApplication appstream: CreateDirectoryConfig appstream: CreateEntitlement appstream: CreateFleet appstream: CreateImageBuilder appstream: CreateImageBuilderStreamingURL appstream: CreateStack appstream: URL CreateStreaming appstream: CreateThemeForStack appstream: CreateUpdatedImage

Prefisso del servizio	Azioni
	appstream: CreateUsageReportSubscription appstream: CreateUser appstream: DeleteAppBlock appstream: DeleteAppBlockBuilder appstream: DeleteApplication appstream: DeleteDirectoryConfig appstream: DeleteEntitlement appstream: DeleteFleet appstream: DeleteImage appstream: DeleteImageBuilder appstream: DeleteImagePermissions appstream: DeleteStack appstream: DeleteThemeForStack appstream: DeleteUsageReportSubscription appstream: DeleteUser appstream: DescribeAppBlockBuilderAppBlockAssociations appstream: DescribeAppBlockBuilders appstream: DescribeAppBlocks appstream: DescribeAppLicenseUsage appstream: DescribeApplicationFleetAssociations appstream: DescribeApplications

Prefisso del servizio	Azioni
	appstream: DescribeDirectoryConfigs appstream: DescribeEntitlements appstream: DescribeFleets appstream: DescribeImageBuilders appstream: DescribeImagePermissions appstream: DescribeImages appstream: DescribeSessions appstream: DescribeStacks appstream: DescribeThemeForStack appstream: DescribeUsageReportSubscriptions appstream: DescribeUserStackAssociations appstream: DescribeUsers appstream: DisableUser appstream: DisassociateAppBlockBuilderAppBlock appstream: DisassociateApplicationFleet appstream: DisassociateApplicationFromEntitlement appstream: DisassociateFleet appstream: DisassociateSoftwareFromImageBuilder appstream: EnableUser appstream: ExpireSession appstream: GetExportImageTask

Prefisso del servizio	Azioni
	appstream: ListAssociatedFleets appstream: ListAssociatedStacks appstream: ListEntitledApplications appstream: ListExportImageTasks appstream: StartAppBlockBuilder appstream: StartFleet appstream: StartImageBuilder appstream: StartSoftwareDeploymentToImageBuilder appstream: StopAppBlockBuilder appstream: StopFleet appstream: StopImageBuilder appstream: UpdateAppBlockBuilder appstream: UpdateApplication appstream: UpdateDirectoryConfig appstream: UpdateEntitlement appstream: UpdateFleet appstream: UpdateImagePermissions appstream: UpdateStack appstream: UpdateThemeForStack

Prefisso del servizio	Azioni
appsync	sincronizzazione delle app: AssociateApi
	sincronizzazione delle app: AssociateMergedGraphqlApi
	sincronizzazione delle app: AssociateSourceGraphqlApi
	sincronizzazione delle app: CreateApi
	sincronizzazione delle app: CreateApiCache
	sincronizzazione delle app: CreateApiKey
	sincronizzazione delle app: CreateChannelNamespace
	sincronizzazione delle app: CreateDataSource
	sincronizzazione delle app: CreateDomainName
	sincronizzazione delle app: CreateFunction
	sincronizzazione delle app: CreateGraphqlApi
	sincronizzazione delle app: CreateResolver
	sincronizzazione delle app: CreateType
	sincronizzazione delle app: DeleteApi
	sincronizzazione delle app: DeleteApiCache
	sincronizzazione delle app: DeleteApiKey
	sincronizzazione delle app: DeleteChannelNamespace
	sincronizzazione delle app: DeleteDataSource
	sincronizzazione delle app: DeleteDomainName
	sincronizzazione delle app: DeleteFunction
	sincronizzazione delle app: DeleteGraphqlApi

Prefisso del servizio	Azioni
	sincronizzazione delle app: DeleteResolver
	sincronizzazione delle app: DeleteType
	sincronizzazione delle app: DisassociateApi
	sincronizzazione delle app: DisassociateMergedGraphqlApi
	sincronizzazione delle app: DisassociateSourceGraphqlApi
	sincronizzazione delle app: EvaluateCode
	sincronizzazione delle app: EvaluateMappingTemplate
	sincronizzazione delle app: FlushApiCache
	sincronizzazione delle app: GetApi
	sincronizzazione delle app: GetApiAssociation
	sincronizzazione delle app: GetApiCache
	sincronizzazione delle app: GetChannelNamespace
	sincronizzazione delle app: GetDataSource
	sincronizzazione delle app: GetDataSourceIntrospection
	sincronizzazione delle app: GetDomainName
	sincronizzazione delle app: GetFunction
	sincronizzazione delle app: GetGraphqlApi
	sincronizzazione delle app: GetGraphqlApiEnvironmentVariables
	sincronizzazione delle app: GetIntrospectionSchema
	sincronizzazione delle app: GetResolver
	sincronizzazione delle app: GetSchemaCreationStatus

Prefisso del servizio	Azioni
	sincronizzazione delle app: GetSourceApiAssociation
	sincronizzazione delle app: GetType
	sincronizzazione delle app: ListApiKeys
	sincronizzazione delle app: ListApis
	sincronizzazione delle app: ListChannelNamespaces
	sincronizzazione delle app: ListDataSources
	sincronizzazione delle app: ListDomainNames
	sincronizzazione delle app: ListFunctions
	sincronizzazione delle app: ListGraphqlApis
	sincronizzazione delle app: ListResolvers
	sincronizzazione delle app: ListResolversByFunction
	sincronizzazione app: ListSourceApiAssociations
	sincronizzazione app: ListTypes
	sincronizzazione app: ListTypesByAssociation
	sincronizzazione app: PutGraphqlApiEnvironmentVariables
	sincronizzazione app: StartDataSourceIntrospection
	sincronizzazione app: StartSchemaCreation
	sincronizzazione app: StartSchemaMerge
	sincronizzazione app: UpdateApi
	sincronizzazione app: UpdateApiCache
	sincronizzazione app: UpdateApiKey

Prefisso del servizio	Azioni
	sincronizzazione app: UpdateChannelNamespace
	sincronizzazione app: UpdateDataSource
	sincronizzazione app: UpdateDomainName
	sincronizzazione app: UpdateFunction
	sincronizzazione app: UpdateGraphqlApi
	sincronizzazione app: UpdateResolver
	sincronizzazione app: UpdateSourceApiAssociation
	sincronizzazione app: UpdateType

Prefisso del servizio	Azioni
aps	app: CreateAlertManagerDefinition rubinetti: CreateAnomalyDetector rubinetti: CreateLoggingConfiguration rubinetti: CreateQueryLoggingConfiguration rubinetti: CreateRuleGroupsNamespace rubinetti: CreateWorkspace rubinetti: DeleteAlertManagerDefinition rubinetti: DeleteAnomalyDetector rubinetti: DeleteLoggingConfiguration rubinetti: DeleteQueryLoggingConfiguration rubinetti: DeleteResourcePolicy rubinetti: DeleteRuleGroupsNamespace rubinetti: DeleteScraper rubinetti: DeleteScraperLoggingConfiguration rubinetti: DeleteWorkspace rubinetti: DescribeAlertManagerDefinition rubinetti: DescribeAnomalyDetector rubinetti: DescribeLoggingConfiguration rubinetti: DescribeQueryLoggingConfiguration rubinetti: DescribeResourcePolicy rubinetti: DescribeRuleGroupsNamespace

Prefisso del servizio	Azioni
	rubinetti: DescribeScraper rubinetti: DescribeScraperLoggingConfiguration rubinetti: DescribeWorkspace rubinetti: DescribeWorkspaceConfiguration rubinetti: GetDefaultScraperConfiguration rubinetti: ListAnomalyDetectors rubinetti: ListRuleGroupsNamespaces rubinetti: ListScrapers rubinetti: ListWorkspaces rubinetti: PutAlertManagerDefinition rubinetti: PutAnomalyDetector rubinetti: PutResourcePolicy rubinetti: PutRuleGroupsNamespace rubinetti: UpdateLoggingConfiguration rubinetti: UpdateQueryLoggingConfiguration rubinetti: UpdateScraper rubinetti: UpdateScraperLoggingConfiguration rubinetti: UpdateWorkspaceAlias rubinetti: UpdateWorkspaceConfiguration

Prefisso del servizio	Azioni
atena	atena: BatchGetNamedQuery
	atena: BatchGetPreparedStatement
	atena: BatchGetQueryExecution
	atena: CancelCapacityReservation
	atena: CreateCapacityReservation
	atena: CreateDataCatalog
	atena: CreateNamedQuery
	atena: CreateNotebook
	atena: CreatePreparedStatement
	atena: CreatePresignedNotebookUrl
	atena: CreateWorkGroup
	atena: DeleteCapacityReservation
	atena: DeleteDataCatalog
	atena: DeleteNamedQuery
	atena: DeleteNotebook
	atena: DeletePreparedStatement
	atena: DeleteWorkGroup
	atena: ExportNotebook
	atena: GetCalculationExecution
	atena: GetCalculationExecutionCode
	atena: GetCalculationExecutionStatus

Prefisso del servizio	Azioni
	atena: GetCapacityAssignmentConfiguration
	atena: GetCapacityReservation
	atena: GetDataCatalog
	atena: GetDatabase
	atena: GetNamedQuery
	atena: GetNotebookMetadata
	atena: GetPreparedStatement
	atena: GetQueryExecution
	atena: GetQueryResults
	atena: GetQueryResultsStream
	atena: GetQueryRuntimeStatistics
	atena: GetResourceDashboard
	atena: GetSession
	atena: GetSessionEndpoint
	atena: GetSessionStatus
	atena: GetTableMetadata
	atena: GetWorkGroup
	atena: ImportNotebook
	atena: ListApplicationDPUSizes
	atena: ListCalculationExecutions
	atena: ListCapacityReservations

Prefisso del servizio	Azioni
	atena: ListDataCatalogs
	atena: ListDatabases
	atena: ListEngineVersions
	atena: ListExecutors
	atena: ListNamedQueries
	atena: ListNotebookMetadata
	atena: ListNotebookSessions
	atena: ListPreparedStatements
	atena: ListQueryExecutions
	atena: ListSessions
	atena: ListTableMetadata
	atena: ListWorkGroups
	atena: PutCapacityAssignmentConfiguration
	atena: StartCalculationExecution
	atena: StartQueryExecution
	atena: StartSession
	atena: StopCalculationExecution
	atena: StopQueryExecution
	atena: TerminateSession
	atena: UpdateCapacityReservation
	atena: UpdateDataCatalog

Prefisso del servizio	Azioni
	atena: UpdateNamedQuery atena: UpdateNotebook atena: UpdateNotebookMetadata atena: UpdatePreparedStatement atena: UpdateWorkGroup

Prefisso del servizio	Azioni
auditmanager	responsabile dell'audit: AssociateAssessmentReportEvidenceFolder responsabile dell'audit: BatchAssociateAssessmentReportEvidence responsabile dell'audit: BatchCreateDelegationByAssessment responsabile dell'audit: BatchDeleteDelegationByAssessment responsabile dell'audit: BatchDisassociateAssessmentReportEvidence responsabile dell'audit: BatchImportEvidenceToAssessmentControl responsabile dell'audit: CreateAssessment responsabile dell'audit: CreateAssessmentFramework responsabile dell'audit: CreateAssessmentReport responsabile dell'audit: CreateControl responsabile dell'audit: DeleteAssessment responsabile dell'audit: DeleteAssessmentFramework responsabile dell'audit: DeleteAssessmentFrameworkShare responsabile dell'audit: DeleteAssessmentReport responsabile dell'audit: DeleteControl responsabile dell'audit: DeregisterAccount responsabile dell'audit: DeregisterOrganizationAdminAccount responsabile dell'audit: DisassociateAssessmentReportEvidenceFolder responsabile dell'audit: GetAccountStatus responsabile dell'audit: GetAssessment

Prefisso del servizio	Azioni
	responsabile dell'audit: GetAssessmentFramework responsabile dell'audit: GetAssessmentReportUrl responsabile dell'audit: GetChangeLogs responsabile dell'audit: GetControl responsabile dell'audit: GetDelegations responsabile dell'audit: GetEvidence responsabile dell'audit: GetEvidenceByEvidenceFolder responsabile dell'audit: GetEvidenceFileUploadUrl responsabile dell'audit: GetEvidenceFolder responsabile dell'audit: GetEvidenceFoldersByAssessment responsabile dell'audit: GetEvidenceFoldersByAssessmentControl responsabile dell'audit: GetInsights responsabile dell'audit: GetInsightsByAssessment responsabile dell'audit: GetOrganizationAdminAccount responsabile dell'audit: GetServicesInScope responsabile dell'audit: GetSettings responsabile dell'audit: ListAssessmentControlInsightsByControlDomain responsabile dell'audit: ListAssessmentFrameworkShareRequests responsabile dell'audit: ListAssessmentFrameworks responsabile dell'audit: ListAssessmentReports

Prefisso del servizio	Azioni
	responsabile dell'audit: ListAssessments responsabile dell'audit: ListControlDomainInsights responsabile dell'audit: ListControlDomainInsightsByAssessment responsabile dell'audit: ListControlInsightsByControlDomain responsabile dell'audit: ListControls responsabile dell'audit: ListKeywordsForDataSource responsabile dell'audit: ListNotifications responsabile dell'audit: RegisterAccount responsabile dell'audit: RegisterOrganizationAdminAccount responsabile dell'audit: StartAssessmentFrameworkShare responsabile dell'audit: UpdateAssessment responsabile dell'audit: UpdateAssessmentControl responsabile dell'audit: UpdateAssessmentControlSetStatus responsabile dell'audit: UpdateAssessmentFramework responsabile dell'audit: UpdateAssessmentFrameworkShare responsabile dell'audit: UpdateAssessmentStatus responsabile dell'audit: UpdateControl responsabile dell'audit: UpdateSettings responsabile dell'audit: ValidateAssessmentReportIntegrity

Prefisso del servizio	Azioni
scalabilità automatica	scalabilità automatica: AttachInstances
	scalabilità automatica: AttachLoadBalancerTargetGroups
	scalabilità automatica: AttachLoadBalancers
	scalabilità automatica: AttachTrafficSources
	scalabilità automatica: BatchDeleteScheduledAction
	scalabilità automatica: BatchPutScheduledUpdateGroupAction
	scalabilità automatica: CancelInstanceRefresh
	scalabilità automatica: CompleteLifecycleAction
	scalabilità automatica: CreateAutoScalingGroup
	scalabilità automatica: CreateLaunchConfiguration
	scalabilità automatica: DeleteAutoScalingGroup
	scalabilità automatica: DeleteLaunchConfiguration
	scalabilità automatica: DeleteLifecycleHook
	scalabilità automatica: DeleteNotificationConfiguration
	scalabilità automatica: DeletePolicy
	scalabilità automatica: DeleteScheduledAction
	scalabilità automatica: DeleteWarmPool
	scalabilità automatica: DescribeAccountLimits
	scalabilità automatica: DescribeAdjustmentTypes
	scalabilità automatica: DescribeAutoScalingGroups
	scalabilità automatica: DescribeAutoScalingInstances

Prefisso del servizio	Azioni
	scalabilità automatica: DescribeAutoScalingNotificationTypes
	scalabilità automatica: DescribeInstanceRefreshes
	scalabilità automatica: DescribeLaunchConfigurations
	scalabilità automatica: DescribeLifecycleHookTypes
	scalabilità automatica: DescribeLifecycleHooks
	scalabilità automatica: DescribeLoadBalancerTargetGroups
	scalabilità automatica: DescribeLoadBalancers
	scalabilità automatica: DescribeMetricCollectionTypes
	scalabilità automatica: DescribeNotificationConfigurations
	scalabilità automatica: DescribePolicies
	scalabilità automatica: DescribeScalingActivities
	scalabilità automatica: DescribeScalingProcessTypes
	scalabilità automatica: DescribeScheduledActions
	scalabilità automatica: DescribeTerminationPolicyTypes
	scalabilità automatica: DescribeTrafficSources
	scalabilità automatica: DescribeWarmPool
	scalabilità automatica: DetachInstances
	scalabilità automatica: DetachLoadBalancerTargetGroups
	scalabilità automatica: DetachLoadBalancers
	scalabilità automatica: DetachTrafficSources
	scalabilità automatica: DisableMetricsCollection

Prefisso del servizio	Azioni
	scalabilità automatica: EnableMetricsCollection
	scalabilità automatica: EnterStandby
	scalabilità automatica: ExecutePolicy
	scalabilità automatica: ExitStandby
	scalabilità automatica: GetPredictiveScalingForecast
	scalabilità automatica: PutLifecycleHook
	scalabilità automatica: PutNotificationConfiguration
	scalabilità automatica: PutScalingPolicy
	scalabilità automatica: PutScheduledUpdateGroupAction
	scalabilità automatica: PutWarmPool
	scalabilità automatica: RecordLifecycleActionHeartbeat
	scalabilità automatica: ResumeProcesses
	scalabilità automatica: RollbackInstanceRefresh
	scalabilità automatica: SetDesiredCapacity
	scalabilità automatica: SetInstanceHealth
	scalabilità automatica: SetInstanceProtection
	scalabilità automatica: StartInstanceRefresh
	scalabilità automatica: SuspendProcesses
	scalabilità automatica: TerminateInstanceInAutoScalingGroup
	scalabilità automatica: UpdateAutoScalingGroup
aws-marketplace	aws-marketplace: GetEntitlements

Prefisso del servizio	Azioni
backup	backup: CancelLegalHold
	backup: CreateBackupPlan
	backup: CreateBackupSelection
	backup: CreateBackupVault
	backup: CreateFramework
	backup: CreateLegalHold
	backup: CreateLogicallyAirGappedBackupVault
	backup: CreateReportPlan
	backup: CreateRestoreAccessBackupVault
	backup: CreateRestoreTestingPlan
	backup: CreateRestoreTestingSelection
	backup: CreateTieringConfiguration
	backup: DeleteBackupPlan
	backup: DeleteBackupSelection
	backup: DeleteBackupVault
	backup: DeleteBackupVaultAccessPolicy
	backup: DeleteBackupVaultLockConfiguration
	backup: DeleteBackupVaultNotifications
	backup: DeleteFramework
	backup: DeleteRecoveryPoint
	backup: DeleteReportPlan

Prefisso del servizio	Azioni
	backup: DeleteRestoreTestingPlan
	backup: DeleteRestoreTestingSelection
	backup: DeleteTieringConfiguration
	backup: DescribeBackupJob
	backup: DescribeBackupVault
	backup: DescribeCopyJob
	backup: DescribeFramework
	backup: DescribeGlobalSettings
	backup: DescribeProtectedResource
	backup: DescribeRecoveryPoint
	backup: DescribeRegionSettings
	backup: DescribeReportJob
	backup: DescribeReportPlan
	backup: DescribeRestoreJob
	backup: DescribeScanJob
	backup: DisassociateRecoveryPoint
	backup: DisassociateRecoveryPointFromParent
	backup: ExportBackupPlanTemplate
	backup: GetBackupPlan
	backup: GetBackupPlanFromJSON
	backup: GetBackupPlanFromTemplate

Prefisso del servizio	Azioni
	backup: GetBackupSelection
	backup: GetBackupVaultAccessPolicy
	backup: GetBackupVaultNotifications
	backup: GetLegalHold
	backup: GetRecoveryPointRestoreMetadata
	backup: GetRestoreJobMetadata
	backup: GetRestoreTestingInferredMetadata
	backup: GetRestoreTestingPlan
	backup: GetRestoreTestingSelection
	backup: GetSupportedResourceTypes
	backup: GetTieringConfiguration
	backup: ListBackupJobSummaries
	backup: ListBackupJobs
	backup: ListBackupPlanTemplates
	backup: ListBackupPlanVersions
	backup: ListBackupPlans
	backup: ListBackupSelections
	backup: ListBackupVaults
	backup: ListCopyJobSummaries
	backup: ListCopyJobs
	backup: ListFrameworks

Prefisso del servizio	Azioni
	backup: ListIndexedRecoveryPoints
	backup: ListLegalHolds
	backup: ListProtectedResources
	backup: ListRecoveryPointsByBackupVault
	backup: ListRecoveryPointsByLegalHold
	backup: ListRecoveryPointsByResource
	backup: ListReportJobs
	backup: ListReportPlans
	backup: ListRestoreAccessBackupVaults
	backup: ListRestoreJobSummaries
	backup: ListRestoreJobs
	backup: ListRestoreJobsByProtectedResource
	backup: ListRestoreTestingPlans
	backup: ListRestoreTestingSelections
	backup: ListScanJobSummaries
	backup: ListScanJobs
	backup: ListTieringConfigurations
	backup: PutBackupVaultAccessPolicy
	backup: PutBackupVaultLockConfiguration
	backup: PutBackupVaultNotifications
	backup: PutRestoreValidationResult

Prefisso del servizio	Azioni
	backup: StartBackupJob
	backup: StartCopyJob
	backup: StartReportJob
	backup: StartRestoreJob
	backup: StopBackupJob
	backup: UpdateBackupPlan
	backup: UpdateFramework
	backup: UpdateGlobalSettings
	backup: UpdateRecoveryPointLifecycle
	backup: UpdateRegionSettings
	backup: UpdateReportPlan
	backup: UpdateRestoreTestingPlan
	backup: UpdateRestoreTestingSelection
	backup: UpdateTieringConfiguration

Prefisso del servizio	Azioni
batch	lotto: CancelJob lotto: CreateComputeEnvironment lotto: CreateConsumableResource lotto: CreateJobQueue lotto: CreateQuotaShare lotto: CreateSchedulingPolicy lotto: CreateServiceEnvironment lotto: DeleteComputeEnvironment lotto: DeleteConsumableResource lotto: DeleteJobQueue lotto: DeleteQuotaShare lotto: DeleteSchedulingPolicy lotto: DeleteServiceEnvironment lotto: DeregisterJobDefinition lotto: DescribeComputeEnvironments lotto: DescribeConsumableResource lotto: DescribeJobDefinitions lotto: DescribeJobQueues lotto: DescribeJobs lotto: DescribeQuotaShare lotto: DescribeSchedulingPolicies

Prefisso del servizio	Azioni
	lotto: DescribeServiceEnvironments lotto: DescribeServiceJob lotto: GetJobQueueSnapshot lotto: ListConsumableResources lotto: ListJobs lotto: ListJobsByConsumableResource lotto: ListQuotaShares lotto: ListSchedulingPolicies lotto: ListServiceJobs lotto: RegisterJobDefinition lotto: SubmitJob lotto: SubmitServiceJob lotto: TerminateJob lotto: TerminateServiceJob lotto: UpdateComputeEnvironment lotto: UpdateConsumableResource lotto: UpdateJobQueue lotto: UpdateQuotaShare lotto: UpdateSchedulingPolicy lotto: UpdateServiceEnvironment lotto: UpdateServiceJob

Prefisso del servizio	Azioni
braket	staffa: CancelJob staffa: CancelQuantumTask staffa: CreateJob staffa: CreateQuantumTask staffa: CreateSpendingLimit staffa: GetDevice staffa: GetJob staffa: GetQuantumTask staffa: SearchDevices staffa: SearchJobs staffa: SearchQuantumTasks staffa: SearchSpendingLimits

Prefisso del servizio	Azioni
budgets	budget: CreateBudgetAction bilanci: DeleteBudgetAction bilanci: DescribeBudgetAction bilanci: DescribeBudgetActionHistories bilanci: DescribeBudgetActionsForAccount bilanci: DescribeBudgetActionsForBudget bilanci: ExecuteBudgetAction bilanci: ModifyBudget bilanci: UpdateBudgetAction bilanci: ViewBudget
cloud9	cloud9: EC2 CreateEnvironment nuvola 9: CreateEnvironmentMembership nuvola 9: DeleteEnvironment nuvola 9: DeleteEnvironmentMembership nuvola 9: DescribeEnvironmentMemberships nuvola 9: DescribeEnvironmentStatus nuvola 9: DescribeEnvironments nuvola 9: ListEnvironments nuvola 9: UpdateEnvironment nuvola 9: UpdateEnvironmentMembership

Prefisso del servizio	Azioni
cloudformation	formazione di nuvole: BatchDescribeTypeConfigurations formazione di nuvole: CancelUpdateStack formazione di nuvole: ContinueUpdateRollback formazione di nuvole: CreateChangeSet formazione di nuvole: CreateGeneratedTemplate formazione di nuvole: CreateStack formazione di nuvole: CreateStackInstances formazione di nuvole: CreateStackSet formazione di nuvole: DeactivateType formazione di nuvole: DeleteChangeSet formazione di nuvole: DeleteGeneratedTemplate formazione di nuvole: DeleteStack formazione di nuvole: DeleteStackInstances formazione di nuvole: DeleteStackSet formazione di nuvole: DeregisterType formazione di nuvole: DescribeAccountLimits formazione di nuvole: DescribeChangeSet formazione di nuvole: DescribeChangeSetHooks formazione di nuvole: DescribeEvents formazione di nuvole: DescribeGeneratedTemplate formazione di nuvole: DescribeOrganizationsAccess

Prefisso del servizio	Azioni
	formazione di nuvole: DescribePublisher formazione di nuvole: DescribeResourceScan formazione di nuvole: DescribeStackDriftDetectionStatus formazione di nuvole: DescribeStackEvents formazione di nuvole: DescribeStackInstance formazione di nuvole: DescribeStackResource formazione di nuvole: DescribeStackResourceDrifts formazione di nuvole: DescribeStackResources formazione di nuvole: DescribeStackSet formazione di nuvole: DescribeStackSetOperation formazione di nuvole: DescribeStacks formazione di nuvole: DescribeType formazione di nuvole: DescribeTypeRegistration formazione di nuvole: DetectStackDrift formazione di nuvole: DetectStackResourceDrift formazione di nuvole: DetectStackSetDrift formazione di nuvole: EstimateTemplateCost formazione di nuvole: ExecuteChangeSet formazione di nuvole: GetGeneratedTemplate formazione di nuvole: GetHookResult formazione di nuvole: GetStackPolicy

Prefisso del servizio	Azioni
	formazione di nuvole: GetTemplate formazione di nuvole: GetTemplateSummary formazione di nuvole: ImportStacksToStackSet formazione di nuvole: ListChangeSets formazione di nuvole: ListExports formazione di nuvole: ListGeneratedTemplates formazione di nuvole: ListHookResults formazione di nuvole: ListImports formazione di nuvole: ListResourceScanRelatedResources formazione di nuvole: ListResourceScanResources formazione di nuvole: ListResourceScans formazione di nuvole: ListStackInstanceResourceDrifts formazione di nuvole: ListStackInstances formazione di nuvole: ListStackRefactors formazione di nuvole: ListStackResources formazione di nuvole: ListStackSetAutoDeploymentTargets formazione di nuvole: ListStackSetOperationResults formazione di nuvole: ListStackSetOperations formazione di nuvole: ListStackSets formazione di nuvole: ListTypeRegistrations formazione di nuvole: ListTypeVersions

Prefisso del servizio	Azioni
	formazione di nuvole: ListTypes formazione di nuvole: PublishType formazione di nuvole: RecordHandlerProgress formazione di nuvole: RegisterPublisher formazione di nuvole: RegisterType formazione di nuvole: RollbackStack formazione di nuvole: SetStackPolicy formazione di nuvole: SetTypeConfiguration formazione di nuvole: SetTypeDefaultVersion formazione di nuvole: SignalResource formazione di nuvole: StartResourceScan formazione di nuvole: StopStackSetOperation formazione di nuvole: TestType formazione di nuvole: UpdateGeneratedTemplate formazione di nuvole: UpdateStack formazione di nuvole: UpdateStackInstances formazione di nuvole: UpdateStackSet formazione di nuvole: UpdateTerminationProtection formazione di nuvole: ValidateTemplate

Prefisso del servizio	Azioni
cloudfront	fronte cloud: AssociateAlias
	fronte cloud: AssociateDistributionTenantWebACL
	fronte cloud: AssociateDistributionWebACL
	fronte cloud: CreateCachePolicy
	fronte cloud: CreateCloudFrontOriginAccessIdentity
	fronte cloud: CreateConnectionFunction
	fronte cloud: CreateContinuousDeploymentPolicy
	fronte cloud: CreateDistributionTenant
	fronte cloud: CreateFieldLevelEncryptionConfig
	fronte cloud: CreateFieldLevelEncryptionProfile
	fronte cloud: CreateFunction
	fronte cloud: CreateInvalidation
	fronte cloud: CreateKeyGroup
	fronte cloud: CreateKeyValueStore
	fronte cloud: CreateMonitoringSubscription
	fronte cloud: CreateOriginAccessControl
	fronte cloud: CreateOriginRequestPolicy
	fronte cloud: CreatePublicKey
	fronte cloud: CreateRealtimeLogConfig
	fronte cloud: CreateResponseHeadersPolicy
	fronte cloud: CreateTrustStore

Prefisso del servizio	Azioni
	fronte cloud: DeleteAnycastIpList
	fronte cloud: DeleteCachePolicy
	fronte cloud: DeleteCloudFrontOriginAccessIdentity
	fronte cloud: DeleteConnectionFunction
	fronte cloud: DeleteConnectionGroup
	fronte cloud: DeleteContinuousDeploymentPolicy
	fronte cloud: DeleteDistribution
	fronte cloud: DeleteDistributionTenant
	fronte cloud: DeleteFieldLevelEncryptionConfig
	fronte cloud: DeleteFieldLevelEncryptionProfile
	fronte cloud: DeleteFunction
	fronte cloud: DeleteKeyGroup
	fronte cloud: DeleteKeyValueStore
	fronte cloud: DeleteMonitoringSubscription
	fronte cloud: DeleteOriginAccessControl
	fronte cloud: DeleteOriginRequestPolicy
	fronte cloud: DeletePublicKey
	fronte cloud: DeleteRealtimeLogConfig
	fronte cloud: DeleteResponseHeadersPolicy
	fronte cloud: DeleteStreamingDistribution
	fronte cloud: DeleteTrustStore

Prefisso del servizio	Azioni
	fronte cloud: DeleteVpcOrigin
	fronte cloud: DescribeFunction
	fronte cloud: DescribeKeyValueStore
	fronte cloud: DisassociateDistributionTenantWebACL
	fronte cloud: DisassociateDistributionWebACL
	fronte cloud: GetAnycastIpList
	fronte cloud: GetCachePolicy
	fronte cloud: GetCachePolicyConfig
	fronte cloud: GetCloudFrontOriginAccessIdentity
	fronte cloud: GetCloudFrontOriginAccessIdentityConfig
	fronte cloud: GetContinuousDeploymentPolicy
	fronte cloud: GetContinuousDeploymentPolicyConfig
	fronte cloud: GetDistributionConfig
	fronte cloud: GetFieldLevelEncryption
	fronte cloud: GetFieldLevelEncryptionConfig
	fronte cloud: GetFieldLevelEncryptionProfile
	fronte cloud: GetFieldLevelEncryptionProfileConfig
	fronte cloud: GetFunction
	fronte cloud: GetInvalidation
	fronte cloud: GetInvalidationForDistributionTenant
	fronte cloud: GetKeyGroup

Prefisso del servizio	Azioni
	fronte cloud: GetKeyGroupConfig
	fronte cloud: GetMonitoringSubscription
	fronte cloud: GetOriginAccessControl
	fronte cloud: GetOriginAccessControlConfig
	fronte cloud: GetOriginRequestPolicy
	fronte cloud: GetOriginRequestPolicyConfig
	fronte cloud: GetPublicKey
	fronte cloud: GetPublicKeyConfig
	fronte cloud: GetRealtimeLogConfig
	fronte cloud: GetResponseHeadersPolicy
	fronte cloud: GetResponseHeadersPolicyConfig
	fronte cloud: GetStreamingDistribution
	fronte cloud: GetStreamingDistributionConfig
	fronte cloud: GetVpcOrigin
	fronte cloud: ListAnycastIpLists
	fronte cloud: ListCachePolicies
	fronte cloud: ListCloudFrontOriginAccessIdentities
	fronte cloud: ListConflictingAliases
	fronte cloud: ListConnectionFunctions
	fronte cloud: ListConnectionGroups
	fronte cloud: ListContinuousDeploymentPolicies

Prefisso del servizio	Azioni
	fronte cloud: ListDistributionTenants
	fronte cloud: ListDistributionTenantsByCustomization
	fronte cloud: ListDistributions
	fronte cloud: ListDistributionsByAnycastIpListId
	fronte cloud: ListDistributionsByCachePolicyId
	fronte cloud: ListDistributionsByConnectionMode
	fronte cloud: ListDistributionsByKeyGroup
	fronte cloud: ListDistributionsByOriginRequestPolicyId
	fronte cloud: ListDistributionsByRealtimeLogConfig
	fronte cloud: ListDistributionsByResponseHeadersPolicyId
	fronte cloud: ListDistributionsByVpcOriginId
	fronte cloud: ListDistributionsByWebACLId
	fronte cloud: ListFieldLevelEncryptionConfigs
	fronte cloud: ListFieldLevelEncryptionProfiles
	fronte cloud: ListFunctions
	fronte cloud: ListInvalidations
	fronte cloud: ListInvalidationsForDistributionTenant
	fronte cloud: ListKeyGroups
	fronte cloud: ListKeyValueStores
	fronte cloud: ListOriginAccessControls
	fronte cloud: ListOriginRequestPolicies

Prefisso del servizio	Azioni
	fronte cloud: ListPublicKeys
	fronte cloud: ListRealtimeLogConfigs
	fronte cloud: ListResponseHeadersPolicies
	fronte cloud: ListStreamingDistributions
	fronte cloud: ListTrustStores
	fronte cloud: PublishConnectionFunction
	fronte cloud: PublishFunction
	fronte cloud: TestConnectionFunction
	fronte cloud: TestFunction
	fronte cloud: UpdateAnycastIpList
	fronte cloud: UpdateCachePolicy
	fronte cloud: UpdateCloudFrontOriginAccessIdentity
	fronte cloud: UpdateConnectionFunction
	fronte cloud: UpdateConnectionGroup
	fronte cloud: UpdateContinuousDeploymentPolicy
	fronte cloud: UpdateDistribution
	fronte cloud: UpdateDistributionTenant
	fronte cloud: UpdateFieldLevelEncryptionConfig
	fronte cloud: UpdateFieldLevelEncryptionProfile
	fronte cloud: UpdateFunction
	fronte cloud: UpdateKeyGroup

Prefisso del servizio	Azioni
	fronte cloud: UpdateKeyValueStore fronte cloud: UpdateOriginAccessControl fronte cloud: UpdateOriginRequestPolicy fronte cloud: UpdatePublicKey fronte cloud: UpdateRealtimeLogConfig fronte cloud: UpdateResponseHeadersPolicy fronte cloud: UpdateTrustStore
cloudhsm	cloudhsm: CreateHsm cloudhsm: DeleteBackup cloudhsm: DeleteHsm cloudhsm: DeleteResourcePolicy cloudhsm: DescribeBackups cloudhsm: DescribeClusters cloudhsm: GetResourcePolicy cloudhsm: InitializeCluster cloudhsm: ModifyBackupAttributes cloudhsm: ModifyCluster cloudhsm: PutResourcePolicy cloudhsm: RestoreBackup

Prefisso del servizio	Azioni
cloudsearch	ricerca nel cloud: BuildSuggesters
	ricerca nel cloud: CreateDomain
	ricerca nel cloud: DefineAnalysisScheme
	ricerca nel cloud: DefineExpression
	ricerca nel cloud: DefineIndexField
	ricerca nel cloud: DefineSuggester
	ricerca nel cloud: DeleteAnalysisScheme
	ricerca nel cloud: DeleteDomain
	ricerca nel cloud: DeleteExpression
	ricerca nel cloud: DeleteIndexField
	ricerca nel cloud: DeleteSuggester
	ricerca nel cloud: DescribeAnalysisSchemes
	ricerca nel cloud: DescribeAvailabilityOptions
	ricerca nel cloud: DescribeDomainEndpointOptions
	ricerca nel cloud: DescribeDomains
	ricerca nel cloud: DescribeExpressions
	ricerca nel cloud: DescribeIndexFields
	ricerca nel cloud: DescribeScalingParameters
	ricerca nel cloud: DescribeServiceAccessPolicies
	ricerca nel cloud: DescribeSuggesters
	ricerca nel cloud: IndexDocuments

Prefisso del servizio	Azioni
	ricerca nel cloud: ListDomainNames
	ricerca nel cloud: UpdateAvailabilityOptions
	ricerca nel cloud: UpdateDomainEndpointOptions
	ricerca nel cloud: UpdateScalingParameters
	ricerca nel cloud: UpdateServiceAccessPolicies

Prefisso del servizio	Azioni
cloudtrail	pista nuvolosa: CancelQuery pista nuvolosa: CreateChannel pista nuvolosa: CreateDashboard pista nuvolosa: CreateEventDataStore pista nuvolosa: CreateTrail pista nuvolosa: DeleteChannel pista nuvolosa: DeleteDashboard pista nuvolosa: DeleteEventDataStore pista nuvolosa: DeleteResourcePolicy pista nuvolosa: DeleteTrail pista nuvolosa: DeregisterOrganizationDelegatedAdmin pista nuvolosa: DescribeQuery pista nuvolosa: DescribeTrails pista nuvolosa: DisableFederation pista nuvolosa: GenerateQuery pista nuvolosa: GetChannel pista nuvolosa: GetDashboard pista nuvolosa: GetEventConfiguration pista nuvolosa: GetEventDataStore pista nuvolosa: GetEventDataStoreData pista nuvolosa: GetEventSelectors

Prefisso del servizio	Azioni
	pista nuvolosa: GetImport pista nuvolosa: GetInsightSelectors pista nuvolosa: GetResourcePolicy pista nuvolosa: GetTrail pista nuvolosa: GetTrailStatus pista nuvolosa: ListChannels pista nuvolosa: ListDashboards pista nuvolosa: ListEventDataStores pista nuvolosa: ListImportFailures pista nuvolosa: ListImports pista nuvolosa: ListInsightsData pista nuvolosa: ListPublicKeys pista nuvolosa: ListQueries pista nuvolosa: ListTrails pista nuvolosa: LookupEvents pista nuvolosa: PutEventConfiguration pista nuvolosa: PutEventSelectors pista nuvolosa: PutInsightSelectors pista nuvolosa: PutResourcePolicy pista nuvolosa: RegisterOrganizationDelegatedAdmin pista nuvolosa: RestoreEventDataStore

Prefisso del servizio	Azioni
	pista nuvolosa: SearchSampleQueries pista nuvolosa: StartEventDataStoreIngestion pista nuvolosa: StartImport pista nuvolosa: StartLogging pista nuvolosa: StartQuery pista nuvolosa: StopEventDataStoreIngestion pista nuvolosa: StopImport pista nuvolosa: StopLogging pista nuvolosa: UpdateChannel pista nuvolosa: UpdateDashboard pista nuvolosa: UpdateEventDataStore pista nuvolosa: UpdateTrail

Prefisso del servizio	Azioni
cloudwatch	orologio cloud: DeleteAlarmMuteRule
	orologio nuvoloso: DeleteAlarms
	orologio nuvoloso: DeleteAnomalyDetector
	orologio nuvoloso: DeleteDashboards
	orologio nuvoloso: DeleteInsightRules
	orologio nuvoloso: DeleteMetricStream
	orologio nuvoloso: DescribeAlarmHistory
	orologio nuvoloso: DescribeAlarms
	orologio nuvoloso: DescribeAlarmsForMetric
	orologio nuvoloso: DescribeAnomalyDetectors
	orologio nuvoloso: DescribeInsightRules
	orologio nuvoloso: DisableAlarmActions
	orologio nuvoloso: DisableInsightRules
	orologio nuvoloso: EnableAlarmActions
	orologio nuvoloso: EnableInsightRules
	orologio nuvoloso: GetAlarmMuteRule
	orologio nuvoloso: GetDashboard
	orologio nuvoloso: GetDataset
	orologio nuvoloso: GetInsightRuleReport
	orologio nuvoloso: GetMetricStatistics
	orologio nuvoloso: GetMetricStream

Prefisso del servizio	Azioni
	orologio nuvoloso: ListAlarmMuteRules
	orologio nuvoloso: ListDashboards
	orologio nuvoloso: ListManagedInsightRules
	orologio nuvoloso: ListMetricStreams
	orologio nuvoloso: PutAlarmMuteRule
	orologio nuvoloso: PutAnomalyDetector
	orologio nuvoloso: PutCompositeAlarm
	orologio nuvoloso: PutDashboard
	orologio nuvoloso: PutInsightRule
	orologio nuvoloso: PutLogAlarm
	orologio nuvoloso: PutManagedInsightRules
	orologio nuvoloso: PutMetricAlarm
	orologio nuvoloso: PutMetricStream
	orologio nuvoloso: SetAlarmState
	orologio nuvoloso: StartMetricStreams
	orologio nuvoloso: StopMetricStreams

Prefisso del servizio	Azioni
codeartifact	artefatto del codice: AssociateExternalConnection artefatto del codice: CopyPackageVersions artefatto del codice: CreateDomain artefatto del codice: CreateRepository artefatto del codice: DeleteDomain artefatto del codice: DeleteDomainPermissionsPolicy artefatto del codice: DeletePackage artefatto del codice: DeletePackageVersions artefatto del codice: DeleteRepository artefatto del codice: DeleteRepositoryPermissionsPolicy artefatto del codice: DescribeDomain artefatto del codice: DescribePackage artefatto del codice: DescribePackageVersion artefatto del codice: DescribeRepository artefatto del codice: DisassociateExternalConnection artefatto del codice: DisposePackageVersions artefatto del codice: GetAssociatedPackageGroup artefatto del codice: GetAuthorizationToken artefatto del codice: GetDomainPermissionsPolicy artefatto del codice: GetPackageVersionAsset artefatto del codice: GetPackageVersionReadme

Prefisso del servizio	Azioni
	artefatto del codice: GetRepositoryEndpoint artefatto del codice: GetRepositoryPermissionsPolicy artefatto del codice: ListDomains artefatto del codice: ListPackageGroups artefatto del codice: ListPackageVersionAssets artefatto del codice: ListPackageVersionDependencies artefatto del codice: ListPackageVersions artefatto del codice: ListPackages artefatto del codice: ListRepositories artefatto del codice: ListRepositoriesInDomain artefatto del codice: PublishPackageVersion artefatto del codice: PutDomainPermissionsPolicy artefatto del codice: PutPackageMetadata artefatto del codice: PutPackageOriginConfiguration artefatto del codice: PutRepositoryPermissionsPolicy artefatto del codice: ReadFromRepository artefatto del codice: UpdatePackageVersionsStatus artefatto del codice: UpdateRepository

Prefisso del servizio	Azioni
codedeploy	distribuzione del codice: BatchGetApplicationRevisions
	codedeploy: BatchGetApplications
	codedeploy: BatchGetDeploymentGroups
	codedeploy: BatchGetDeploymentInstances
	codedeploy: BatchGetDeploymentTargets
	codedeploy: BatchGetDeployments
	codedeploy: BatchGetOnPremisesInstances
	codedeploy: ContinueDeployment
	codedeploy: CreateApplication
	codedeploy: CreateDeployment
	codedeploy: CreateDeploymentConfig
	codedeploy: CreateDeploymentGroup
	codedeploy: DeleteApplication
	codedeploy: DeleteDeploymentConfig
	codedeploy: DeleteDeploymentGroup
	codedeploy: DeleteGitHubAccountToken
	codedeploy: DeleteResourcesByExternalId
	codedeploy: DeregisterOnPremisesInstance
	codedeploy: GetApplication
	codedeploy: GetApplicationRevision
	codedeploy: GetDeployment

Prefisso del servizio	Azioni
	codedeploy: GetDeploymentConfig
	codedeploy: GetDeploymentGroup
	codedeploy: GetDeploymentInstance
	codedeploy: GetDeploymentTarget
	codedeploy: GetOnPremisesInstance
	codedeploy: ListApplicationRevisions
	codedeploy: ListApplications
	codedeploy: ListDeploymentConfigs
	codedeploy: ListDeploymentGroups
	codedeploy: ListDeploymentInstances
	codedeploy: ListDeploymentTargets
	codedeploy: ListDeployments
	codedeploy: ListGitHubAccountTokenNames
	codedeploy: ListOnPremisesInstances
	codedeploy: PutLifecycleEventHookExecutionStatus
	codedeploy: RegisterApplicationRevision
	codedeploy: RegisterOnPremisesInstance
	codedeploy: SkipWaitTimeForInstanceTermination
	codedeploy: StopDeployment
	codedeploy: UpdateApplication
	codedeploy: UpdateDeploymentGroup

Prefisso del servizio	Azioni
codeguru-profiler	profilo codeguru: AddNotificationChannels codeguru profiler: BatchGetFrameMetricData codeguru profiler: CreateProfilingGroup codeguru profiler: DeleteProfilingGroup codeguru profiler: DescribeProfilingGroup codeguru profiler: GetFindingsReportAccountSummary codeguru profiler: GetNotificationConfiguration codeguru profiler: GetPolicy codeguru profiler: GetProfile codeguru profiler: GetRecommendations codeguru profiler: ListFindingsReports codeguru profiler: ListProfileTimes codeguru profiler: ListProfilingGroups codeguru profiler: PutPermission codeguru profiler: RemoveNotificationChannel codeguru profiler: RemovePermission codeguru profiler: SubmitFeedback codeguru profiler: UpdateProfilingGroup

Prefisso del servizio	Azioni
codeguru-reviewer	revisore di codeguru: AssociateRepository
	revisore di codeguru: CreateCodeReview
	revisore di codeguru: DescribeCodeReview
	revisore di codeguru: DescribeRecommendationFeedback
	revisore di codeguru: DescribeRepositoryAssociation
	revisore di codeguru: DisassociateRepository
	revisore di codeguru: ListCodeReviews
	revisore di codeguru: ListRecommendationFeedback
	revisore di codeguru: ListRecommendations
	revisore di codeguru: ListRepositoryAssociations
	revisore di codeguru: PutRecommendationFeedback

Prefisso del servizio	Azioni
codepipeline	pipeline di codici: AcknowledgeJob pipeline di codice: AcknowledgeThirdPartyJob pipeline di codice: CreateCustomActionType pipeline di codice: CreatePipeline pipeline di codice: DeleteCustomActionType pipeline di codice: DeletePipeline pipeline di codice: DeleteWebhook pipeline di codice: DeregisterWebhookWithThirdParty pipeline di codice: GetActionType pipeline di codice: GetJobDetails pipeline di codice: GetPipeline pipeline di codice: GetPipelineExecution pipeline di codice: GetPipelineState pipeline di codice: GetThirdPartyJobDetails pipeline di codice: ListActionExecutions pipeline di codice: ListActionTypes pipeline di codice: ListPipelineExecutions pipeline di codice: ListPipelines pipeline di codice: ListRuleExecutions pipeline di codice: ListRuleTypes pipeline di codice: ListWebhooks

Prefisso del servizio	Azioni
	pipeline di codice: OverrideStageCondition pipeline di codice: PollForJobs pipeline di codice: PollForThirdPartyJobs pipeline di codice: PutActionRevision pipeline di codice: PutApprovalResult pipeline di codice: PutJobFailureResult pipeline di codice: PutJobSuccessResult pipeline di codice: PutThirdPartyJobFailureResult pipeline di codice: PutThirdPartyJobSuccessResult pipeline di codice: PutWebhook pipeline di codice: RegisterWebhookWithThirdParty pipeline di codice: RollbackStage pipeline di codice: StartPipelineExecution pipeline di codice: StopPipelineExecution pipeline di codice: UpdateActionType pipeline di codice: UpdatePipeline

Prefisso del servizio	Azioni
codestar	codestar: AssociateTeamMember codestar: CreateProject codestar: CreateUserProfile codestar: DeleteProject codestar: DeleteUserProfile codestar: DescribeProject codestar: DescribeUserProfile codestar: DisassociateTeamMember codestar: ListProjects codestar: ListResources codestar: ListTeamMembers codestar: ListUserProfiles codestar: UpdateProject codestar: UpdateTeamMember codestar: UpdateUserProfile

Prefisso del servizio	Azioni
codestar-notifications	notifiche codestar: CreateNotificationRule notifiche codestar: DeleteNotificationRule notifiche codestar: DeleteTarget notifiche codestar: DescribeNotificationRule notifiche codestar: ListEventTypes notifiche codestar: ListNotificationRules notifiche codestar: ListTargets codestar-notifications:Subscribe codestar-notifications:Unsubscribe notifiche codestar: UpdateNotificationRule

Prefisso del servizio	Azioni
cognito-identity	identità cognitiva: CreateIdentityPool
	identità cognitiva: DeleteIdentities
	identità cognitiva: DeleteIdentityPool
	identità cognitiva: DescribeIdentity
	identità cognitiva: DescribeIdentityPool
	identità cognitiva: GetIdentityPoolRoles
	identità cognitiva: ListIdentities
	identità cognitiva: ListIdentityPools
	identità cognitiva: LookupDeveloperIdentity
	identità cognitiva: MergeDeveloperIdentities
	identità cognitiva: SetIdentityPoolRoles
	identità cognitiva: UnlinkDeveloperIdentity
	identità cognitiva: UpdateIdentityPool

Prefisso del servizio	Azioni
cognito-idp	cognito-idp: AddCustomAttributes cognito-idp: AddUserPoolClientSecret cognito-idp: AdminAddUserToGroup cognito-idp: AdminConfirmSignUp cognito-idp: AdminCreateUser cognito-idp: AdminDeleteUser cognito-idp: AdminDeleteUserAttributes cognito-idp: AdminDisableProviderForUser cognito-idp: AdminDisableUser cognito-idp: AdminEnableUser cognito-idp: AdminForgetDevice cognito-idp: AdminGetDevice cognito-idp: AdminGetUser cognito-idp: AdminGetUserAuthFactors cognito-idp: AdminInitiateAuth cognito-idp: AdminLinkProviderForUser cognito-idp: AdminListDevices cognito-idp: AdminListGroupsForUser cognito-idp: AdminListUserAuthEvents cognito-idp: AdminRemoveUserFromGroup cognito-idp: AdminResetUserPassword

Prefisso del servizio	Azioni
	cognito-idp: AdminRespondToAuthChallenge cognito-idp: AdminSetUserMFAPreference cognito-idp: AdminSetUserPassword cognito-idp: AdminSetUserSettings cognito-idp: AdminUpdateAuthEventFeedback cognito-idp: AdminUpdateDeviceStatus cognito-idp: AdminUpdateUserAttributes cognito-idp: AdminUserGlobalSignOut cognito-idp: AssociateSoftwareToken cognito-idp: ChangePassword cognito-idp: ConfirmDevice cognito-idp: ConfirmForgotPassword cognito-idp: ConfirmSignUp cognito-idp: CreateGroup cognito-idp: CreateIdentityProvider cognito-idp: CreateManagedLoginBranding cognito-idp: CreateResourceServer cognito-idp: CreateTerms cognito-idp: CreateUserImportJob cognito-idp: CreateUserPool cognito-idp: CreateUserPoolClient

Prefisso del servizio	Azioni
	cognito-idp: CreateUserPoolDomain cognito-idp: CreateUserPoolReplica cognito-idp: DeleteGroup cognito-idp: DeleteIdentityProvider cognito-idp: DeleteManagedLoginBranding cognito-idp: DeleteResourceServer cognito-idp: DeleteTerms cognito-idp: DeleteUser cognito-idp: DeleteUserAttributes cognito-idp: DeleteUserPool cognito-idp: DeleteUserPoolClient cognito-idp: DeleteUserPoolClientSecret cognito-idp: DeleteUserPoolDomain cognito-idp: DeleteUserPoolReplica cognito-idp: DescribeIdentityProvider cognito-idp: DescribeManagedLoginBranding cognito-idp: DescribeManagedLoginBrandingByClient cognito-idp: DescribeResourceServer cognito-idp: DescribeRiskConfiguration cognito-idp: DescribeTerms cognito-idp: DescribeUserImportJob

Prefisso del servizio	Azioni
	cognito-idp: DescribeUserPool cognito-idp: DescribeUserPoolClient cognito-idp: DescribeUserPoolDomain cognito-idp: ForgetDevice cognito-idp: ForgotPassword cognito-idp:GetCSVHeader cognito-idp: GetDevice cognito-idp: GetGroup cognito-idp: GetIdentityProviderByIdentifier cognito-idp: GetLogDeliveryConfiguration cognito-idp: GetProvisionedLimit cognito-idp: GetSigningCertificate cognito-idp:GetUICustomization cognito-idp: GetUser cognito-idp: GetUserAttributeVerificationCode cognito-idp: GetUserPoolMfaConfig cognito-idp: GlobalSignOut cognito-idp: InitiateAuth cognito-idp: ListDevices cognito-idp: ListGroups cognito-idp: ListIdentityProviders

Prefisso del servizio	Azioni
	cognito-idp: ListResourceServers cognito-idp: ListTerms cognito-idp: ListUserImportJobs cognito-idp: ListUserPoolClientSecrets cognito-idp: ListUserPoolClients cognito-idp: ListUserPoolReplicas cognito-idp: ListUserPools cognito-idp: ListUsers cognito-idp: ListUsersInGroup cognito-idp: ResendConfirmationCode cognito-idp: RespondToAuthChallenge cognito-idp: RevokeToken cognito-idp: SetLogDeliveryConfiguration cognito-idp: SetRiskConfiguration cognito-idp:SetUICustomization cognito-idp: SetUserMFAPreference cognito-idp: SetUserPoolMfaConfig cognito-idp: SetUserSettings cognito-idp: SignUp cognito-idp: StartUserImportJob cognito-idp: StopUserImportJob

Prefisso del servizio	Azioni
	cognito-idp: UpdateAuthEventFeedback cognito-idp: UpdateDeviceStatus cognito-idp: UpdateGroup cognito-idp: UpdateIdentityProvider cognito-idp: UpdateProvisionedLimit cognito-idp: UpdateResourceServer cognito-idp: UpdateTerms cognito-idp: UpdateUserAttributes cognito-idp: UpdateUserPool cognito-idp: UpdateUserPoolClient cognito-idp: UpdateUserPoolDomain cognito-idp: UpdateUserPoolReplica cognito-idp: VerifySoftwareToken cognito-idp: VerifyUserAttribute

Prefisso del servizio	Azioni
cognito-sync	sincronizzazione cognitiva: BulkPublish
	sincronizzazione cognitiva: DeleteDataset
	sincronizzazione cognitiva: DescribeDataset
	sincronizzazione cognitiva: DescribeIdentityPoolUsage
	sincronizzazione cognitiva: DescribeIdentityUsage
	sincronizzazione cognitiva: GetBulkPublishDetails
	sincronizzazione cognitiva: GetCognitoEvents
	sincronizzazione cognitiva: GetIdentityPoolConfiguration
	sincronizzazione cognitiva: ListDatasets
	sincronizzazione cognitiva: ListIdentityPoolUsage
	sincronizzazione cognitiva: ListRecords
	sincronizzazione cognitiva: RegisterDevice
	sincronizzazione cognitiva: SetCognitoEvents
	sincronizzazione cognitiva: SetIdentityPoolConfiguration
	sincronizzazione cognitiva: SubscribeToDataset
	sincronizzazione cognitiva: UnsubscribeFromDataset
	sincronizzazione cognitiva: UpdateRecords

Prefisso del servizio	Azioni
comprehendmedical	comprende aspetti medici: DescribeEntitiesDetectionV2Job comprendere la medicina: DescribeICD10CMInferenceJob comprendere la medicina: DescribePHIDetectionJob comprendere la medicina: DescribeRxNormInferenceJob comprendere la medicina: DescribeSNOMEDCTInferenceJob comprendere la medicina: V2 DetectEntities comprehendmedical:DetectPHI comprehendmedical:InferICD10CM comprendere la medicina: InferRxNorm comprehendmedical:InferSNOMEDCT comprendere la medicina: ListEntitiesDetectionV2Jobs comprendere la medicina: ListICD10CMInferenceJobs comprendere la medicina: ListPHIDetectionJobs comprendere la medicina: ListRxNormInferenceJobs comprendere la medicina: ListSNOMEDCTInferenceJobs comprendere la medicina: StartEntitiesDetectionV2Job comprendere la medicina: StartICD10CMInferenceJob comprendere la medicina: StartPHIDetectionJob comprendere la medicina: StartRxNormInferenceJob comprendere la medicina: StartSNOMEDCTInferenceJob comprendere la medicina: StopEntitiesDetectionV2Job

Prefisso del servizio	Azioni
	comprendere la medicina: StopICD10CMInferenceJob comprendere la medicina: StopPHIDetectionJob comprendere la medicina: StopRxNormInferenceJob comprendere la medicina: StopSNOMEDCTInferenceJob

Prefisso del servizio	Azioni
compute-optimizer	ottimizzatore di calcolo: DeleteRecommendationPreferences
	ottimizzatore di calcolo: DescribeRecommendationExportJobs
	ottimizzatore di calcolo: ExportAutoScalingGroupRecommendations
	ottimizzatore di calcolo: ExportEBSVolumeRecommendations
	ottimizzatore di calcolo: ExportEC2InstanceRecommendations
	ottimizzatore di calcolo: ExportECSServiceRecommendations
	ottimizzatore di calcolo: ExportIdleRecommendations
	ottimizzatore di calcolo: ExportLambdaFunctionRecommendations
	ottimizzatore di calcolo: ExportLicenseRecommendations
	ottimizzatore di calcolo: ExportRDSDatabaseRecommendations
	ottimizzatore di calcolo: GetEC2RecommendationProjectedMetrics
	ottimizzatore di calcolo: GetECSServiceRecommendationProjectedMetrics
	ottimizzatore di calcolo: GetEffectiveRecommendationPreferences
	ottimizzatore di calcolo: GetEnrollmentStatus
	ottimizzatore di calcolo: GetEnrollmentStatusesForOrganization
	ottimizzatore di calcolo: GetRDSDatabaseRecommendationProjectedMetrics
	ottimizzatore di calcolo: GetRecommendationPreferences
	ottimizzatore di calcolo: GetRecommendationSummaries
	ottimizzatore di calcolo: PutRecommendationPreferences

Prefisso del servizio	Azioni
	ottimizzatore di calcolo: UpdateEnrollmentStatus

Prefisso del servizio	Azioni
config	configurazione: BatchGetResourceConfig configurazione: DeleteAggregationAuthorization configurazione: DeleteConfigRule configurazione: DeleteConfigurationAggregator configurazione: DeleteConfigurationRecorder configurazione: DeleteConformancePack configurazione: DeleteDeliveryChannel configurazione: DeleteEvaluationResults configurazione: DeleteOrganizationConfigRule configurazione: DeleteOrganizationConformancePack configurazione: DeletePendingAggregationRequest configurazione: DeleteRemediationConfiguration configurazione: DeleteRemediationExceptions configurazione: DeleteResourceConfig configurazione: DeleteRetentionConfiguration configurazione: DeleteStoredQuery configurazione: DeliverConfigSnapshot configurazione: DescribeAggregateComplianceByConfigRules configurazione: DescribeAggregateComplianceByConformancePacks configurazione: DescribeAggregationAuthorizations

Prefisso del servizio	Azioni
	configurazione: DescribeComplianceByConfigRule
	configurazione: DescribeComplianceByResource
	configurazione: DescribeConfigRuleEvaluationStatus
	configurazione: DescribeConfigRules
	configurazione: DescribeConfigurationAggregatorSourcesStatus
	configurazione: DescribeConfigurationAggregators
	configurazione: DescribeConfigurationRecorderStatus
	configurazione: DescribeConfigurationRecorders
	configurazione: DescribeConformancePackCompliance
	configurazione: DescribeConformancePackStatus
	configurazione: DescribeConformancePacks
	configurazione: DescribeDeliveryChannelStatus
	configurazione: DescribeDeliveryChannels
	configurazione: DescribeOrganizationConfigRuleStatuses
	configurazione: DescribeOrganizationConfigRules
	configurazione: DescribeOrganizationConformancePackStatuses
	configurazione: DescribeOrganizationConformancePacks
	configurazione: DescribePendingAggregationRequests
	configurazione: DescribeRemediationConfigurations
	configurazione: DescribeRemediationExceptions
	configurazione: DescribeRemediationExecutionStatus

Prefisso del servizio	Azioni
	configurazione: DescribeRetentionConfigurations configurazione: GetComplianceDetailsByConfigRule configurazione: GetComplianceDetailsByResource configurazione: GetComplianceSummaryByConfigRule configurazione: GetComplianceSummaryByResourceType configurazione: GetConformancePackComplianceDetails configurazione: GetConformancePackComplianceSummary configurazione: GetCustomRulePolicy configurazione: GetDiscoveredResourceCounts configurazione: GetOrganizationConfigRuleDetailedStatus configurazione: GetOrganizationConformancePackDetailedStatus configurazione: GetOrganizationCustomRulePolicy configurazione: GetResourceConfigHistory configurazione: GetResourceEvaluationSummary configurazione: GetStoredQuery configurazione: ListConfigurationRecorders configurazione: ListConformancePackComplianceScores configurazione: ListConnectors configurazione: ListDiscoveredResources configurazione: ListResourceEvaluations configurazione: ListStoredQueries

Prefisso del servizio	Azioni
	configurazione: PutConfigRule configurazione: PutConfigurationAggregator configurazione: PutConfigurationRecorder configurazione: PutConformancePack configurazione: PutConnector configurazione: PutDeliveryChannel configurazione: PutEvaluations configurazione: PutExternalEvaluation configurazione: PutOrganizationConfigRule configurazione: PutOrganizationConformancePack configurazione: PutRemediationConfigurations configurazione: PutRemediationExceptions configurazione: PutResourceConfig configurazione: PutRetentionConfiguration configurazione: PutStoredQuery configurazione: SelectResourceConfig configurazione: StartConfigRulesEvaluation configurazione: StartConfigurationRecorder configurazione: StartRemediationExecution configurazione: StartResourceEvaluation configurazione: StopConfigurationRecorder

Prefisso del servizio	Azioni
connect	connettere: ActivateEvaluationForm
	connettere: AssociateAnalyticsDataSet
	connettere: AssociateApprovedOrigin
	connettere: AssociateBot
	connettere: AssociateContactWithUser
	connettere: AssociateDefaultVocabulary
	connettere: AssociateEmailAddressAlias
	connettere: AssociateFlow
	connettere: AssociateInstanceStorageConfig
	connettere: AssociateLambdaFunction
	connettere: AssociateLexBot
	connettere: AssociatePhoneNumberContactFlow
	connettere: AssociateQueueEmailAddresses
	connettere: AssociateQueueQuickConnects
	connettere: AssociateRoutingProfileQueues
	connettere: AssociateSecurityKey
	connettere: AssociateUserProficiencies
	connettere: BatchAssociateAnalyticsDataSet
	connettere: BatchCreateDataTableValue
	connettere: BatchDeleteDataTableValue
	connettere: BatchDescribeDataTableValue

Prefisso del servizio	Azioni
	connettere: BatchDisassociateAnalyticsDataSet
	connettere: BatchGetFlowAssociation
	connettere: BatchPutContact
	connettere: BatchUpdateDataTableValue
	connettere: ClaimPhoneNumber
	connettere: CreateAgentStatus
	connettere: CreateAttachedFile
	connettere: CreateContact
	connettere: CreateContactFlow
	connettere: CreateContactFlowModule
	connettere: CreateContactFlowModuleAlias
	connettere: CreateContactFlowModuleVersion
	connettere: CreateContactFlowVersion
	connettere: CreateDataTable
	connettere: CreateDataTableAttribute
	connettere: CreateEmailAddress
	connettere: CreateEvaluationForm
	connettere: CreateHoursOfOperation
	connettere: CreateInstance
	connettere: CreateIntegrationAssociation
	connettere: CreateParticipant

Prefisso del servizio	Azioni
	connettere: CreatePersistentContactAssociation
	connettere: CreatePredefinedAttribute
	connettere: CreatePrompt
	connettere: CreatePushNotificationRegistration
	connettere: CreateQueue
	connettere: CreateQuickConnect
	connettere: CreateRoutingProfile
	connettere: CreateRule
	connettere: CreateSecurityProfile
	connettere: CreateTaskTemplate
	connettere: CreateTrafficDistributionGroup
	connettere: CreateUseCase
	connettere: CreateUser
	connettere: CreateUserHierarchyGroup
	connettere: CreateView
	connettere: CreateViewVersion
	connettere: CreateVocabulary
	connettere: CreateWorkspace
	connettere: DeactivateEvaluationForm
	connettere: DeleteContactEvaluation
	connettere: DeleteContactFlow

Prefisso del servizio	Azioni
	connettere: DeleteContactFlowModule
	connettere: DeleteContactFlowModuleAlias
	connettere: DeleteContactFlowModuleVersion
	connettere: DeleteContactFlowVersion
	connettere: DeleteDataTable
	connettere: DeleteDataTableAttribute
	connettere: DeleteEmailAddress
	connettere: DeleteEvaluationForm
	connettere: DeleteHoursOfOperation
	connettere: DeleteHoursOfOperationOverride
	connettere: DeleteInstance
	connettere: DeleteIntegrationAssociation
	connettere: DeleteNotification
	connettere: DeletePredefinedAttribute
	connettere: DeletePrompt
	connettere: DeletePushNotificationRegistration
	connettere: DeleteQueue
	connettere: DeleteQuickConnect
	connettere: DeleteRoutingProfile
	connettere: DeleteRule
	connettere: DeleteSecurityProfile

Prefisso del servizio	Azioni
	connettere: DeleteTaskTemplate
	connettere: DeleteTrafficDistributionGroup
	connettere: DeleteUseCase
	connettere: DeleteUser
	connettere: DeleteUserHierarchyGroup
	connettere: DeleteView
	connettere: DeleteVocabulary
	connettere: DeleteWorkspace
	connettere: DeleteWorkspaceMedia
	connettere: DescribeAttachedFilesConfiguration
	connettere: DescribeAuthenticationProfile
	connettere: DescribeContactFlowModuleAlias
	connettere: DescribeDataTableAttribute
	connettere: DescribeHoursOfOperationOverride
	connettere: DescribeInstanceAttribute
	connettere: DescribeInstanceStorageConfig
	connettere: DescribeNotification
	connettere: DescribePhoneNumber
	connettere: DescribeRule
	connettere: DescribeTrafficDistributionGroup
	connettere: DescribeUserHierarchyStructure

Prefisso del servizio	Azioni
	connettere: DescribeVocabulary
	connettere: DisassociateAnalyticsDataSet
	connettere: DisassociateApprovedOrigin
	connettere: DisassociateBot
	connettere: DisassociateEmailAddressAlias
	connettere: DisassociateFlow
	connettere: DisassociateInstanceStorageConfig
	connettere: DisassociateLambdaFunction
	connettere: DisassociateLexBot
	connettere: DisassociatePhoneNumberContactFlow
	connettere: DisassociateQueueEmailAddresses
	connettere: DisassociateQueueQuickConnects
	connettere: DisassociateRoutingProfileQueues
	connettere: DisassociateSecurityKey
	connettere: DisassociateUserProficiencies
	connettere: DismissUserContact
	connettere: EvaluateDataTableValues
	connettere: GetContactAttributes
	connettere: GetContactMetrics
	connettere: GetCurrentMetricData
	connettere: GetCurrentUserData

Prefisso del servizio	Azioni
	connettere: GetEffectiveHoursOfOperations
	connettere: GetEvaluationFormValidation
	connettere: GetFederationToken
	connettere: GetFlowAssociation
	connettere: GetMetricData
	connettere: GetMetricDataV2
	connettere: GetPromptFile
	connettere: GetTaskTemplate
	connettere: GetTrafficDistribution
	connettere: ImportPhoneNumber
	connettere: ImportWorkspaceMedia
	connettere: ListAnalyticsDataAssociations
	connettere: ListAnalyticsDataLakeDataSets
	connettere: ListApprovedOrigins
	connettere: ListAssociatedContacts
	connettere: ListAttachedFilesConfigurations
	connettere: ListAuthenticationProfiles
	connettere: ListBots
	connettere: ListContactEvaluations
	connettere: ListContactFlowModuleAliases
	connettere: ListContactFlowModuleVersions

Prefisso del servizio	Azioni
	connettere: ListContactFlowModules
	connettere: ListContactFlowVersions
	connettere: ListContactFlows
	connettere: ListContactReferences
	connettere: ListDataTableAttributes
	connettere: ListDataTablePrimaryValues
	connettere: ListDataTableValues
	connettere: ListDataTables
	connettere: ListDefaultVocabularies
	connettere: ListEvaluationFormVersions
	connettere: ListEvaluationForms
	connettere: ListFlowAssociations
	connettere: ListHoursOfOperations
	connettere: ListInstanceAttributes
	connettere: ListInstanceStorageConfigs
	connettere: ListIntegrationAssociations
	connettere: ListLambdaFunctions
	connettere: ListLexBots
	connettere: ListNotifications
	connettere: ListPhoneNumbers
	connettere: ListPhoneNumbersV2

Prefisso del servizio	Azioni
	connettere: ListPredefinedAttributes
	connettere: ListPrompts
	connettere: ListQueueEmailAddresses
	connettere: ListQueueQuickConnects
	connettere: ListQueues
	connettere: ListQuickConnects
	connettere: ListRealtimeContactAnalysisSegmentsV2
	connettere: ListRoutingProfileManualAssignmentQueues
	connettere: ListRoutingProfileQueues
	connettere: ListRoutingProfiles
	connettere: ListRules
	connettere: ListSecurityKeys
	connettere: ListSecurityProfileApplications
	connettere: ListSecurityProfileFlowModules
	connettere: ListSecurityProfilePermissions
	connettere: ListSecurityProfiles
	connettere: ListTaskTemplates
	connettere: ListTrafficDistributionGroups
	connettere: ListUseCases
	connettere: ListUserHierarchyGroups
	connettere: ListUserNotifications

Prefisso del servizio	Azioni
	connettere: ListUsers
	connettere: ListViewVersions
	connettere: ListViews
	connettere: ListWorkspaceMedia
	connettere: ListWorkspacePages
	connettere: ListWorkspaces
	connettere: MonitorContact
	connettere: PauseContact
	connettere: PutUserStatus
	connettere: ReleasePhoneNumber
	connettere: ReplicateInstance
	connettere: ResumeContact
	connettere: ResumeContactRecording
	connettere: SearchAgentStatuses
	connettere: SearchAvailablePhoneNumbers
	connettere: SearchContactEvaluations
	connettere: SearchContactFlowModules
	connettere: SearchContactFlows
	connettere: SearchContacts
	connettere: SearchDataTables
	connettere: SearchEmailAddresses

Prefisso del servizio	Azioni
	connettere: SearchEvaluationForms
	connettere: SearchHoursOfOperations
	connettere: SearchPredefinedAttributes
	connettere: SearchPrompts
	connettere: SearchQueues
	connettere: SearchQuickConnects
	connettere: SearchRoutingProfiles
	connettere: SearchRules
	connettere: SearchSecurityProfiles
	connettere: SearchUserHierarchyGroups
	connettere: SearchViews
	connettere: SearchVocabularies
	connettere: SearchWorkspaceAssociations
	connettere: SearchWorkspaces
	connettere: SendChatIntegrationEvent
	connettere: SendOutboundEmail
	connettere: StartChatContact
	connettere: StartContactConversationalAnalyticsJob
	connettere: StartContactEvaluation
	connettere: StartContactMediaProcessing
	connettere: StartContactRecording

Prefisso del servizio	Azioni
	connettere: StartContactStreaming
	connettere: StartEmailContact
	connettere: StartOutboundChatContact
	connettere: StartOutboundEmailContact
	connettere: StartOutboundVoiceContact
	connettere: StartScreenSharing
	connettere: StartTaskContact
	connettere: StartWebRTCContact
	connettere: StopContact
	connettere: StopContactMediaProcessing
	connettere: StopContactRecording
	connettere: StopContactStreaming
	connettere: SubmitContactEvaluation
	connettere: SuspendContactRecording
	connettere: TransferContact
	connettere: UpdateAgentStatus
	connettere: UpdateAttachedFilesConfiguration
	connettere: UpdateAuthenticationProfile
	connettere: UpdateContact
	connettere: UpdateContactAttributes
	connettere: UpdateContactEvaluation

Prefisso del servizio	Azioni
	connettere: UpdateContactFlowContent
	connettere: UpdateContactFlowMetadata
	connettere: UpdateContactFlowModuleAlias
	connettere: UpdateContactFlowModuleContent
	connettere: UpdateContactFlowModuleMetadata
	connettere: UpdateContactFlowName
	connettere: UpdateContactRoutingData
	connettere: UpdateContactSchedule
	connettere: UpdateDataTableAttribute
	connettere: UpdateDataTableMetadata
	connettere: UpdateDataTablePrimaryValues
	connettere: UpdateEmailAddressMetadata
	connettere: UpdateEvaluationForm
	connettere: UpdateHoursOfOperation
	connettere: UpdateHoursOfOperationOverride
	connettere: UpdateInstanceAttribute
	connettere: UpdateInstanceStorageConfig
	connettere: UpdateNotificationContent
	connettere: UpdateParticipantAuthentication
	connettere: UpdateParticipantRoleConfig
	connettere: UpdatePhoneNumber

Prefisso del servizio	Azioni
	connettere: UpdatePhoneNumberMetadata
	connettere: UpdatePredefinedAttribute
	connettere: UpdatePrompt
	connettere: UpdateQueueHoursOfOperation
	connettere: UpdateQueueMaxContacts
	connettere: UpdateQueueName
	connettere: UpdateQueueOutboundCallerConfig
	connettere: UpdateQueueOutboundEmailConfig
	connettere: UpdateQueueStatus
	connettere: UpdateQuickConnectConfig
	connettere: UpdateQuickConnectName
	connettere: UpdateRoutingProfileAgentAvailabilityTimer
	connettere: UpdateRoutingProfileConcurrency
	connettere: UpdateRoutingProfileDefaultOutboundQueue
	connettere: UpdateRoutingProfileName
	connettere: UpdateRoutingProfileQueues
	connettere: UpdateRule
	connettere: UpdateSecurityProfile
	connettere: UpdateTaskTemplate
	connettere: UpdateTrafficDistribution
	connettere: UpdateUserHierarchy

Prefisso del servizio	Azioni
	connettere: UpdateUserHierarchyGroupName connettere: UpdateUserHierarchyStructure connettere: UpdateUserIdentityInfo connettere: UpdateUserNotificationStatus connettere: UpdateUserPhoneConfig connettere: UpdateUserProficiencies connettere: UpdateUserRoutingProfile connettere: UpdateUserSecurityProfiles connettere: UpdateViewContent connettere: UpdateViewMetadata connettere: UpdateWorkspaceMetadata connettere: UpdateWorkspaceTheme connettere: UpdateWorkspaceVisibility
cur	cura: DeleteReportDefinition cura: DescribeReportDefinitions cura: ModifyReportDefinition cura: PutReportDefinition

Prefisso del servizio	Azioni
databrew	data brew: BatchDeleteRecipeVersion
	data brew: CreateDataset
	data brew: CreateProfileJob
	data brew: CreateProject
	data brew: CreateRecipe
	data brew: CreateRecipeJob
	data brew: CreateRuleset
	data brew: CreateSchedule
	data brew: DeleteDataset
	data brew: DeleteJob
	data brew: DeleteProject
	data brew: DeleteRecipeVersion
	data brew: DeleteRuleset
	data brew: DeleteSchedule
	data brew: DescribeDataset
	data brew: DescribeJob
	data brew: DescribeJobRun
	data brew: DescribeProject
	data brew: DescribeRecipe
	data brew: DescribeRuleset
	data brew: DescribeSchedule

Prefisso del servizio	Azioni
	data brew: ListDatasets
	data brew: ListJobRuns
	data brew: ListJobs
	data brew: ListProjects
	data brew: ListRecipeVersions
	data brew: ListRecipes
	data brew: ListRulesets
	data brew: ListSchedules
	data brew: PublishRecipe
	data brew: SendProjectSessionAction
	data brew: StartJobRun
	data brew: StartProjectSession
	data brew: StopJobRun
	data brew: UpdateDataset
	data brew: UpdateProfileJob
	data brew: UpdateProject
	data brew: UpdateRecipe
	data brew: UpdateRecipeJob
	data brew: UpdateRuleset
	data brew: UpdateSchedule

Prefisso del servizio	Azioni
dataexchange	scambio di dati: AcceptDataGrant scambio di dati: CancelJob scambio di dati: CreateDataGrant scambio di dati: CreateDataSet scambio di dati: CreateEventAction scambio di dati: CreateJob scambio di dati: CreateRevision scambio di dati: DeleteAsset scambio di dati: DeleteDataGrant scambio di dati: DeleteEventAction scambio di dati: DeleteRevision scambio di dati: GetDataGrant scambio di dati: GetEventAction scambio di dati: GetJob scambio di dati: GetReceivedDataGrant scambio di dati: ListDataGrants scambio di dati: ListDataSetRevisions scambio di dati: ListDataSets scambio di dati: ListEventActions scambio di dati: ListJobs scambio di dati: ListReceivedDataGrants

Prefisso del servizio	Azioni
	scambio di dati: ListRevisionAssets scambio di dati: RevokeRevision scambio di dati: SendDataSetNotification scambio di dati: StartJob scambio di dati: UpdateAsset scambio di dati: UpdateDataSet scambio di dati: UpdateEventAction scambio di dati: UpdateRevision

Prefisso del servizio	Azioni
datapipeline	pipeline dati: ActivatePipeline
	tubazione dati: CreatePipeline
	tubazione dati: DeactivatePipeline
	tubazione dati: DeletePipeline
	tubazione dati: DescribeObjects
	tubazione dati: DescribePipelines
	tubazione dati: EvaluateExpression
	tubazione dati: GetPipelineDefinition
	tubazione dati: ListPipelines
	tubazione dati: PollForTask
	tubazione dati: PutPipelineDefinition
	tubazione dati: QueryObjects
	tubazione dati: ReportTaskProgress
	tubazione dati: ReportTaskRunnerHeartbeat
	tubazione dati: SetStatus
	tubazione dati: SetTaskStatus
	tubazione dati: ValidatePipelineDefinition

Prefisso del servizio	Azioni
dax	fax: CreateCluster fax: DecreaseReplicationFactor fax: DeleteCluster fax: DeleteParameterGroup fax: DeleteSubnetGroup fax: DescribeClusters fax: DescribeDefaultParameters fax: DescribeEvents fax: DescribeParameterGroups fax: DescribeParameters fax: DescribeSubnetGroups fax: IncreaseReplicationFactor fax: RebootNode fax: UpdateCluster fax: UpdateParameterGroup fax: UpdateSubnetGroup

Prefisso del servizio	Azioni
devicefarm	fabbrica di dispositivi: CreateDevicePool
	fabbrica di dispositivi: CreateInstanceProfile
	fabbrica di dispositivi: CreateNetworkProfile
	fabbrica di dispositivi: CreateProject
	fabbrica di dispositivi: CreateRemoteAccessSession
	fabbrica di dispositivi: CreateTestGridProject
	fabbrica di dispositivi: CreateTestGridUrl
	fabbrica di dispositivi: CreateUpload
	devicefarm:CreateVPCEConfiguration
	fabbrica di dispositivi: DeleteDevicePool
	fabbrica di dispositivi: DeleteInstanceProfile
	fabbrica di dispositivi: DeleteNetworkProfile
	fabbrica di dispositivi: DeleteProject
	fabbrica di dispositivi: DeleteRemoteAccessSession
	fabbrica di dispositivi: DeleteRun
	fabbrica di dispositivi: DeleteTestGridProject
	fattoria di dispositivi: DeleteUpload
	devicefarm:DeleteVPCEConfiguration
	fattoria di dispositivi: GetAccountSettings
	fattoria di dispositivi: GetDevice
	fattoria di dispositivi: GetDeviceInstance

Prefisso del servizio	Azioni
	fattoria di dispositivi: GetDevicePool
	fattoria di dispositivi: GetDevicePoolCompatibility
	fattoria di dispositivi: GetInstanceProfile
	fattoria di dispositivi: GetJob
	fattoria di dispositivi: GetNetworkProfile
	fattoria di dispositivi: GetOfferingStatus
	fattoria di dispositivi: GetProject
	fattoria di dispositivi: GetRemoteAccessSession
	fattoria di dispositivi: GetRun
	fattoria di dispositivi: GetSuite
	fattoria di dispositivi: GetTest
	fattoria di dispositivi: GetTestGridProject
	fattoria di dispositivi: GetTestGridSession
	fattoria di dispositivi: GetUpload
	devicefarm:GetVPCEConfiguration
	fattoria di dispositivi: ListArtifacts
	fattoria di dispositivi: ListDeviceInstances
	fattoria di dispositivi: ListDevicePools
	fattoria di dispositivi: ListDevices
	fattoria di dispositivi: ListInstanceProfiles
	fattoria di dispositivi: ListJobs

Prefisso del servizio	Azioni
	fattoria di dispositivi: ListNetworkProfiles
	fattoria di dispositivi: ListOfferingPromotions
	fattoria di dispositivi: ListOfferingTransactions
	fattoria di dispositivi: ListOfferings
	fattoria di dispositivi: ListProjects
	fattoria di dispositivi: ListRemoteAccessSessions
	fattoria di dispositivi: ListRuns
	fattoria di dispositivi: ListSamples
	fattoria di dispositivi: ListSuites
	fattoria di dispositivi: ListTestGridProjects
	fattoria di dispositivi: ListTestGridSessionActions
	fattoria di dispositivi: ListTestGridSessionArtifacts
	fattoria di dispositivi: ListTestGridSessions
	fattoria di dispositivi: ListTests
	fattoria di dispositivi: ListUniqueProblems
	fattoria di dispositivi: ListUploads
	devicefarm:ListVPCEConfigurations
	fattoria di dispositivi: PurchaseOffering
	fattoria di dispositivi: RenewOffering
	fattoria di dispositivi: ScheduleRun
	fattoria di dispositivi: StopJob

Prefisso del servizio	Azioni
	fattoria di dispositivi: StopRemoteAccessSession
	fattoria di dispositivi: StopRun
	fattoria di dispositivi: UpdateDeviceInstance
	fattoria di dispositivi: UpdateDevicePool
	fattoria di dispositivi: UpdateInstanceProfile
	fattoria di dispositivi: UpdateNetworkProfile
	fattoria di dispositivi: UpdateProject
	fattoria di dispositivi: UpdateTestGridProject
	fattoria di dispositivi: UpdateUpload
	devicefarm:UpdateVPCEConfiguration

Prefisso del servizio	Azioni
devops-guru	devops-guru: AddNotificationChannel devops-guru: DeleteInsight devops-guru: DescribeAccountHealth devops-guru: DescribeAccountOverview devops-guru: DescribeAnomaly devops-guru: DescribeEventSourcesConfig devops-guru: DescribeFeedback devops-guru: DescribeInsight devops-guru: DescribeOrganizationHealth devops-guru: DescribeOrganizationOverview devops-guru: DescribeOrganizationResourceCollectionHealth devops-guru: DescribeResourceCollectionHealth devops-guru: DescribeServiceIntegration devops-guru: GetCostEstimation devops-guru: GetResourceCollection devops-guru: ListAnomaliesForInsight devops-guru: ListAnomalousLogGroups devops-guru: ListEvents devops-guru: ListInsights devops-guru: ListMonitoredResources devops-guru: ListNotificationChannels

Prefisso del servizio	Azioni
	devops-guru: ListOrganizationInsights devops-guru: ListRecommendations devops-guru: PutFeedback devops-guru: RemoveNotificationChannel devops-guru: SearchInsights devops-guru: SearchOrganizationInsights devops-guru: StartCostEstimation devops-guru: UpdateEventSourcesConfig devops-guru: UpdateResourceCollection devops-guru: UpdateServiceIntegration

Prefisso del servizio	Azioni
directconnect	connessione diretta: AcceptDirectConnectGatewayAssociationProposal
	connessione diretta: AllocateConnectionOnInterconnect
	connessione diretta: AllocateHostedConnection
	connessione diretta: AllocatePrivateVirtualInterface
	connessione diretta: AllocatePublicVirtualInterface
	connessione diretta: AllocateTransitVirtualInterface
	connessione diretta: AssociateConnectionWithLag
	connessione diretta: AssociateHostedConnection
	connessione diretta: AssociateMacSecKey
	connessione diretta: AssociateVirtualInterface
	connessione diretta: ConfirmConnection
	connessione diretta: ConfirmCustomerAgreement
	connessione diretta: ConfirmPrivateVirtualInterface
	connessione diretta: ConfirmPublicVirtualInterface
	connessione diretta: ConfirmTransitVirtualInterface
	directconnect:CreateBGPPeer
	connessione diretta: CreateConnection
	connessione diretta: CreateDirectConnectGateway
	connessione diretta: CreateDirectConnectGatewayAssociation
	connessione diretta: CreateDirectConnectGatewayAssociationProposal

Prefisso del servizio	Azioni
	connessione diretta: CreateInterconnect
	connessione diretta: CreateLag
	connessione diretta: CreatePrivateVirtualInterface
	connessione diretta: CreatePublicVirtualInterface
	connessione diretta: CreateTransitVirtualInterface
	directconnect:DeleteBGPPeer
	connessione diretta: DeleteConnection
	connessione diretta: DeleteDirectConnectGateway
	connessione diretta: DeleteDirectConnectGatewayAssociation
	connessione diretta: DeleteDirectConnectGatewayAssociationProposal
	connessione diretta: DeleteInterconnect
	connessione diretta: DeleteLag
	connessione diretta: DeleteVirtualInterface
	connessione diretta: DescribeConnectionLoa
	connessione diretta: DescribeConnections
	connessione diretta: DescribeConnectionsOnInterconnect
	connessione diretta: DescribeCustomerMetadata
	connessione diretta: DescribeDirectConnectGatewayAssociationProposals
	connessione diretta: DescribeDirectConnectGatewayAssociations
	connessione diretta: DescribeDirectConnectGatewayAttachments

Prefisso del servizio	Azioni
	connessione diretta: DescribeDirectConnectGateways
	connessione diretta: DescribeHostedConnections
	connessione diretta: DescribeInterconnectLoa
	connessione diretta: DescribeInterconnects
	connessione diretta: DescribeLags
	connessione diretta: DescribeLoa
	connessione diretta: DescribeLocations
	connessione diretta: DescribeRouterConfiguration
	connessione diretta: DescribeVirtualGateways
	connessione diretta: DescribeVirtualInterfaces
	connessione diretta: DisassociateConnectionFromLag
	connessione diretta: DisassociateMacSecKey
	connessione diretta: ListVirtualInterfaceTestHistory
	connessione diretta: StartBgpFailoverTest
	connessione diretta: StopBgpFailoverTest
	connessione diretta: UpdateConnection
	connessione diretta: UpdateDirectConnectGateway
	connessione diretta: UpdateDirectConnectGatewayAssociation
	connessione diretta: UpdateLag
	connessione diretta: UpdateVirtualInterfaceAttributes

Prefisso del servizio	Azioni
dlm	dm: CreateLifecyclePolicy dpm: DeleteLifecyclePolicy dpm: GetLifecyclePolicies dpm: GetLifecyclePolicy dpm: UpdateLifecyclePolicy

Prefisso del servizio	Azioni
dms	dighe: ApplyPendingMaintenanceAction
	dms: AssociateExtensionPack
	dms: BatchStartRecommendations
	dms: CancelMetadataModelCreation
	dms: CancelReplicationTaskAssessmentRun
	dms: CreateDataProvider
	dms: CreateEndpoint
	dms: CreateEventSubscription
	dms: CreateInstanceProfile
	dms: CreateMigrationProject
	dms: CreateReplicationConfig
	dms: CreateReplicationInstance
	dms: CreateReplicationSubnetGroup
	dms: CreateReplicationTask
	dms: DeleteCertificate
	dms: DeleteConnection
	dms: DeleteDataMigration
	dms: DeleteDataProvider
	dms: DeleteEndpoint
	dms: DeleteEventSubscription
	dms: DeleteFleetAdvisorCollector

Prefisso del servizio	Azioni
	dms: DeleteFleetAdvisorDatabases
	dms: DeleteInstanceProfile
	dms: DeleteMigrationProject
	dms: DeleteReplicationConfig
	dms: DeleteReplicationInstance
	dms: DeleteReplicationSubnetGroup
	dms: DeleteReplicationTask
	dms: DeleteReplicationTaskAssessmentRun
	dms: DescribeAccountAttributes
	dms: DescribeApplicableIndividualAssessments
	dms: DescribeCertificates
	dms: DescribeConnections
	dms: DescribeDataMigrations
	dms: DescribeEndpointSettings
	dms: DescribeEndpointTypes
	dms: DescribeEndpoints
	dms: DescribeEngineVersions
	dms: DescribeEventCategories
	dms: DescribeEventSubscriptions
	dms: DescribeEvents
	dms: DescribeFleetAdvisorCollectors

Prefisso del servizio	Azioni
	dms: DescribeFleetAdvisorDatabases
	dms: DescribeFleetAdvisorLsaAnalysis
	dms: DescribeFleetAdvisorSchemaObjectSummary
	dms: DescribeFleetAdvisorSchemas
	dms: DescribeMetadataModel
	dms: DescribeMetadataModelChildren
	dms: DescribeMetadataModelCreations
	dms: DescribeMetadataModelImports
	dms: DescribeOrderableReplicationInstances
	dms: DescribePendingMaintenanceActions
	dms: DescribeRecommendationLimitations
	dms: DescribeRecommendations
	dms: DescribeRefreshSchemasStatus
	dms: DescribeReplicationConfigs
	dms: DescribeReplicationInstanceTaskLogs
	dms: DescribeReplicationInstances
	dms: DescribeReplicationSubnetGroups
	dms: DescribeReplicationTableStatistics
	dms: DescribeReplicationTaskAssessmentResults
	dms: DescribeReplicationTaskAssessmentRuns
	dms: DescribeReplicationTaskIndividualAssessments

Prefisso del servizio	Azioni
	dms: DescribeReplicationTasks
	dms: DescribeReplications
	dms: DescribeSchemas
	dms: DescribeTableStatistics
	dms: ExportMetadataModelAssessment
	dms: ImportCertificate
	dms: ListDataProviders
	dms: ListExtensionPacks
	dms: ListInstanceProfiles
	dms: ListMetadataModelAssessments
	dms: ListMetadataModelConversions
	dms: ListMetadataModelExports
	dms: ListMigrationProjects
	dms: ModifyDataMigration
	dms: ModifyEndpoint
	dms: ModifyEventSubscription
	dms: ModifyReplicationConfig
	dms: ModifyReplicationInstance
	dms: ModifyReplicationSubnetGroup
	dms: ModifyReplicationTask
	dms: MoveReplicationTask

Prefisso del servizio	Azioni
	dms: RebootReplicationInstance
	dms: RefreshSchemas
	dms: ReloadReplicationTables
	dms: ReloadTables
	dms: RunFleetAdvisorLsaAnalysis
	dms: StartMetadataModelAssessment
	dms: StartMetadataModelConversion
	dms: StartMetadataModelCreation
	dms: StartMetadataModelExportAsScripts
	dms: StartMetadataModelExportToTarget
	dms: StartRecommendations
	dms: StartReplication
	dms: StartReplicationTask
	dms: StartReplicationTaskAssessment
	dms: StopDataMigration
	dms: StopReplicationTask
	dms: TestConnection
	dms: UpdateConversionConfiguration
	dms: UpdateDataProvider
	dms: UpdateInstanceProfile
	dms: UpdateMigrationProject

Prefisso del servizio	Azioni
	dms: UpdateSubscriptionsToEventBridge
docdb-elastic	docdb elastico: ApplyPendingMaintenanceAction docdb elastico: CopyClusterSnapshot docdb elastico: DeleteCluster docdb elastico: DeleteClusterSnapshot docdb elastico: GetCluster docdb elastico: GetClusterSnapshot docdb elastico: GetPendingMaintenanceAction docdb elastico: ListClusterSnapshots docdb elastico: ListClusters docdb elastico: ListPendingMaintenanceActions docdb elastico: RestoreClusterFromSnapshot docdb elastico: StartCluster docdb elastico: StopCluster docdb elastico: UpdateCluster

Prefisso del servizio	Azioni
dynamodb	dinamodb: CreateBackup
	dinamodb: CreateGlobalTable
	dinamodb: CreateTable
	dinamodb: DeleteBackup
	dinamodb: DeleteTable
	dinamodb: DescribeBackup
	dinamodb: DescribeContinuousBackups
	dinamodb: DescribeContributorInsights
	dinamodb: DescribeEndpoints
	dinamodb: DescribeExport
	dinamodb: DescribeGlobalTable
	dinamodb: DescribeGlobalTableSettings
	dinamodb: DescribeImport
	dinamodb: DescribeKinesisStreamingDestination
	dinamodb: DescribeLimits
	dinamodb: DescribeStream
	dinamodb: DescribeTable
	dinamodb: DescribeTableReplicaAutoScaling
	dinamodb: DescribeTimeToLive
	dinamodb: DisableKinesisStreamingDestination
	dinamodb: EnableKinesisStreamingDestination

Prefisso del servizio	Azioni
	dinamodb: ExportTableToPointInTime
	dinamodb: GetResourcePolicy
	dinamodb: ImportTable
	dinamodb: ListBackups
	dinamodb: ListContributorInsights
	dinamodb: ListExports
	dinamodb: ListGlobalTables
	dinamodb: ListImports
	dinamodb: ListStreams
	dinamodb: ListTables
	dinamodb: RestoreTableFromBackup
	dinamodb: RestoreTableToPointInTime
	dinamodb: UpdateContinuousBackups
	dinamodb: UpdateContributorInsights
	dinamodb: UpdateGlobalTable
	dinamodb: UpdateGlobalTableSettings
	dinamodb: UpdateKinesisStreamingDestination
	dinamodb: UpdateTable
	dinamodb: UpdateTableReplicaAutoScaling
	dinamodb: UpdateTimeToLive

Prefisso del servizio	Azioni
ebs	Web: CompleteSnapshot ebs: StartSnapshot

Prefisso del servizio	Azioni
ec2	ec2: AcceptAddressTransfer
	ec2: AcceptCapacityReservationBillingOwnership
	ec2: AcceptReservedInstancesExchangeQuote
	ec2: AcceptTransitGatewayClientVpnAttachment
	ec2: AcceptTransitGatewayMulticastDomainAssociations
	ec2: AcceptTransitGatewayPeeringAttachment
	ec2: AcceptTransitGatewayVpcAttachment
	ec2: AcceptVpcEndpointConnections
	ec2: AcceptVpcPeeringConnection
	ec2: AdvertiseByoipCidr
	ec2: AllocateAddress
	ec2: AllocateHosts
	ec2: AllocateIpamPoolCidr
	ec2: ApplySecurityGroupsToClientVpnTargetNetwork
	ec2: AssignIpv6Addresses
	ec2: AssignPrivateIpAddresses
	ec2: AssignPrivateNatGatewayAddress
	ec2: AssociateAddress
	ec2: AssociateCapacityReservationBillingOwner
	ec2: AssociateClientVpnTargetNetwork
	ec2: AssociateDhcpOptions

Prefisso del servizio	Azioni
	ec2: AssociateEnclaveCertificateIamRole
	ec2: AssociateIamInstanceProfile
	ec2: AssociateInstanceEventWindow
	ec2: AssociateIpamByoasn
	ec2: AssociateIpamResourceDiscovery
	ec2: AssociateNatGatewayAddress
	ec2: AssociateRouteServer
	ec2: AssociateRouteTable
	ec2: AssociateSecurityGroupVpc
	ec2: AssociateSubnetCidrBlock
	ec2: AssociateTransitGatewayMulticastDomain
	ec2: AssociateTransitGatewayPolicyTable
	ec2: AssociateTransitGatewayRouteTable
	ec2: AssociateTrunkInterface
	ec2: AssociateVpcCidrBlock
	ec2: AttachClassicLinkVpc
	ec2: AttachImageWatermark
	ec2: AttachInternetGateway
	ec2: AttachNetworkInterface
	ec2: AttachVerifiedAccessTrustProvider
	ec2: AttachVolume

Prefisso del servizio	Azioni
	ec2: AttachVpnGateway
	ec2: AuthorizeClientVpnIngress
	ec2: AuthorizeSecurityGroupEgress
	ec2: AuthorizeSecurityGroupIngress
	ec2: BundleInstance
	ec2: CancelBundleTask
	ec2: CancelCapacityReservation
	ec2: CancelCapacityReservationFleets
	ec2: CancelConversionTask
	ec2: CancelDeclarativePoliciesReport
	ec2: CancelExportTask
	ec2: CancelImageLaunchPermission
	ec2: CancelImportTask
	ec2: CancelReservedInstancesListing
	ec2: CancelSpotFleetRequests
	ec2: CancelSpotInstanceRequests
	ec2: ConfirmProductInstance
	ec2: CopyFpgaImage
	ec2: CopyImage
	ec2: CopySnapshot
	ec2: CopyVolumes

Prefisso del servizio	Azioni
	ec2: CreateCapacityManagerDataExport
	ec2: CreateCapacityReservation
	ec2: CreateCapacityReservationBySplitting
	ec2: CreateCapacityReservationCancellationQuote
	ec2: CreateCapacityReservationFleet
	ec2: CreateCarrierGateway
	ec2: CreateClientVpnEndpoint
	ec2: CreateClientVpnRoute
	ec2: CreateCoipCidr
	ec2: CreateCoipPool
	ec2: CreateCustomerGateway
	ec2: CreateDefaultSubnet
	ec2: CreateDefaultVpc
	ec2: CreateDelegateMacVolumeOwnershipTask
	ec2: CreateDhcpOptions
	ec2: CreateEgressOnlyInternetGateway
	ec2: CreateFleet
	ec2: CreateFlowLogs
	ec2: CreateFpgaImage
	ec2: CreateImage
	ec2: CreateImageUsageReport

Prefisso del servizio	Azioni
	ec2: CreateInstanceConnectEndpoint
	ec2: CreateInstanceEventWindow
	ec2: CreateInstanceExportTask
	ec2: CreateInternetGateway
	ec2: CreateInterruptibleCapacityReservationAllocation
	ec2: CreateIpam
	ec2: CreateIpamExternalResourceVerificationToken
	ec2: CreateIpamPolicy
	ec2: CreateIpamPool
	ec2: CreateIpamPrefixListResolver
	ec2: CreateIpamPrefixListResolverTarget
	ec2: CreateIpamResourceDiscovery
	ec2: CreateIpamScope
	ec2: CreateKeyPair
	ec2: CreateLaunchTemplateVersion
	ec2: CreateLocalGatewayRoute
	ec2: CreateLocalGatewayRouteTable
	ec2: CreateLocalGatewayRouteTableVirtualInterfaceGroupAssociation
	ec2: CreateLocalGatewayRouteTableVpcAssociation
	ec2: CreateLocalGatewayVirtualInterface

Prefisso del servizio	Azioni
	ec2: CreateLocalGatewayVirtualInterfaceGroup
	ec2: CreateMacSystemIntegrityProtectionModificationTask
	ec2: CreateManagedPrefixList
	ec2: CreateNatGateway
	ec2: CreateNetworkAcl
	ec2: CreateNetworkAclEntry
	ec2: CreateNetworkInsightsAccessScope
	ec2: CreateNetworkInsightsPath
	ec2: CreateNetworkInterface
	ec2: CreateNetworkInterfacePermission
	ec2: CreatePlacementGroup
	ec2: CreatePublicIpv4Pool
	ec2: CreateReplaceRootVolumeTask
	ec2: CreateReservedInstancesListing
	ec2: CreateRestoreImageTask
	ec2: CreateRoute
	ec2: CreateRouteServer
	ec2: CreateRouteServerEndpoint
	ec2: CreateRouteServerPeer
	ec2: CreateRouteTable
	ec2: CreateSecondaryNetwork

Prefisso del servizio	Azioni
	ec2: CreateSecondarySubnet
	ec2: CreateSecurityGroup
	ec2: CreateSnapshots
	ec2: CreateSpotDatafeedSubscription
	ec2: CreateStoreImageTask
	ec2: CreateSubnet
	ec2: CreateSubnetCidrReservation
	ec2: CreateTrafficMirrorFilter
	ec2: CreateTrafficMirrorFilterRule
	ec2: CreateTrafficMirrorSession
	ec2: CreateTrafficMirrorTarget
	ec2: CreateTransitGateway
	ec2: CreateTransitGatewayConnect
	ec2: CreateTransitGatewayConnectPeer
	ec2: CreateTransitGatewayMeteringPolicy
	ec2: CreateTransitGatewayMeteringPolicyEntry
	ec2: CreateTransitGatewayMulticastDomain
	ec2: CreateTransitGatewayPeeringAttachment
	ec2: CreateTransitGatewayPolicyTable
	ec2: CreateTransitGatewayPrefixListReference
	ec2: CreateTransitGatewayRoute

Prefisso del servizio	Azioni
	ec2: CreateTransitGatewayRouteTable
	ec2: CreateTransitGatewayRouteTableAnnouncement
	ec2: CreateTransitGatewayVpcAttachment
	ec2: CreateVerifiedAccessEndpoint
	ec2: CreateVerifiedAccessGroup
	ec2: CreateVerifiedAccessInstance
	ec2: CreateVerifiedAccessTrustProvider
	ec2: CreateVolume
	ec2: CreateVpc
	ec2: CreateVpcBlockPublicAccessExclusion
	ec2: CreateVpcEncryptionControl
	ec2: CreateVpcEndpoint
	ec2: CreateVpcEndpointConnectionNotification
	ec2: CreateVpcEndpointServiceConfiguration
	ec2: CreateVpcPeeringConnection
	ec2: CreateVpnConcentrator
	ec2: CreateVpnConnection
	ec2: CreateVpnConnectionRoute
	ec2: CreateVpnGateway
	ec2: DeleteCapacityManagerDataExport
	ec2: DeleteCarrierGateway

Prefisso del servizio	Azioni
	ec2: DeleteClientVpnEndpoint
	ec2: DeleteClientVpnRoute
	ec2: DeleteCoipCidr
	ec2: DeleteCoipPool
	ec2: DeleteCustomerGateway
	ec2: DeleteDhcpOptions
	ec2: DeleteEgressOnlyInternetGateway
	ec2: DeleteFleets
	ec2: DeleteFlowLogs
	ec2: DeleteFpgaImage
	ec2: DeleteImageUsageReport
	ec2: DeleteInstanceConnectEndpoint
	ec2: DeleteInstanceEventWindow
	ec2: DeleteInternetGateway
	ec2: DeleteIpam
	ec2: DeleteIpamExternalResourceVerificationToken
	ec2: DeleteIpamPolicy
	ec2: DeleteIpamPool
	ec2: DeleteIpamPrefixListResolver
	ec2: DeleteIpamPrefixListResolverTarget
	ec2: DeleteIpamResourceDiscovery

Prefisso del servizio	Azioni
	ec2: DeleteIpamScope
	ec2: DeleteKeyPair
	ec2: DeleteLaunchTemplate
	ec2: DeleteLaunchTemplateVersions
	ec2: DeleteLocalGatewayRoute
	ec2: DeleteLocalGatewayRouteTable
	ec2: DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssoc iation
	ec2: DeleteLocalGatewayRouteTableVpcAssociation
	ec2: DeleteLocalGatewayVirtualInterface
	ec2: DeleteLocalGatewayVirtualInterfaceGroup
	ec2: DeleteManagedPrefixList
	ec2: DeleteNatGateway
	ec2: DeleteNetworkAcl
	ec2: DeleteNetworkAclEntry
	ec2: DeleteNetworkInsightsAccessScope
	ec2: DeleteNetworkInsightsAccessScopeAnalysis
	ec2: DeleteNetworkInsightsAnalysis
	ec2: DeleteNetworkInsightsPath
	ec2: DeleteNetworkInterface
	ec2: DeleteNetworkInterfacePermission

Prefisso del servizio	Azioni
	ec2: DeletePlacementGroup
	ec2: DeletePublicIpv4Pool
	ec2: DeleteQueuedReservedInstances
	ec2: DeleteRoute
	ec2: DeleteRouteServer
	ec2: DeleteRouteServerEndpoint
	ec2: DeleteRouteServerPeer
	ec2: DeleteRouteTable
	ec2: DeleteSecondaryNetwork
	ec2: DeleteSecondarySubnet
	ec2: DeleteSecurityGroup
	ec2: DeleteSpotDatafeedSubscription
	ec2: DeleteSubnet
	ec2: DeleteSubnetCidrReservation
	ec2: DeleteTrafficMirrorFilter
	ec2: DeleteTrafficMirrorFilterRule
	ec2: DeleteTrafficMirrorSession
	ec2: DeleteTrafficMirrorTarget
	ec2: DeleteTransitGateway
	ec2: DeleteTransitGatewayClientVpnAttachment
	ec2: DeleteTransitGatewayConnect

Prefisso del servizio	Azioni
	ec2: DeleteTransitGatewayConnectPeer
	ec2: DeleteTransitGatewayMeteringPolicy
	ec2: DeleteTransitGatewayMeteringPolicyEntry
	ec2: DeleteTransitGatewayMulticastDomain
	ec2: DeleteTransitGatewayPeeringAttachment
	ec2: DeleteTransitGatewayPolicyTable
	ec2: DeleteTransitGatewayPrefixListReference
	ec2: DeleteTransitGatewayRoute
	ec2: DeleteTransitGatewayRouteTable
	ec2: DeleteTransitGatewayRouteTableAnnouncement
	ec2: DeleteTransitGatewayVpcAttachment
	ec2: DeleteVerifiedAccessEndpoint
	ec2: DeleteVerifiedAccessGroup
	ec2: DeleteVerifiedAccessInstance
	ec2: DeleteVerifiedAccessTrustProvider
	ec2: DeleteVolume
	ec2: DeleteVpc
	ec2: DeleteVpcBlockPublicAccessExclusion
	ec2: DeleteVpcEncryptionControl
	ec2: DeleteVpcEndpointConnectionNotifications
	ec2: DeleteVpcEndpointServiceConfigurations

Prefisso del servizio	Azioni
	ec2: DeleteVpcEndpoints
	ec2: DeleteVpcPeeringConnection
	ec2: DeleteVpnConcentrator
	ec2: DeleteVpnConnection
	ec2: DeleteVpnConnectionRoute
	ec2: DeleteVpnGateway
	ec2: DeprovisionByoipCidr
	ec2: DeprovisionIpamByoasn
	ec2: DeprovisionIpamPoolCidr
	ec2: DeprovisionPublicIpv4PoolCidr
	ec2: DeregisterImage
	ec2: DeregisterInstanceEventNotificationAttributes
	ec2: DeregisterTransitGatewayMulticastGroupMembers
	ec2: DeregisterTransitGatewayMulticastGroupSources
	ec2: DescribeAccountAttributes
	ec2: DescribeAddressTransfers
	ec2: DescribeAddresses
	ec2: DescribeAddressesAttribute
	ec2: DescribeAggregateIdFormat
	ec2: DescribeAvailabilityZones
	ec2: DescribeAwsNetworkPerformanceMetricSubscriptions

Prefisso del servizio	Azioni
	ec2: DescribeBundleTasks
	ec2: DescribeByoipCidrs
	ec2: DescribeCapacityBlockExtensionHistory
	ec2: DescribeCapacityBlockExtensionOfferings
	ec2: DescribeCapacityBlockStatus
	ec2: DescribeCapacityBlocks
	ec2: DescribeCapacityManagerDataExports
	ec2: DescribeCapacityReservationBillingRequests
	ec2: DescribeCapacityReservationCancellationQuotes
	ec2: DescribeCapacityReservationFleets
	ec2: DescribeCapacityReservationTopology
	ec2: DescribeCapacityReservations
	ec2: DescribeCarrierGateways
	ec2: DescribeClassicLinkInstances
	ec2: DescribeClientVpnAuthorizationRules
	ec2: DescribeClientVpnConnections
	ec2: DescribeClientVpnEndpoints
	ec2: DescribeClientVpnRoutes
	ec2: DescribeClientVpnTargetNetworks
	ec2: DescribeCoipPools
	ec2: DescribeConversionTasks

Prefisso del servizio	Azioni
	ec2: DescribeCustomerGateways
	ec2: DescribeDeclarativePoliciesReports
	ec2: DescribeDhcpOptions
	ec2: DescribeEgressOnlyInternetGateways
	ec2: DescribeElasticGpus
	ec2: DescribeExportImageTasks
	ec2: DescribeExportTasks
	ec2: DescribeFastLaunchImages
	ec2: DescribeFastSnapshotRestores
	ec2: DescribeFleetHistory
	ec2: DescribeFleetInstances
	ec2: DescribeFleets
	ec2: DescribeFlowLogs
	ec2: DescribeFpgaImageAttribute
	ec2: DescribeFpgaImages
	ec2: DescribeHostReservationOfferings
	ec2: DescribeHostReservations
	ec2: DescribeHosts
	ec2: DescribeIamInstanceProfileAssociations
	ec2: DescribeIdFormat
	ec2: DescribeIdentityIdFormat

Prefisso del servizio	Azioni
	ec2: DescribeImageAttribute
	ec2: DescribeImageReferences
	ec2: DescribeImageUsageReportEntries
	ec2: DescribeImageUsageReports
	ec2: DescribeImportImageTasks
	ec2: DescribeImportSnapshotTasks
	ec2: DescribeInstanceConnectEndpoints
	ec2: DescribeInstanceCreditSpecifications
	ec2: DescribeInstanceEventNotificationAttributes
	ec2: DescribeInstanceEventWindows
	ec2: DescribeInstanceImageMetadata
	ec2: DescribeInstanceSqlHaHistoryStates
	ec2: DescribeInstanceSqlHaStates
	ec2: DescribeInstanceTopology
	ec2: DescribeInstanceTypes
	ec2: DescribeInternetGateways
	ec2: DescribeIpamByoasn
	ec2: DescribeIpamExternalResourceVerificationTokens
	ec2: DescribeIpamPolicies
	ec2: DescribeIpamPoolAllocations
	ec2: DescribeIpamPools

Prefisso del servizio	Azioni
	ec2: DescribeIpamPrefixListResolverTargets
	ec2: DescribeIpamPrefixListResolvers
	ec2: DescribeIpamResourceDiscoveries
	ec2: DescribeIpamResourceDiscoveryAssociations
	ec2: DescribeIpamScopes
	ec2: DescribeIpams
	ec2: DescribeIpv6Pools
	ec2: DescribeKeyPairs
	ec2: DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations
	ec2: DescribeLocalGatewayRouteTableVpcAssociations
	ec2: DescribeLocalGatewayRouteTables
	ec2: DescribeLocalGatewayVirtualInterfaceGroups
	ec2: DescribeLocalGatewayVirtualInterfaces
	ec2: DescribeLocalGateways
	ec2: DescribeLockedSnapshots
	ec2: DescribeMacHosts
	ec2: DescribeMacModificationTasks
	ec2: DescribeManagedPrefixLists
	ec2: DescribeMovingAddresses
	ec2: DescribeNatGateways

Prefisso del servizio	Azioni
	ec2: DescribeNetworkAcls
	ec2: DescribeNetworkInsightsAccessScopeAnalyses
	ec2: DescribeNetworkInsightsAccessScopes
	ec2: DescribeNetworkInsightsAnalyses
	ec2: DescribeNetworkInsightsPaths
	ec2: DescribeNetworkInterfaceAttribute
	ec2: DescribeNetworkInterfacePermissions
	ec2: DescribeNetworkInterfaces
	ec2: DescribeOutpostLags
	ec2: DescribePlacementGroups
	ec2: DescribePrefixLists
	ec2: DescribePrincipalIdFormat
	ec2: DescribePublicIpv4Pools
	ec2: DescribeRegions
	ec2: DescribeReplaceRootVolumeTasks
	ec2: DescribeReservedInstances
	ec2: DescribeReservedInstancesListings
	ec2: DescribeReservedInstancesModifications
	ec2: DescribeReservedInstancesOfferings
	ec2: DescribeRouteServerEndpoints
	ec2: DescribeRouteServerPeers

Prefisso del servizio	Azioni
	ec2: DescribeRouteServers
	ec2: DescribeRouteTables
	ec2: DescribeScheduledInstanceAvailability
	ec2: DescribeScheduledInstances
	ec2: DescribeSecondaryInterfaces
	ec2: DescribeSecondaryNetworks
	ec2: DescribeSecondarySubnets
	ec2: DescribeSecurityGroupReferences
	ec2: DescribeSecurityGroupRules
	ec2: DescribeSecurityGroupVpcAssociations
	ec2: DescribeSecurityGroups
	ec2: DescribeServiceLinkVirtualInterfaces
	ec2: DescribeSnapshotAttribute
	ec2: DescribeSnapshotTierStatus
	ec2: DescribeSpotDatafeedSubscription
	ec2: DescribeSpotFleetInstances
	ec2: DescribeSpotFleetRequestHistory
	ec2: DescribeSpotFleetRequests
	ec2: DescribeSpotInstanceRequests
	ec2: DescribeSpotPriceHistory
	ec2: DescribeStaleSecurityGroups

Prefisso del servizio	Azioni
	ec2: DescribeStoreImageTasks
	ec2: DescribeTrafficMirrorFilterRules
	ec2: DescribeTrafficMirrorFilters
	ec2: DescribeTrafficMirrorSessions
	ec2: DescribeTrafficMirrorTargets
	ec2: DescribeTransitGatewayAttachments
	ec2: DescribeTransitGatewayConnectPeers
	ec2: DescribeTransitGatewayConnects
	ec2: DescribeTransitGatewayMeteringPolicies
	ec2: DescribeTransitGatewayMulticastDomains
	ec2: DescribeTransitGatewayPeeringAttachments
	ec2: DescribeTransitGatewayPolicyTables
	ec2: DescribeTransitGatewayRouteTableAnnouncements
	ec2: DescribeTransitGatewayRouteTables
	ec2: DescribeTransitGatewayVpcAttachments
	ec2: DescribeTransitGateways
	ec2: DescribeTrunkInterfaceAssociations
	ec2: DescribeVerifiedAccessEndpoints
	ec2: DescribeVerifiedAccessGroups
	ec2: DescribeVerifiedAccessInstanceLoggingConfigurations
	ec2: DescribeVerifiedAccessInstances

Prefisso del servizio	Azioni
	ec2: DescribeVerifiedAccessTrustProviders
	ec2: DescribeVolumeAttribute
	ec2: DescribeVolumeStatus
	ec2: DescribeVolumes
	ec2: DescribeVolumesModifications
	ec2: DescribeVpcAttribute
	ec2: DescribeVpcBlockPublicAccessExclusions
	ec2: DescribeVpcBlockPublicAccessOptions
	ec2: DescribeVpcClassicLink
	ec2: DescribeVpcClassicLinkDnsSupport
	ec2: DescribeVpcEncryptionControls
	ec2: DescribeVpcEndpointAssociations
	ec2: DescribeVpcEndpointConnectionNotifications
	ec2: DescribeVpcEndpointConnections
	ec2: DescribeVpcEndpointServiceConfigurations
	ec2: DescribeVpcEndpointServicePermissions
	ec2: DescribeVpcEndpointServices
	ec2: DescribeVpcEndpoints
	ec2: DescribeVpcPeeringConnections
	ec2: DescribeVpcs
	ec2: DescribeVpnConcentrators

Prefisso del servizio	Azioni
	ec2: DescribeVpnConnections
	ec2: DescribeVpnGateways
	ec2: DetachClassicLinkVpc
	ec2: DetachImageWatermark
	ec2: DetachInternetGateway
	ec2: DetachNetworkInterface
	ec2: DetachVerifiedAccessTrustProvider
	ec2: DetachVolume
	ec2: DetachVpnGateway
	ec2: DisableAddressTransfer
	ec2: DisableAllowedImagesSettings
	ec2: DisableAwsNetworkPerformanceMetricSubscription
	ec2: DisableCapacityManager
	ec2: DisableEbsEncryptionByDefault
	ec2: DisableFastLaunch
	ec2: DisableFastSnapshotRestores
	ec2: DisableImage
	ec2: DisableImageBlockPublicAccess
	ec2: DisableImageDeprecation
	ec2: DisableImageDeregistrationProtection
	ec2: DisableInstanceSqlHaStandbyDetections

Prefisso del servizio	Azioni
	ec2: DisableIpamOrganizationAdminAccount
	ec2: DisableIpamPolicy
	ec2: DisableRouteServerPropagation
	ec2: DisableSerialConsoleAccess
	ec2: DisableSnapshotBlockPublicAccess
	ec2: DisableTransitGatewayRouteTablePropagation
	ec2: DisableVgwRoutePropagation
	ec2: DisableVpcClassicLink
	ec2: DisableVpcClassicLinkDnsSupport
	ec2: DisassociateAddress
	ec2: DisassociateCapacityReservationBillingOwner
	ec2: DisassociateClientVpnTargetNetwork
	ec2: DisassociateEnclaveCertificateIamRole
	ec2: DisassociateIamInstanceProfile
	ec2: DisassociateInstanceEventWindow
	ec2: DisassociateIpamByoasn
	ec2: DisassociateIpamResourceDiscovery
	ec2: DisassociateNatGatewayAddress
	ec2: DisassociateRouteServer
	ec2: DisassociateRouteTable
	ec2: DisassociateSecurityGroupVpc

Prefisso del servizio	Azioni
	ec2: DisassociateSubnetCidrBlock
	ec2: DisassociateTransitGatewayMulticastDomain
	ec2: DisassociateTransitGatewayPolicyTable
	ec2: DisassociateTransitGatewayRouteTable
	ec2: DisassociateTrunkInterface
	ec2: DisassociateVpcCidrBlock
	ec2: EnableAddressTransfer
	ec2: EnableAllowedImagesSettings
	ec2: EnableAwsNetworkPerformanceMetricSubscription
	ec2: EnableCapacityManager
	ec2: EnableEbsEncryptionByDefault
	ec2: EnableFastLaunch
	ec2: EnableFastSnapshotRestores
	ec2: EnableImage
	ec2: EnableImageBlockPublicAccess
	ec2: EnableImageDeprecation
	ec2: EnableImageDeregistrationProtection
	ec2: EnableInstanceSqlHaStandbyDetections
	ec2: EnableIpamOrganizationAdminAccount
	ec2: EnableIpamPolicy
	ec2: EnableReachabilityAnalyzerOrganizationSharing

Prefisso del servizio	Azioni
	ec2: EnableRouteServerPropagation
	ec2: EnableSerialConsoleAccess
	ec2: EnableSnapshotBlockPublicAccess
	ec2: EnableTransitGatewayRouteTablePropagation
	ec2: EnableVgwRoutePropagation
	ec2: IO EnableVolume
	ec2: EnableVpcClassicLink
	ec2: EnableVpcClassicLinkDnsSupport
	ec2: ExportClientVpnClientCertificateRevocationList
	ec2: ExportClientVpnClientConfiguration
	ec2: ExportImage
	ec2: ExportTransitGatewayRoutes
	ec2: ExportVerifiedAccessInstanceClientConfiguration
	ec2: GetActiveVpnTunnelStatus
	ec2: GetAllowedImagesSettings
	ec2: GetAssociatedEnclaveCertificateIamRoles
	ec2: GetAssociatedIpv6PoolCidrs
	ec2: GetAwsNetworkPerformanceData
	ec2: GetCapacityManagerAttributes
	ec2: GetCapacityManagerMetricData
	ec2: GetCapacityManagerMetricDimensions

Prefisso del servizio	Azioni
	ec2: GetCapacityReservationUsage
	ec2: GetCoipPoolUsage
	ec2: GetConsoleOutput
	ec2: GetConsoleScreenshot
	ec2: GetDeclarativePoliciesReportSummary
	ec2: GetDefaultCreditSpecification
	ec2: GetEbsDefaultKmsKeyId
	ec2: GetEbsEncryptionByDefault
	ec2: GetEnabledIpamPolicy
	ec2: GetFlowLogsIntegrationTemplate
	ec2: GetGroupsForCapacityReservation
	ec2: GetHostReservationPurchasePreview
	ec2: GetImageAncestry
	ec2: GetImageBlockPublicAccessState
	ec2: GetInstanceMetadataDefaults
	ec2: GetInstanceTpmEkPub
	ec2: GetInstanceTypesFromInstanceRequirements
	ec2: GetInstanceUefiData
	ec2: GetIpamAddressHistory
	ec2: GetIpamDiscoveredAccounts
	ec2: GetIpamDiscoveredPublicAddresses

Prefisso del servizio	Azioni
	ec2: GetIpamDiscoveredResourceCidrs
	ec2: GetIpamPolicyAllocationRules
	ec2: GetIpamPolicyOrganizationTargets
	ec2: GetIpamPoolAllocations
	ec2: GetIpamPoolCidrs
	ec2: GetIpamPrefixListResolverRules
	ec2: GetIpamPrefixListResolverVersionEntries
	ec2: GetIpamPrefixListResolverVersions
	ec2: GetIpamResourceCidrs
	ec2: GetLaunchTemplateData
	ec2: GetManagedPrefixListAssociations
	ec2: GetManagedPrefixListEntries
	ec2: GetManagedResourceVisibility
	ec2: GetNetworkInsightsAccessScopeAnalysisFindings
	ec2: GetNetworkInsightsAccessScopeContent
	ec2: GetPasswordData
	ec2: GetReservedInstancesExchangeQuote
	ec2: GetRouteServerAssociations
	ec2: GetRouteServerPropagations
	ec2: GetRouteServerRoutingDatabase
	ec2: GetSecurityGroupsForVpc

Prefisso del servizio	Azioni
	ec2: GetSerialConsoleAccessStatus
	ec2: GetSnapshotBlockPublicAccessState
	ec2: GetSpotPlacementScores
	ec2: GetSubnetCidrReservations
	ec2: GetTransitGatewayAttachmentPropagations
	ec2: GetTransitGatewayMeteringPolicyEntries
	ec2: GetTransitGatewayMulticastDomainAssociations
	ec2: GetTransitGatewayPolicyTableAssociations
	ec2: GetTransitGatewayPolicyTableEntries
	ec2: GetTransitGatewayPrefixListReferences
	ec2: GetTransitGatewayRouteTableAssociations
	ec2: GetTransitGatewayRouteTablePropagations
	ec2: GetVerifiedAccessEndpointPolicy
	ec2: GetVerifiedAccessEndpointTargets
	ec2: GetVerifiedAccessGroupPolicy
	ec2: GetVpcResourcesBlockingEncryptionEnforcement
	ec2: GetVpnConnectionDeviceSampleConfiguration
	ec2: GetVpnConnectionDeviceTypes
	ec2: GetVpnTunnelReplacementStatus
	ec2: ImportClientVpnClientCertificateRevocationList
	ec2: ImportImage

Prefisso del servizio	Azioni
	ec2: ImportInstance
	ec2: ImportKeyPair
	ec2: ImportSnapshot
	ec2: ImportVolume
	ec2: ListImagesInRecycleBin
	ec2: ListSnapshotsInRecycleBin
	ec2: ListVolumesInRecycleBin
	ec2: LockSnapshot
	ec2: ModifyAddressAttribute
	ec2: ModifyAvailabilityZoneGroup
	ec2: ModifyCapacityReservation
	ec2: ModifyCapacityReservationFleet
	ec2: ModifyClientVpnEndpoint
	ec2: ModifyDefaultCreditSpecification
	ec2: ModifyEbsDefaultKmsKeyId
	ec2: ModifyFleet
	ec2: ModifyFpgaImageAttribute
	ec2: ModifyHosts
	ec2: ModifyIdFormat
	ec2: ModifyIdentityIdFormat
	ec2: ModifyImageAttribute

Prefisso del servizio	Azioni
	ec2: ModifyInstanceAttribute
	ec2: ModifyInstanceCapacityReservationAttributes
	ec2: ModifyInstanceConnectEndpoint
	ec2: ModifyInstanceCpuOptions
	ec2: ModifyInstanceCreditSpecification
	ec2: ModifyInstanceEventStartTime
	ec2: ModifyInstanceEventWindow
	ec2: ModifyInstanceMaintenanceOptions
	ec2: ModifyInstanceMetadataDefaults
	ec2: ModifyInstanceMetadataOptions
	ec2: ModifyInstanceNetworkPerformanceOptions
	ec2: ModifyInstancePlacement
	ec2: ModifyIpam
	ec2: ModifyIpamPolicyAllocationRules
	ec2: ModifyIpamPool
	ec2: ModifyIpamPoolAllocation
	ec2: ModifyIpamPrefixListResolver
	ec2: ModifyIpamPrefixListResolverTarget
	ec2: ModifyIpamResourceCidr
	ec2: ModifyIpamResourceDiscovery
	ec2: ModifyIpamScope

Prefisso del servizio	Azioni
	ec2: ModifyLaunchTemplate
	ec2: ModifyLocalGatewayRoute
	ec2: ModifyManagedPrefixList
	ec2: ModifyManagedResourceVisibility
	ec2: ModifyNetworkInterfaceAttribute
	ec2: ModifyPrivateDnsNameOptions
	ec2: ModifyPublicIpDnsNameOptions
	ec2: ModifyReservedInstances
	ec2: ModifyRouteServer
	ec2: ModifySecurityGroupRules
	ec2: ModifySnapshotAttribute
	ec2: ModifySnapshotTier
	ec2: ModifySpotFleetRequest
	ec2: ModifySubnetAttribute
	ec2: ModifyTrafficMirrorFilterNetworkServices
	ec2: ModifyTrafficMirrorFilterRule
	ec2: ModifyTrafficMirrorSession
	ec2: ModifyTransitGateway
	ec2: ModifyTransitGatewayMeteringPolicy
	ec2: ModifyTransitGatewayPrefixListReference
	ec2: ModifyTransitGatewayVpcAttachment

Prefisso del servizio	Azioni
	ec2: ModifyVerifiedAccessEndpoint
	ec2: ModifyVerifiedAccessEndpointPolicy
	ec2: ModifyVerifiedAccessGroup
	ec2: ModifyVerifiedAccessGroupPolicy
	ec2: ModifyVerifiedAccessInstance
	ec2: ModifyVerifiedAccessInstanceLoggingConfiguration
	ec2: ModifyVerifiedAccessTrustProvider
	ec2: ModifyVolume
	ec2: ModifyVolumeAttribute
	ec2: ModifyVpcAttribute
	ec2: ModifyVpcBlockPublicAccessExclusion
	ec2: ModifyVpcBlockPublicAccessOptions
	ec2: ModifyVpcEncryptionControl
	ec2: ModifyVpcEndpoint
	ec2: ModifyVpcEndpointConnectionNotification
	ec2: ModifyVpcEndpointServiceConfiguration
	ec2: ModifyVpcEndpointServicePayerResponsibility
	ec2: ModifyVpcEndpointServicePermissions
	ec2: ModifyVpcPeeringConnectionOptions
	ec2: ModifyVpcTenancy
	ec2: ModifyVpnConnection

Prefisso del servizio	Azioni
	ec2: ModifyVpnConnectionOptions
	ec2: ModifyVpnTunnelCertificate
	ec2: ModifyVpnTunnelOptions
	ec2: MonitorInstances
	ec2: MoveAddressToVpc
	ec2: MoveByoipCidrToIpam
	ec2: MoveCapacityReservationInstances
	ec2: ProvisionByoipCidr
	ec2: ProvisionIpamByoasn
	ec2: ProvisionIpamPoolCidr
	ec2: ProvisionPublicIpv4PoolCidr
	ec2: PurchaseCapacityBlockExtension
	ec2: PurchaseHostReservation
	ec2: PurchaseReservedInstancesOffering
	ec2: PurchaseScheduledInstances
	ec2: RebootInstances
	ec2: RegisterImage
	ec2: RegisterInstanceEventNotificationAttributes
	ec2: RegisterTransitGatewayMulticastGroupMembers
	ec2: RegisterTransitGatewayMulticastGroupSources
	ec2: RejectCapacityReservationBillingOwnership

Prefisso del servizio	Azioni
	ec2: RejectTransitGatewayClientVpnAttachment
	ec2: RejectTransitGatewayMulticastDomainAssociations
	ec2: RejectTransitGatewayPeeringAttachment
	ec2: RejectTransitGatewayVpcAttachment
	ec2: RejectVpcEndpointConnections
	ec2: RejectVpcPeeringConnection
	ec2: ReleaseAddress
	ec2: ReleaseHosts
	ec2: ReleaseIpamPoolAllocation
	ec2: ReplaceIamInstanceProfileAssociation
	ec2: ReplaceImageCriteriaInAllowedImagesSettings
	ec2: ReplaceNetworkAclAssociation
	ec2: ReplaceNetworkAclEntry
	ec2: ReplaceRoute
	ec2: ReplaceRouteTableAssociation
	ec2: ReplaceTransitGatewayRoute
	ec2: ReplaceVpnTunnel
	ec2: ReportInstanceStatus
	ec2: RequestSpotFleet
	ec2: RequestSpotInstances
	ec2: ResetAddressAttribute

Prefisso del servizio	Azioni
	ec2: ResetEbsDefaultKmsKeyId
	ec2: ResetFpgaImageAttribute
	ec2: ResetImageAttribute
	ec2: ResetInstanceAttribute
	ec2: ResetNetworkInterfaceAttribute
	ec2: ResetSnapshotAttribute
	ec2: RestoreAddressToClassic
	ec2: RestoreImageFromRecycleBin
	ec2: RestoreManagedPrefixListVersion
	ec2: RestoreSnapshotFromRecycleBin
	ec2: RestoreSnapshotTier
	ec2: RestoreVolumeFromRecycleBin
	ec2: RevokeClientVpnIngress
	ec2: RevokeSecurityGroupEgress
	ec2: RevokeSecurityGroupIngress
	ec2: RunInstances
	ec2: RunScheduledInstances
	ec2: SearchLocalGatewayRoutes
	ec2: SearchTransitGatewayMulticastGroups
	ec2: SearchTransitGatewayRoutes
	ec2: SendDiagnosticInterrupt

Prefisso del servizio	Azioni
	ec2: StartDeclarativePoliciesReport
	ec2: StartInstances
	ec2: StartNetworkInsightsAccessScopeAnalysis
	ec2: StartNetworkInsightsAnalysis
	ec2: StartVpcEndpointServicePrivateDnsVerification
	ec2: TerminateClientVpnConnections
	ec2: UnassignIpv6Addresses
	ec2: UnassignPrivateIpAddresses
	ec2: UnassignPrivateNatGatewayAddress
	ec2: UnlockSnapshot
	ec2: UnmonitorInstances
	ec2: UpdateCapacityManagerOrganizationsAccess
	ec2: UpdateInterruptibleCapacityReservationAllocation
	ec2: UpdateSecurityGroupRuleDescriptionsEgress
	ec2: UpdateSecurityGroupRuleDescriptionsIngress
	ec2: WithdrawByoipCidr

Prefisso del servizio	Azioni
ecr	ecr: BatchCheckLayerAvailability ecr: BatchDeleteImage ecr: BatchGetImage ecr: BatchGetRepositoryScanningConfiguration ecr: CompleteLayerUpload ecr: CreatePullThroughCacheRule ecr: CreateRepositoryCreationTemplate ecr: DeleteLifecyclePolicy ecr: DeletePullThroughCacheRule ecr: DeleteRegistryPolicy ecr: DeleteRepository ecr: DeleteRepositoryCreationTemplate ecr: DeleteRepositoryPolicy ecr: DeleteSigningConfiguration ecr: DescribeImageReplicationStatus ecr: DescribeImageScanFindings ecr: DescribeImages ecr: DescribePullThroughCacheRules ecr: DescribeRegistry ecr: DescribeRepositories ecr: DescribeRepositoryCreationTemplates

Prefisso del servizio	Azioni
	ecr: GetAccountSetting ecr: GetAuthorizationToken ecr: GetDownloadUrlForLayer ecr: GetLifecyclePolicy ecr: GetLifecyclePolicyPreview ecr: GetRegistryPolicy ecr: GetRegistryScanningConfiguration ecr: GetRepositoryPolicy ecr: GetSigningConfiguration ecr: InitiateLayerUpload ecr: ListImages ecr: ListPullTimeUpdateExclusions ecr: PutAccountSetting ecr: PutImage ecr: PutImageScanningConfiguration ecr: PutRegistryPolicy ecr: PutRegistryScanningConfiguration ecr: PutReplicationConfiguration ecr: StartImageScan ecr: StartLifecyclePolicyPreview ecr: UpdatePullThroughCacheRule

Prefisso del servizio	Azioni
	ecr: UpdateRepositoryCreationTemplate ecr: UploadLayerPart ecr: ValidatePullThroughCacheRule

Prefisso del servizio	Azioni
ecr-public	ecr-pubblico: BatchCheckLayerAvailability
	ecr-pubblico: BatchDeleteImage
	ecr-pubblico: CompleteLayerUpload
	ecr-pubblico: CreateRepository
	ecr-pubblico: DeleteRepository
	ecr-pubblico: DeleteRepositoryPolicy
	ecr-pubblico: DescribeImages
	ecr-pubblico: DescribeRegistries
	ecr-pubblico: DescribeRepositories
	ecr-pubblico: GetAuthorizationToken
	ecr-pubblico: GetRegistryCatalogData
	ecr-pubblico: GetRepositoryCatalogData
	ecr-pubblico: GetRepositoryPolicy
	ecr-pubblico: InitiateLayerUpload
	ecr-pubblico: PutImage
	ecr-pubblico: PutRegistryCatalogData
	ecr-pubblico: PutRepositoryCatalogData
	ecr-pubblico: SetRepositoryPolicy
	ecr-pubblico: UploadLayerPart

Prefisso del servizio	Azioni
ecs	ecs: ContinueServiceDeployment ecs: CreateCapacityProvider ecs: CreateCluster ecs: CreateDaemon ecs: CreateService ecs: CreateTaskSet ecs: DeleteAccountSetting ecs: DeleteAttributes ecs: DeleteCapacityProvider ecs: DeleteCluster ecs: DeleteDaemon ecs: DeleteDaemonTaskDefinition ecs: DeleteExpressGatewayService ecs: DeleteService ecs: DeleteTaskDefinitions ecs: DeleteTaskSet ecs: DeregisterContainerInstance ecs: DeregisterTaskDefinition ecs: DescribeCapacityProviders ecs: DescribeClusters ecs: DescribeContainerInstances

Prefisso del servizio	Azioni
	ecs: DescribeDaemon ecs: DescribeDaemonDeployments ecs: DescribeDaemonRevisions ecs: DescribeDaemonTaskDefinition ecs: DescribeExpressGatewayService ecs: DescribeServiceDeployments ecs: DescribeServiceRevisions ecs: DescribeServices ecs: DescribeTaskDefinition ecs: DescribeTaskSets ecs: DescribeTasks ecs: DiscoverPollEndpoint ecs: ExecuteCommand ecs: GetTaskProtection ecs: ListAccountSettings ecs: ListAttributes ecs: ListClusters ecs: ListContainerInstances ecs: ListDaemonDeployments ecs: ListDaemonTaskDefinitions ecs: ListDaemons

Prefisso del servizio	Azioni
	ecs: ListServiceDeployments ecs: ListServices ecs: ListServicesByNamespace ecs: ListTaskDefinitionFamilies ecs: ListTaskDefinitions ecs: ListTasks ecs: PutAccountSetting ecs: PutAccountSettingDefault ecs: PutAttributes ecs: PutClusterCapacityProviders ecs: RegisterContainerInstance ecs: RegisterDaemonTaskDefinition ecs: RunTask ecs: StartTask ecs: StopServiceDeployment ecs: StopTask ecs: SubmitAttachmentStateChanges ecs: SubmitContainerStateChange ecs: SubmitTaskStateChange ecs: UpdateCapacityProvider ecs: UpdateCluster

Prefisso del servizio	Azioni
	ecs: UpdateClusterSettings ecs: UpdateContainerAgent ecs: UpdateContainerInstancesState ecs: UpdateDaemon ecs: UpdateExpressGatewayService ecs: UpdateService ecs: UpdateServicePrimaryTaskSet ecs: UpdateTaskProtection ecs: UpdateTaskSet

Prefisso del servizio	Azioni
eks	ex: AssociateAccessPolicy
	ex: AssociateEncryptionConfig
	ex: AssociateIdentityProviderConfig
	ex: CancelUpdate
	ex: CreateAccessEntry
	ex: CreateAddon
	ex: CreateCluster
	ex: CreateEksAnywhereSubscription
	ex: CreateFargateProfile
	ex: CreateNodegroup
	ex: DeleteAccessEntry
	ex: DeleteAddon
	ex: DeleteCapability
	ex: DeleteCluster
	ex: DeleteEksAnywhereSubscription
	ex: DeleteFargateProfile
	ex: DeleteNodegroup
	ex: DeletePodIdentityAssociation
	ex: DeregisterCluster
	ex: DescribeAccessEntry
	ex: DescribeAddon

Prefisso del servizio	Azioni
	ex: DescribeAddonConfiguration
	ex: DescribeAddonVersions
	ex: DescribeCapability
	ex: DescribeCluster
	ex: DescribeClusterVersions
	ex: DescribeEksAnywhereSubscription
	ex: DescribeFargateProfile
	ex: DescribeIdentityProviderConfig
	ex: DescribeInsight
	ex: DescribeInsightsRefresh
	ex: DescribeNodegroup
	ex: DescribePodIdentityAssociation
	ex: DescribeUpdate
	ex: DisassociateAccessPolicy
	ex: DisassociateIdentityProviderConfig
	ex: ListAccessEntries
	ex: ListAccessPolicies
	ex: ListAddons
	ex: ListAssociatedAccessPolicies
	ex: ListCapabilities
	ex: ListClusters

Prefisso del servizio	Azioni
	ex: ListEksAnywhereSubscriptions
	ex: ListFargateProfiles
	ex: ListIdentityProviderConfigs
	ex: ListInsights
	ex: ListNodegroups
	ex: ListPodIdentityAssociations
	ex: ListUpdates
	ex: RegisterCluster
	ex: StartInsightsRefresh
	ex: UpdateAccessEntry
	ex: UpdateAddon
	ex: UpdateCapability
	ex: UpdateClusterConfig
	ex: UpdateClusterVersion
	ex: UpdateEksAnywhereSubscription
	ex: UpdateNodegroupConfig
	ex: UpdateNodegroupVersion
	ex: UpdatePodIdentityAssociation

Prefisso del servizio	Azioni
elasticache	dolore elastico: AuthorizeCacheSecurityGroupIngress dolore elastico: BatchApplyUpdateAction dolore elastico: BatchStopUpdateAction dolore elastico: CompleteMigration dolore elastico: CopyServerlessCacheSnapshot dolore elastico: CopySnapshot dolore elastico: CreateCacheCluster dolore elastico: CreateCacheParameterGroup dolore elastico: CreateCacheSecurityGroup dolore elastico: CreateCacheSubnetGroup dolore elastico: CreateGlobalReplicationGroup dolore elastico: CreateReplicationGroup dolore elastico: CreateServerlessCache dolore elastico: CreateServerlessCacheSnapshot dolore elastico: CreateSnapshot dolore elastico: CreateUser dolore elastico: CreateUserGroup dolore elastico: DecreaseNodeGroupsInGlobalReplicationGroup dolore elastico: DecreaseReplicaCount dolore elastico: DeleteCacheCluster dolore elastico: DeleteCacheParameterGroup

Prefisso del servizio	Azioni
	dolore elastico: DeleteCacheSecurityGroup dolore elastico: DeleteCacheSubnetGroup dolore elastico: DeleteGlobalReplicationGroup dolore elastico: DeleteReplicationGroup dolore elastico: DeleteServerlessCache dolore elastico: DeleteServerlessCacheSnapshot dolore elastico: DeleteSnapshot dolore elastico: DeleteUser dolore elastico: DeleteUserGroup dolore elastico: DescribeCacheClusters dolore elastico: DescribeCacheEngineVersions dolore elastico: DescribeCacheParameterGroups dolore elastico: DescribeCacheParameters dolore elastico: DescribeCacheSecurityGroups dolore elastico: DescribeCacheSubnetGroups dolore elastico: DescribeEngineDefaultParameters dolore elastico: DescribeEvents dolore elastico: DescribeGlobalReplicationGroups dolore elastico: DescribeReplicationGroups dolore elastico: DescribeReservedCacheNodes dolore elastico: DescribeReservedCacheNodesOfferings

Prefisso del servizio	Azioni
	dolore elastico: DescribeServerlessCacheSnapshots dolore elastico: DescribeServerlessCaches dolore elastico: DescribeServiceUpdates dolore elastico: DescribeSnapshots dolore elastico: DescribeUpdateActions dolore elastico: DescribeUserGroups dolore elastico: DescribeUsers dolore elastico: DisassociateGlobalReplicationGroup dolore elastico: ExportServerlessCacheSnapshot dolore elastico: FailoverGlobalReplicationGroup dolore elastico: IncreaseNodeGroupsInGlobalReplicationGroup dolore elastico: IncreaseReplicaCount dolore elastico: ListAllowedNodeTypeModifications dolore elastico: ModifyCacheCluster dolore elastico: ModifyCacheParameterGroup dolore elastico: ModifyCacheSubnetGroup dolore elastico: ModifyGlobalReplicationGroup dolore elastico: ModifyReplicationGroup dolore elastico: ModifyReplicationGroupShardConfiguration dolore elastico: ModifyServerlessCache dolore elastico: ModifyUser

Prefisso del servizio	Azioni
	dolore elastico: ModifyUserGroup dolore elastico: PurchaseReservedCacheNodesOffering dolore elastico: RebalanceSlotsInGlobalReplicationGroup dolore elastico: RebootCacheCluster dolore elastico: ResetCacheParameterGroup dolore elastico: RevokeCacheSecurityGroupIngress dolore elastico: StartMigration dolore elastico: TestFailover dolore elastico: TestMigration

Prefisso del servizio	Azioni
elasticbeanstalk	gambo elastico di fagioli: AbortEnvironmentUpdate
	gambo elastico di fagioli: ApplyEnvironmentManagedAction
	gambo elastico di fagioli: AssociateEnvironmentOperationsRole
	elasticbeanstalk:CheckDNSAvailability
	gambo elastico di fagioli: ComposeEnvironments
	gambo elastico di fagioli: CreateApplication
	gambo elastico di fagioli: CreateApplicationVersion
	gambo elastico di fagioli: CreateConfigurationTemplate
	gambo elastico di fagioli: CreateEnvironment
	gambo elastico di fagioli: CreatePlatformVersion
	gambo elastico di fagioli: CreateStorageLocation
	gambo elastico di fagioli: DeleteApplication
	gambo elastico di fagioli: DeleteApplicationVersion
	gambo elastico di fagioli: DeleteConfigurationTemplate
	gambo elastico di fagioli: DeleteEnvironmentConfiguration
	gambo elastico di fagioli: DeletePlatformVersion
	gambo elastico di fagioli: DescribeAccountAttributes
	gambo elastico di fagioli: DescribeApplicationVersions
	gambo elastico di fagioli: DescribeApplications
	gambo elastico di fagioli: DescribeConfigurationOptions
	gambo elastico di fagioli: DescribeConfigurationSettings

Prefisso del servizio	Azioni
	gambo elastico di fagioli: DescribeEnvironmentHealth
	gambo elastico di fagioli: DescribeEnvironmentManagedActionHistory
	gambo elastico di fagioli: DescribeEnvironmentManagedActions
	gambo elastico di fagioli: DescribeEnvironmentResources
	gambo elastico di fagioli: DescribeEnvironments
	gambo elastico di fagioli: DescribeEvents
	gambo elastico di fagioli: DescribeInstancesHealth
	gambo elastico di fagioli: DescribePlatformVersion
	gambo elastico di fagioli: DisassociateEnvironmentOperationsRole
	gambo elastico di fagioli: ListAvailableSolutionStacks
	gambo elastico di fagioli: ListPlatformBranches
	gambo elastico di fagioli: ListPlatformVersions
	gambo elastico di fagioli: RebuildEnvironment
	gambo elastico di fagioli: RequestEnvironmentInfo
	gambo elastico di fagioli: RestartAppServer
	gambo elastico di fagioli: RetrieveEnvironmentInfo
	gambo elastico di fagioli: SwapEnvironmentCNAMEs
	gambo elastico di fagioli: TerminateEnvironment
	gambo elastico di fagioli: UpdateApplication
	gambo elastico di fagioli: UpdateApplicationResourceLifecycle

Prefisso del servizio	Azioni
	gambo elastico di fagioli: UpdateApplicationVersion
	gambo elastico di fagioli: UpdateConfigurationTemplate
	gambo elastico di fagioli: UpdateEnvironment
	gambo elastico di fagioli: ValidateConfigurationSettings

Prefisso del servizio	Azioni
elasticfilesystem	file system elastico: CreateAccessPoint filesystem elastico: CreateFileSystem filesystem elastico: CreateMountTarget filesystem elastico: CreateReplicationConfiguration filesystem elastico: DeleteAccessPoint filesystem elastico: DeleteFileSystem filesystem elastico: DeleteFileSystemPolicy filesystem elastico: DeleteMountTarget filesystem elastico: DeleteReplicationConfiguration filesystem elastico: DescribeAccessPoints filesystem elastico: DescribeAccountPreferences filesystem elastico: DescribeBackupPolicy filesystem elastico: DescribeFileSystemPolicy filesystem elastico: DescribeFileSystems filesystem elastico: DescribeLifecycleConfiguration filesystem elastico: DescribeMountTargetSecurityGroups filesystem elastico: DescribeMountTargets filesystem elastico: DescribeReplicationConfigurations filesystem elastico: ModifyMountTargetSecurityGroups filesystem elastico: PutAccountPreferences filesystem elastico: PutBackupPolicy

Prefisso del servizio	Azioni
	filesystem elastico: PutFileSystemPolicy filesystem elastico: PutLifecycleConfiguration filesystem elastico: UpdateFileSystem filesystem elastico: UpdateFileSystemProtection

Prefisso del servizio	Azioni
elasticloadbalancing	bilanciamento elastico del carico: AddListenerCertificates
	bilanciamento elastico del carico: AddTrustStoreRevocations
	bilanciamento elastico del carico: ApplySecurityGroupsToLoadBalancer
	bilanciamento elastico del carico: AttachLoadBalancerToSubnets
	bilanciamento elastico del carico: ConfigureHealthCheck
	bilanciamento elastico del carico: CreateAppCookieStickinessPolicy
	bilanciamento elastico del carico: CreateLBCookieStickinessPolicy
	bilanciamento elastico del carico: CreateListener
	bilanciamento elastico del carico: CreateLoadBalancer
	bilanciamento elastico del carico: CreateLoadBalancerListeners
	bilanciamento elastico del carico: CreateLoadBalancerPolicy
	bilanciamento elastico del carico: CreateRule
	bilanciamento elastico del carico: CreateTargetGroup
	bilanciamento elastico del carico: CreateTrustStore
	bilanciamento elastico del carico: DeleteListener
	bilanciamento elastico del carico: DeleteLoadBalancer
	bilanciamento elastico del carico: DeleteLoadBalancerListeners
	bilanciamento elastico del carico: DeleteLoadBalancerPolicy
	bilanciamento elastico del carico: DeleteRule
	bilanciamento elastico del carico: DeleteSharedTrustStoreAssociation

Prefisso del servizio	Azioni
	bilanciamento elastico del carico: DeleteTargetGroup
	bilanciamento elastico del carico: DeleteTrustStore
	bilanciamento elastico del carico: DeregisterInstancesFromLoad Balancer
	bilanciamento elastico del carico: DeregisterTargets
	bilanciamento elastico del carico: DescribeAccountLimits
	bilanciamento elastico del carico: DescribeCapacityReservation
	bilanciamento elastico del carico: DescribeInstanceHealth
	bilanciamento elastico del carico: DescribeListenerAttributes
	bilanciamento elastico del carico: DescribeListenerCertificates
	bilanciamento elastico del carico: DescribeListeners
	bilanciamento elastico del carico: DescribeLoadBalancerAttributes
	bilanciamento elastico del carico: DescribeLoadBalancerPolicies
	bilanciamento elastico del carico: DescribeLoadBalancerPolicyT ypes
	bilanciamento elastico del carico: DescribeLoadBalancers
	bilanciamento elastico del carico: DescribeRules
	elasticloadbalancing:DescribeSSLPolicies
	bilanciamento elastico del carico: DescribeTargetGroupAttributes
	bilanciamento elastico del carico: DescribeTargetGroups
	bilanciamento elastico del carico: DescribeTargetHealth
	bilanciamento elastico del carico: DescribeTrustStoreAssociations

Prefisso del servizio	Azioni
	bilanciamento elastico del carico: DescribeTrustStoreRevocations
	bilanciamento elastico del carico: DescribeTrustStores
	bilanciamento elastico del carico: DetachLoadBalancerFromSubnets
	bilanciamento elastico del carico: DisableAvailabilityZonesForLoadBalancer
	bilanciamento elastico del carico: EnableAvailabilityZonesForLoadBalancer
	bilanciamento elastico del carico: GetResourcePolicy
	bilanciamento elastico del carico: GetTrustStoreCaCertificatesBundle
	bilanciamento elastico del carico: GetTrustStoreRevocationContent
	bilanciamento elastico del carico: ModifyCapacityReservation
	bilanciamento elastico del carico: ModifyIpPools
	bilanciamento elastico del carico: ModifyListener
	bilanciamento elastico del carico: ModifyLoadBalancerAttributes
	bilanciamento elastico del carico: ModifyRule
	bilanciamento elastico del carico: ModifyTargetGroup
	bilanciamento elastico del carico: ModifyTargetGroupAttributes
	bilanciamento elastico del carico: ModifyTrustStore
	bilanciamento elastico del carico: RegisterInstancesWithLoadBalancer
	bilanciamento elastico del carico: RegisterTargets

Prefisso del servizio	Azioni
	bilanciamento elastico del carico: RemoveListenerCertificates
	bilanciamento elastico del carico: RemoveTrustStoreRevocations
	bilanciamento elastico del carico: SetIpAddressType
	bilanciamento elastico del carico: SetLoadBalancerListenerSSLCertificate
	bilanciamento elastico del carico: SetLoadBalancerPoliciesForBackendServer
	bilanciamento elastico del carico: SetLoadBalancerPoliciesOfListener
	bilanciamento elastico del carico: SetRulePriorities
	bilanciamento elastico del carico: SetSecurityGroups
	bilanciamento elastico del carico: SetSubnets

Prefisso del servizio	Azioni
emr-containers	contenitori emr: CancelJobRun contenitori emr: CreateJobTemplate contenitori emr: CreateManagedEndpoint contenitori emr: CreateSecurityConfiguration contenitori emr: CreateVirtualCluster contenitori emr: DeleteJobTemplate contenitori emr: DeleteManagedEndpoint contenitori emr: DeleteSecurityConfiguration contenitori emr: DeleteVirtualCluster contenitori emr: DescribeJobRun contenitori emr: DescribeJobTemplate contenitori emr: DescribeManagedEndpoint contenitori emr: DescribeSecurityConfiguration contenitori emr: DescribeVirtualCluster contenitori emr: GetManagedEndpointSessionCredentials contenitori emr: ListJobRuns contenitori emr: ListJobTemplates contenitori emr: ListManagedEndpoints contenitori emr: ListSecurityConfigurations contenitori emr: ListVirtualClusters contenitori emr: StartJobRun

Prefisso del servizio	Azioni
emr-serverless	emr-senza server: CancelJobRun
	emr-senza server: CreateApplication
	emr-senza server: DeleteApplication
	emr-senza server: GetApplication
	emr-senza server: GetDashboardForJobRun
	emr-senza server: GetJobRun
	emr-senza server: GetResourceDashboard
	emr-senza server: GetSession
	emr-senza server: GetSessionEndpoint
	emr-senza server: ListApplications
	emr-senza server: ListJobRunAttempts
	emr-senza server: ListJobRuns
	emr-senza server: ListSessions
	emr-senza server: StartApplication
	emr-senza server: StartJobRun
	emr-senza server: StopApplication
	emr-senza server: TerminateSession
	emr-senza server: UpdateApplication

Prefisso del servizio	Azioni
es	Sì: AcceptInboundConnection Sì: AcceptInboundCrossClusterSearchConnection Sì: AssociatePackage Sì: AuthorizeVpcEndpointAccess Sì: CancelElasticsearchServiceSoftwareUpdate Sì: CancelServiceSoftwareUpdate Sì: CreateDomain Sì: CreateElasticsearchDomain Sì: CreateIndex Sì: CreateOutboundConnection Sì: CreateOutboundCrossClusterSearchConnection Sì: CreatePackage Sì: CreateVpcEndpoint Sì: DeleteDomain Sì: DeleteElasticsearchDomain Sì: DeleteElasticsearchServiceRole Sì: DeleteInboundConnection Sì: DeleteInboundCrossClusterSearchConnection Sì: DeleteIndex Sì: DeleteOutboundConnection Sì: DeleteOutboundCrossClusterSearchConnection

Prefisso del servizio	Azioni
	Sì: DeletePackage Sì: DeleteVpcEndpoint Sì: DescribeDomain Sì: DescribeDomainAutoTunes Sì: DescribeDomainChangeProgress Sì: DescribeDomainConfig Sì: DescribeDomainHealth Sì: DescribeDomainNodes Sì: DescribeDomains Sì: DescribeDryRunProgress Sì: DescribeElasticsearchDomain Sì: DescribeElasticsearchDomainConfig Sì: DescribeElasticsearchDomains Sì: DescribeElasticsearchInstanceTypeLimits Sì: DescribeInboundConnections Sì: DescribeInboundCrossClusterSearchConnections Sì: DescribeInstanceTypeLimits Sì: DescribeOutboundConnections Sì: DescribeOutboundCrossClusterSearchConnections Sì: DescribePackages Sì: DescribeReservedElasticsearchInstanceOfferings

Prefisso del servizio	Azioni
	Sì: DescribeReservedElasticsearchInstances Sì: DescribeReservedInstanceOfferings Sì: DescribeReservedInstances Sì: DescribeVpcEndpoints Sì: DissociatePackage Sì: DissociatePackages Sì: GetCompatibleElasticsearchVersions Sì: GetCompatibleVersions Sì: GetDataSource Sì: GetDomainMaintenanceStatus Sì: GetPackageVersionHistory Sì: GetUpgradeHistory Sì: GetUpgradeStatus Sì: ListDataSources Sì: ListDomainNames Sì: ListDomainsForPackage Sì: ListElasticsearchInstanceTypes Sì: ListElasticsearchVersions Sì: ListInsights Sì: ListInstanceTypeDetails Sì: ListPackagesForDomain

Prefisso del servizio	Azioni
	Sì: ListScheduledActions Sì: ListVersions Sì: ListVpcEndpointAccess Sì: ListVpcEndpoints Sì: ListVpcEndpointsForDomain Sì: PurchaseReservedElasticsearchInstanceOffering Sì: PurchaseReservedInstanceOffering Sì: RejectInboundConnection Sì: RejectInboundCrossClusterSearchConnection Sì: RevokeVpcEndpointAccess Sì: RollbackServiceSoftwareUpdate Sì: StartDomainMaintenance Sì: StartElasticsearchServiceSoftwareUpdate Sì: StartServiceSoftwareUpdate Sì: UpdateDataSource Sì: UpdateDomainConfig Sì: UpdateElasticsearchDomainConfig Sì: UpdateIndex Sì: UpdatePackage Sì: UpdatePackageScope Sì: UpdateScheduledAction

Prefisso del servizio	Azioni
	Sì: UpdateVpcEndpoint Sì: UpgradeDomain Sì: UpgradeElasticsearchDomain

Prefisso del servizio	Azioni
eventi	eventi: ActivateEventSource
	eventi: CancelReplay
	eventi: CreateApiDestination
	eventi: CreateArchive
	eventi: CreateConnection
	eventi: CreateEndpoint
	eventi: CreateEventBus
	eventi: CreatePartnerEventSource
	eventi: DeactivateEventSource
	eventi: DeauthorizeConnection
	eventi: DeleteApiDestination
	eventi: DeleteArchive
	eventi: DeleteConnection
	eventi: DeleteEndpoint
	eventi: DeleteEventBus
	eventi: DeletePartnerEventSource
	eventi: DeleteRule
	eventi: DescribeApiDestination
	eventi: DescribeArchive
	eventi: DescribeConnection
	eventi: DescribeEndpoint

Prefisso del servizio	Azioni
	eventi: DescribeEventBus
	eventi: DescribeEventSource
	eventi: DescribePartnerEventSource
	eventi: DescribeReplay
	eventi: DescribeRule
	eventi: DisableRule
	eventi: EnableRule
	eventi: ListApiDestinations
	eventi: ListArchives
	eventi: ListConnections
	eventi: ListEndpoints
	eventi: ListEventBuses
	eventi: ListEventSources
	eventi: ListPartnerEventSourceAccounts
	eventi: ListPartnerEventSources
	eventi: ListReplays
	eventi: ListRuleNamesByTarget
	eventi: ListRules
	eventi: ListTargetsByRule
	eventi: PutPermission
	eventi: PutRule

Prefisso del servizio	Azioni
	eventi: PutTargets
	eventi: RemovePermission
	eventi: RemoveTargets
	eventi: StartReplay
	eventi: TestEventPattern
	eventi: UpdateApiDestination
	eventi: UpdateArchive
	eventi: UpdateConnection
	eventi: UpdateEndpoint
	eventi: UpdateEventBus

Prefisso del servizio	Azioni
finspace	finspace: CreateEnvironment spazio interno: CreateKxChangeset spazio interno: CreateKxCluster spazio interno: CreateKxDatabase spazio interno: CreateKxDataview spazio interno: CreateKxEnvironment spazio interno: CreateKxScalingGroup spazio interno: CreateKxUser spazio interno: CreateKxVolume spazio interno: CreateUser spazio interno: DeleteEnvironment spazio interno: DeleteKxCluster spazio interno: DeleteKxClusterNode spazio interno: DeleteKxDatabase spazio interno: DeleteKxDataview spazio interno: DeleteKxEnvironment spazio interno: DeleteKxScalingGroup spazio interno: DeleteKxUser spazio interno: DeleteKxVolume spazio interno: GetEnvironment spazio interno: GetKxChangeset

Prefisso del servizio	Azioni
	spazio interno: GetKxCluster spazio interno: GetKxConnectionString spazio interno: GetKxDatabase spazio interno: GetKxDataview spazio interno: GetKxEnvironment spazio interno: GetKxScalingGroup spazio interno: GetKxUser spazio interno: GetKxVolume spazio interno: GetLoadSampleDataSetGroupIntoEnvironmentStatus spazio interno: GetUser spazio interno: ListEnvironments spazio interno: ListKxChangesets spazio interno: ListKxClusterNodes spazio interno: ListKxClusters spazio interno: ListKxDatabases spazio interno: ListKxDataviews spazio interno: ListKxEnvironments spazio interno: ListKxScalingGroups spazio interno: ListKxUsers spazio interno: ListKxVolumes

Prefisso del servizio	Azioni
	spazio interno: ListUsers spazio interno: LoadSampleDataSetGroupIntoEnvironment spazio interno: ResetUserPassword spazio interno: UpdateEnvironment spazio interno: UpdateKxClusterCodeConfiguration spazio interno: UpdateKxClusterDatabases spazio interno: UpdateKxDatabase spazio interno: UpdateKxDataview spazio interno: UpdateKxEnvironment spazio interno: UpdateKxEnvironmentNetwork spazio interno: UpdateKxUser spazio interno: UpdateKxVolume spazio interno: UpdateUser
firehose	manichetta antincendio: CreateDeliveryStream manichetta antincendio: DeleteDeliveryStream manichetta antincendio: DescribeDeliveryStream manichetta antincendio: ListDeliveryStreams manichetta antincendio: StartDeliveryStreamEncryption manichetta antincendio: StopDeliveryStreamEncryption manichetta antincendio: UpdateDestination

Prefisso del servizio	Azioni
fis	pinza: CreateExperimentTemplate pesce: CreateTargetAccountConfiguration pesce: DeleteExperimentTemplate pesce: DeleteTargetAccountConfiguration pesce: GetAction pesce: GetExperiment pesce: GetExperimentTargetAccountConfiguration pesce: GetExperimentTemplate pesce: GetSafetyLever pesce: GetTargetAccountConfiguration pesce: GetTargetResourceType pesce: ListActions pesce: ListExperimentResolvedTargets pesce: ListExperimentTargetAccountConfigurations pesce: ListExperimentTemplates pesce: ListExperiments pesce: ListTargetAccountConfigurations pesce: ListTargetResourceTypes pesce: StartExperiment pesce: StopExperiment pesce: UpdateExperimentTemplate

Prefisso del servizio	Azioni
	pesce: UpdateSafetyLeverState pesce: UpdateTargetAccountConfiguration

Prefisso del servizio	Azioni
fms	fms: AssociateAdminAccount
	fms: AssociateThirdPartyFirewall
	fms: BatchAssociateResource
	fms: BatchDisassociateResource
	fms: DeleteAppsList
	fms: DeleteNotificationChannel
	fms: DeletePolicy
	fms: DeleteProtocolsList
	fms: DeleteResourceSet
	fms: DisassociateAdminAccount
	fms: DisassociateThirdPartyFirewall
	fms: GetAdminAccount
	fms: GetAdminScope
	fms: GetAppsList
	fms: GetComplianceDetail
	fms: GetNotificationChannel
	fms: GetPolicy
	fms: GetProtectionStatus
	fms: GetProtocolsList
	fms: GetResourceSet
	fms: GetThirdPartyFirewallAssociationStatus

Prefisso del servizio	Azioni
	fms: GetViolationDetails
	fms: ListAdminAccountsForOrganization
	fms: ListAdminsManagingAccount
	fms: ListAppsLists
	fms: ListComplianceStatus
	fms: ListDiscoveredResources
	fms: ListMemberAccounts
	fms: ListPolicies
	fms: ListProtocolsLists
	fms: ListResourceSetResources
	fms: ListResourceSets
	fms: ListThirdPartyFirewallFirewallPolicies
	fms: PutAdminAccount
	fms: PutAppsList
	fms: PutNotificationChannel
	fms: PutPolicy
	fms: PutProtocolsList
	fms: PutResourceSet

Prefisso del servizio	Azioni
frauddetector	rilevatore di frodi: BatchCreateVariable rilevatore di frodi: BatchGetVariable rilevatore di frodi: CancelBatchImportJob rilevatore di frodi: CancelBatchPredictionJob rilevatore di frodi: CreateBatchImportJob rilevatore di frodi: CreateBatchPredictionJob rilevatore di frodi: CreateDetectorVersion rilevatore di frodi: CreateList rilevatore di frodi: CreateModel rilevatore di frodi: CreateModelVersion rilevatore di frodi: CreateRule rilevatore di frodi: CreateVariable rilevatore di frodi: DeleteBatchImportJob rilevatore di frodi: DeleteBatchPredictionJob rilevatore di frodi: DeleteDetector rilevatore di frodi: DeleteDetectorVersion rilevatore di frodi: DeleteEntityType rilevatore di frodi: DeleteEvent rilevatore di frodi: DeleteEventType rilevatore di frodi: DeleteEventsByEventType rilevatore di frodi: DeleteExternalModel

Prefisso del servizio	Azioni
	rilevatore di frodi: DeleteLabel rilevatore di frodi: DeleteList rilevatore di frodi: DeleteModel rilevatore di frodi: DeleteModelVersion rilevatore di frodi: DeleteOutcome rilevatore di frodi: DeleteRule rilevatore di frodi: DeleteVariable rilevatore di frodi: DescribeDetector rilevatore di frodi: DescribeModelVersions rilevatore di frodi: GetBatchImportJobs rilevatore di frodi: GetBatchPredictionJobs rilevatore di frodi: GetDeleteEventsByEventTypeStatus rilevatore di frodi: GetDetectorVersion rilevatore di frodi: GetDetectors rilevatore di frodi: GetEntityTypes rilevatore di frodi: GetEvent rilevatore di frodi: GetEventPrediction rilevatore di frodi: GetEventPredictionMetadata rilevatore di frodi: GetEventTypes rilevatore di frodi: GetExternalModels rilevatore di frodi: GetKMSEncryptionKey

Prefisso del servizio	Azioni
	rilevatore di frodi: GetLabels rilevatore di frodi: GetListElements rilevatore di frodi: GetListsMetadata rilevatore di frodi: GetModelVersion rilevatore di frodi: GetModels rilevatore di frodi: GetOutcomes rilevatore di frodi: GetRules rilevatore di frodi: GetVariables rilevatore di frodi: ListEventPredictions rilevatore di frodi: PutDetector rilevatore di frodi: PutEntityType rilevatore di frodi: PutEventType rilevatore di frodi: PutExternalModel rilevatore di frodi: PutKMSEncryptionKey rilevatore di frodi: PutLabel rilevatore di frodi: PutOutcome rilevatore di frodi: SendEvent rilevatore di frodi: UpdateDetectorVersion rilevatore di frodi: UpdateDetectorVersionMetadata rilevatore di frodi: UpdateDetectorVersionStatus rilevatore di frodi: UpdateEventLabel

Prefisso del servizio	Azioni
	rilevatore di frodi: UpdateList rilevatore di frodi: UpdateModel rilevatore di frodi: UpdateModelVersion rilevatore di frodi: UpdateModelVersionStatus rilevatore di frodi: UpdateRuleMetadata rilevatore di frodi: UpdateRuleVersion rilevatore di frodi: UpdateVariable

Prefisso del servizio	Azioni
fsx	fax: AssociateFileSystemAliases
	fax: CancelDataRepositoryTask
	fax: CopyBackup
	fax: CreateDataRepositoryTask
	fax: CreateFileCache
	fax: CreateFileSystem
	fax: CreateFileSystemFromBackup
	fax: CreateSnapshot
	fax: CreateStorageVirtualMachine
	fax: CreateVolume
	fax: CreateVolumeFromBackup
	fax: DeleteBackup
	fax: DeleteFileCache
	fax: DeleteFileSystem
	fax: DeleteSnapshot
	fax: DeleteStorageVirtualMachine
	fax: DeleteVolume
	fax: DescribeBackups
	fax: DescribeDataRepositoryAssociations
	fax: DescribeDataRepositoryTasks
	fax: DescribeFileCaches

Prefisso del servizio	Azioni
	fax: DescribeFileSystemAliases
	fax: DescribeFileSystems
	fax: DescribeS3AccessPointAttachments
	fax: DescribeSharedVpcConfiguration
	fax: DescribeSnapshots
	fax: DescribeStorageVirtualMachines
	fax: DescribeVolumes
	fax: DetachAndDeleteS3AccessPoint
	fax: DisassociateFileSystemAliases
	fax: ReleaseFileSystemNfsV3Locks
	fax: RestoreVolumeFromSnapshot
	fax: StartMisconfiguredStateRecovery
	fax: UpdateDataRepositoryAssociation
	fax: UpdateFileCache
	fax: UpdateFileSystem
	fax: UpdateSharedVpcConfiguration
	fax: UpdateSnapshot
	fax: UpdateStorageVirtualMachine
	fax: UpdateVolume

Prefisso del servizio	Azioni
gamelift	rinnovamento del gioco: AcceptMatch rinnovamento del gioco: ClaimGameServer rinnovamento del gioco: CreateAlias rinnovamento del gioco: CreateBuild rinnovamento del gioco: CreateContainerGroupDefinition rinnovamento del gioco: CreateFleet rinnovamento del gioco: CreateFleetLocations rinnovamento del gioco: CreateGameServerGroup rinnovamento del gioco: CreateGameSession rinnovamento del gioco: CreateGameSessionQueue rinnovamento del gioco: CreateLocation rinnovamento del gioco: CreateMatchmakingConfiguration rinnovamento del gioco: CreateMatchmakingRuleSet rinnovamento del gioco: CreatePlayerSession rinnovamento del gioco: CreatePlayerSessions rinnovamento del gioco: CreateScript rinnovamento del gioco: CreateVpcPeeringAuthorization rinnovamento del gioco: CreateVpcPeeringConnection rinnovamento del gioco: DeleteAlias rinnovamento del gioco: DeleteBuild rinnovamento del gioco: DeleteContainerGroupDefinition

Prefisso del servizio	Azioni
	rinnovamento del gioco: DeleteFleet rinnovamento del gioco: DeleteFleetLocations rinnovamento del gioco: DeleteGameServerGroup rinnovamento del gioco: DeleteGameSessionQueue rinnovamento del gioco: DeleteLocation rinnovamento del gioco: DeleteMatchmakingConfiguration rinnovamento del gioco: DeleteMatchmakingRuleSet rinnovamento del gioco: DeleteScalingPolicy rinnovamento del gioco: DeleteScript rinnovamento del gioco: DeleteVpcPeeringAuthorization rinnovamento del gioco: DeleteVpcPeeringConnection rinnovamento del gioco: DeregisterCompute rinnovamento del gioco: DeregisterGameServer rinnovamento del gioco: DescribeAlias rinnovamento del gioco: DescribeBuild rinnovamento del gioco: DescribeCompute rinnovamento del gioco: DescribeContainerFleet rinnovamento del gioco: DescribeContainerGroupDefinition rinnovamento del gioco: DescribeContainerGroupPortMappings rinnovamento del gioco: DescribeEC2InstanceLimits rinnovamento del gioco: DescribeFleetAttributes

Prefisso del servizio	Azioni
	rinnovamento del gioco: DescribeFleetCapacity
	rinnovamento del gioco: DescribeFleetEvents
	rinnovamento del gioco: DescribeFleetLocationAttributes
	rinnovamento del gioco: DescribeFleetLocationCapacity
	rinnovamento del gioco: DescribeFleetLocationUtilization
	rinnovamento del gioco: DescribeFleetPortSettings
	rinnovamento del gioco: DescribeFleetUtilization
	rinnovamento del gioco: DescribeGameServer
	rinnovamento del gioco: DescribeGameServerGroup
	rinnovamento del gioco: DescribeGameServerInstances
	rinnovamento del gioco: DescribeGameSessionDetails
	rinnovamento del gioco: DescribeGameSessionPlacement
	rinnovamento del gioco: DescribeGameSessionQueues
	rinnovamento del gioco: DescribeGameSessions
	rinnovamento del gioco: DescribeInstances
	rinnovamento del gioco: DescribeMatchmaking
	rinnovamento del gioco: DescribeMatchmakingConfigurations
	rinnovamento del gioco: DescribeMatchmakingRuleSets
	rinnovamento del gioco: DescribePlayerSessions
	rinnovamento del gioco: DescribeRuntimeConfiguration
	rinnovamento del gioco: DescribeScalingPolicies

Prefisso del servizio	Azioni
	rinnovamento del gioco: DescribeScript rinnovamento del gioco: DescribeVpcPeeringAuthorizations rinnovamento del gioco: DescribeVpcPeeringConnections rinnovamento del gioco: GetComputeAccess rinnovamento del gioco: GetComputeAuthToken rinnovamento del gioco: GetGameSessionLogUrl rinnovamento del gioco: GetInstanceAccess rinnovamento del gioco: GetPlayerConnectionDetails rinnovamento del gioco: ListAliases rinnovamento del gioco: ListBuilds rinnovamento del gioco: ListCompute rinnovamento del gioco: ListContainerFleets rinnovamento del gioco: ListContainerGroupDefinitionVersions rinnovamento del gioco: ListContainerGroupDefinitions rinnovamento del gioco: ListFleetDeployments rinnovamento del gioco: ListFleets rinnovamento del gioco: ListGameServerGroups rinnovamento del gioco: ListGameServers rinnovamento del gioco: ListLocations rinnovamento del gioco: ListScripts rinnovamento del gioco: PutScalingPolicy

Prefisso del servizio	Azioni
	rinnovamento del gioco: RegisterCompute rinnovamento del gioco: RegisterGameServer rinnovamento del gioco: RequestUploadCredentials rinnovamento del gioco: ResolveAlias rinnovamento del gioco: ResumeGameServerGroup rinnovamento del gioco: SearchGameSessions rinnovamento del gioco: StartFleetActions rinnovamento del gioco: StartGameSessionPlacement rinnovamento del gioco: StartMatchBackfill rinnovamento del gioco: StartMatchmaking rinnovamento del gioco: StopFleetActions rinnovamento del gioco: StopGameSessionPlacement rinnovamento del gioco: StopMatchmaking rinnovamento del gioco: SuspendGameServerGroup rinnovamento del gioco: TerminateGameSession rinnovamento del gioco: UpdateAlias rinnovamento del gioco: UpdateBuild rinnovamento del gioco: UpdateContainerGroupDefinition rinnovamento del gioco: UpdateFleetAttributes rinnovamento del gioco: UpdateFleetCapacity rinnovamento del gioco: UpdateFleetPortSettings

Prefisso del servizio	Azioni
	rinnovamento del gioco: UpdateGameServer rinnovamento del gioco: UpdateGameServerGroup rinnovamento del gioco: UpdateGameSession rinnovamento del gioco: UpdateGameSessionQueue rinnovamento del gioco: UpdateMatchmakingConfiguration rinnovamento del gioco: UpdateRuntimeConfiguration rinnovamento del gioco: UpdateScript rinnovamento del gioco: ValidateMatchmakingRuleSet

Prefisso del servizio	Azioni
geo	geo: AssociateTrackerConsumer
	geo: BatchDeleteDevicePositionHistory
	geo: BatchDeleteGeofence
	geo: BatchEvaluateGeofences
	geo: BatchGetDevicePosition
	geo: BatchPutGeofence
	geo: BatchUpdateDevicePosition
	geo: CalculateRoute
	geo: CalculateRouteMatrix
	geo: CancelJob
	geo: CreateGeofenceCollection
	geo: CreateMap
	geo: CreatePlaceIndex
	geo: CreateRouteCalculator
	geo: CreateTracker
	geo: DeleteGeofenceCollection
	geo: DeleteKey
	geo: DeleteMap
	geo: DeletePlaceIndex
	geo: DeleteRouteCalculator
	geo: DeleteTracker

Prefisso del servizio	Azioni
	geo: DescribeGeofenceCollection
	geo: DescribeKey
	geo: DescribeMap
	geo: DescribePlaceIndex
	geo: DescribeRouteCalculator
	geo: DescribeTracker
	geo: DisassociateTrackerConsumer
	geo: ForecastGeofenceEvents
	geo: GetDevicePosition
	geo: GetDevicePositionHistory
	geo: GetGeofence
	geo: GetJob
	geo: GetMapGlyphs
	geo: GetMapSprites
	geo: GetMapStyleDescriptor
	geo: GetMapTile
	geo: GetPlace
	geo: ListDevicePositions
	geo: ListGeofenceCollections
	geo: ListGeofences
	geo: ListJobs

Prefisso del servizio	Azioni
	geo: ListKeys
	geo: ListMaps
	geo: ListPlaceIndexes
	geo: ListRouteCalculators
	geo: ListTrackerConsumers
	geo: ListTrackers
	geo: PutGeofence
	geo: SearchPlaceIndexForPosition
	geo: SearchPlaceIndexForSuggestions
	geo: SearchPlaceIndexForText
	geo: UpdateGeofenceCollection
	geo: UpdateKey
	geo: UpdateMap
	geo: UpdatePlaceIndex
	geo: UpdateRouteCalculator
	geo: UpdateTracker
	geo: VerifyDevicePosition

Prefisso del servizio	Azioni
glacier	ghiacciaio: AbortMultipartUpload ghiacciaio: AbortVaultLock ghiacciaio: CompleteMultipartUpload ghiacciaio: CompleteVaultLock ghiacciaio: CreateVault ghiacciaio: DeleteArchive ghiacciaio: DeleteVault ghiacciaio: DeleteVaultAccessPolicy ghiacciaio: DeleteVaultNotifications ghiacciaio: DescribeJob ghiacciaio: DescribeVault ghiacciaio: GetDataRetrievalPolicy ghiacciaio: GetJobOutput ghiacciaio: GetVaultAccessPolicy ghiacciaio: GetVaultLock ghiacciaio: GetVaultNotifications ghiacciaio: InitiateJob ghiacciaio: InitiateMultipartUpload ghiacciaio: InitiateVaultLock ghiacciaio: ListJobs ghiacciaio: ListMultipartUploads

Prefisso del servizio	Azioni
	ghiacciaio: ListParts ghiacciaio: ListProvisionedCapacity ghiacciaio: ListVaults ghiacciaio: PurchaseProvisionedCapacity ghiacciaio: SetDataRetrievalPolicy ghiacciaio: SetVaultAccessPolicy ghiacciaio: SetVaultNotifications ghiacciaio: UploadArchive ghiacciaio: UploadMultipartPart

Prefisso del servizio	Azioni
grafana	grafana: AssociateLicense
	grafana: CreateWorkspace
	grafana: CreateWorkspaceApiKey
	grafana: CreateWorkspaceServiceAccount
	grafana: CreateWorkspaceServiceAccountToken
	grafana: DeleteWorkspace
	grafana: DeleteWorkspaceApiKey
	grafana: DeleteWorkspaceServiceAccount
	grafana: DeleteWorkspaceServiceAccountToken
	grafana: DescribeWorkspace
	grafana: DescribeWorkspaceAuthentication
	grafana: DescribeWorkspaceConfiguration
	grafana: DisassociateLicense
	grafana: ListPermissions
	grafana: ListVersions
	grafana: ListWorkspaceServiceAccountTokens
	grafana: ListWorkspaceServiceAccounts
	grafana: ListWorkspaces
	grafana: UpdatePermissions
	grafana: UpdateWorkspace
	grafana: UpdateWorkspaceAuthentication

Prefisso del servizio	Azioni
	grafana: UpdateWorkspaceConfiguration

Prefisso del servizio	Azioni
greengrass	erba verde: AssociateRoleToGroup erba verde: AssociateServiceRoleToAccount erba verde: BatchAssociateClientDeviceWithCoreDevice erba verde: BatchDisassociateClientDeviceFromCoreDevice erba verde: CancelDeployment erba verde: CreateComponentVersion erba verde: CreateConnectorDefinition erba verde: CreateConnectorDefinitionVersion erba verde: CreateCoreDefinition erba verde: CreateCoreDefinitionVersion erba verde: CreateDeployment erba verde: CreateDeviceDefinition erba verde: CreateDeviceDefinitionVersion erba verde: CreateFunctionDefinition erba verde: CreateFunctionDefinitionVersion erba verde: CreateGroup erba verde: CreateGroupCertificateAuthority erba verde: CreateGroupVersion erba verde: CreateLoggerDefinition erba verde: CreateLoggerDefinitionVersion erba verde: CreateResourceDefinition

Prefisso del servizio	Azioni
	erba verde: CreateResourceDefinitionVersion erba verde: CreateSoftwareUpdateJob erba verde: CreateSubscriptionDefinition erba verde: CreateSubscriptionDefinitionVersion erba verde: DeleteComponent erba verde: DeleteConnectorDefinition erba verde: DeleteCoreDefinition erba verde: DeleteCoreDevice erba verde: DeleteDeployment erba verde: DeleteDeviceDefinition erba verde: DeleteFunctionDefinition erba verde: DeleteGroup erba verde: DeleteLoggerDefinition erba verde: DeleteResourceDefinition erba verde: DeleteSubscriptionDefinition erba verde: DescribeComponent erba verde: DisassociateRoleFromGroup erba verde: DisassociateServiceRoleFromAccount erba verde: GetAssociatedRole erba verde: GetBulkDeploymentStatus erba verde: GetComponent

Prefisso del servizio	Azioni
	erba verde: GetComponentVersionArtifact erba verde: GetConnectivityInfo erba verde: GetConnectorDefinition erba verde: GetConnectorDefinitionVersion erba verde: GetCoreDefinition erba verde: GetCoreDefinitionVersion erba verde: GetCoreDevice erba verde: GetDeployment erba verde: GetDeploymentStatus erba verde: GetDeviceDefinition erba verde: GetDeviceDefinitionVersion erba verde: GetFunctionDefinition erba verde: GetFunctionDefinitionVersion erba verde: GetGroup erba verde: GetGroupCertificateAuthority erba verde: GetGroupCertificateConfiguration erba verde: GetGroupVersion erba verde: GetLoggerDefinition erba verde: GetLoggerDefinitionVersion erba verde: GetResourceDefinition erba verde: GetResourceDefinitionVersion

Prefisso del servizio	Azioni
	erba verde: GetServiceRoleForAccount erba verde: GetSubscriptionDefinition erba verde: GetSubscriptionDefinitionVersion erba verde: GetThingRuntimeConfiguration erba verde: ListBulkDeploymentDetailedReports erba verde: ListBulkDeployments erba verde: ListClientDevicesAssociatedWithCoreDevice erba verde: ListComponentVersions erba verde: ListComponents erba verde: ListConnectorDefinitionVersions erba verde: ListConnectorDefinitions erba verde: ListCoreDefinitionVersions erba verde: ListCoreDefinitions erba verde: ListCoreDevices erba verde: ListDeployments erba verde: ListDeviceDefinitionVersions erba verde: ListDeviceDefinitions erba verde: ListEffectiveDeployments erba verde: ListFunctionDefinitionVersions erba verde: ListFunctionDefinitions erba verde: ListGroupCertificateAuthorities

Prefisso del servizio	Azioni
	erba verde: ListGroupVersions erba verde: ListGroups erba verde: ListInstalledComponents erba verde: ListLoggerDefinitionVersions erba verde: ListLoggerDefinitions erba verde: ListResourceDefinitionVersions erba verde: ListResourceDefinitions erba verde: ListSubscriptionDefinitionVersions erba verde: ListSubscriptionDefinitions erba verde: ResetDeployments erba verde: StartBulkDeployment erba verde: StopBulkDeployment erba verde: UpdateConnectivityInfo erba verde: UpdateConnectorDefinition erba verde: UpdateCoreDefinition erba verde: UpdateDeviceDefinition erba verde: UpdateFunctionDefinition erba verde: UpdateGroup erba verde: UpdateGroupCertificateConfiguration erba verde: UpdateLoggerDefinition erba verde: UpdateResourceDefinition

Prefisso del servizio	Azioni
	erba verde: UpdateSubscriptionDefinition erba verde: UpdateThingRuntimeConfiguration

Prefisso del servizio	Azioni
groundstation	stazione a terra: CancelContact
	stazione a terra: CreateConfig
	stazione a terra: CreateDataflowEndpointGroup
	stazione a terra: CreateDataflowEndpointGroupV2
	stazione a terra: CreateEphemeris
	stazione a terra: CreateMissionProfile
	stazione a terra: DeleteConfig
	stazione a terra: DeleteDataflowEndpointGroup
	stazione a terra: DeleteEphemeris
	stazione a terra: DeleteMissionProfile
	stazione a terra: DescribeContact
	stazione a terra: DescribeContactVersion
	stazione a terra: DescribeEphemeris
	stazione a terra: GetConfig
	stazione a terra: GetDataflowEndpointGroup
	stazione a terra: GetMinuteUsage
	stazione a terra: GetMissionProfile
	stazione a terra: GetSatellite
	stazione a terra: ListAntennas
	stazione a terra: ListConfigs
	stazione a terra: ListContacts

Prefisso del servizio	Azioni
	stazione a terra: ListDataflowEndpointGroups
	stazione a terra: ListEphemerides
	stazione a terra: ListGroundStationReservations
	stazione a terra: ListGroundStations
	stazione a terra: ListMissionProfiles
	stazione a terra: ListSatellites
	stazione a terra: RegisterAgent
	stazione a terra: ReserveContact
	stazione a terra: UpdateAgentStatus
	stazione a terra: UpdateConfig
	stazione a terra: UpdateEphemeris
	stazione a terra: UpdateMissionProfile

Prefisso del servizio	Azioni
guardduty	servizio di guardia: AcceptAdministratorInvitation servizio di guardia: AcceptInvitation servizio di guardia: ArchiveFindings servizio di guardia: CreateDetector servizio di guardia: CreateFilter guardduty:CreateIPSet servizio di guardia: CreateInvestigation servizio di guardia: CreateMalwareProtectionPlan servizio di guardia: CreateMembers servizio di guardia: CreatePublishingDestination servizio di guardia: CreateSampleFindings servizio di guardia: CreateThreatEntitySet servizio di guardia: CreateThreatIntelSet servizio di guardia: CreateTrustedEntitySet servizio di guardia: DeclineInvitations servizio di guardia: DeleteDetector servizio di guardia: DeleteFilter guardduty:DeleteIPSet servizio di guardia: DeleteInvitations servizio di guardia: DeleteMalwareProtectionPlan servizio di guardia: DeleteMembers

Prefisso del servizio	Azioni
	servizio di guardia: DeletePublishingDestination servizio di guardia: DeleteThreatEntitySet servizio di guardia: DeleteThreatIntelSet servizio di guardia: DeleteTrustedEntitySet servizio di guardia: DescribeMalwareScans servizio di guardia: DescribeOrganizationConfiguration servizio di guardia: DescribePublishingDestination servizio di guardia: DisableOrganizationAdminAccount servizio di guardia: DisassociateFromAdministratorAccount servizio di guardia: DisassociateFromMasterAccount servizio di guardia: DisassociateMembers servizio di guardia: EnableOrganizationAdminAccount servizio di guardia: GetAdministratorAccount servizio di guardia: GetCoverageStatistics servizio di guardia: GetDetector servizio di guardia: GetFilter servizio di guardia: GetFindings servizio di guardia: GetFindingsStatistics guardduty:GetIPSet servizio di guardia: GetInvestigation servizio di guardia: GetInvitationsCount

Prefisso del servizio	Azioni
	servizio di guardia: GetMalwareProtectionPlan servizio di guardia: GetMalwareScan servizio di guardia: GetMalwareScanSettings servizio di guardia: GetMasterAccount servizio di guardia: GetMemberDetectors servizio di guardia: GetMembers servizio di guardia: GetOrganizationStatistics servizio di guardia: GetRemainingFreeTrialDays servizio di guardia: GetThreatEntitySet servizio di guardia: GetThreatIntelSet servizio di guardia: GetTrustedEntitySet servizio di guardia: GetUsageStatistics servizio di guardia: InviteMembers servizio di guardia: ListCoverage servizio di guardia: ListDetectors servizio di guardia: ListFilters servizio di guardia: ListFindings guardduty:ListIPSets servizio di guardia: ListInvestigations servizio di guardia: ListInvitations servizio di guardia: ListMalwareProtectionPlans

Prefisso del servizio	Azioni
	servizio di guardia: ListMalwareScans servizio di guardia: ListMembers servizio di guardia: ListOrganizationAdminAccounts servizio di guardia: ListPublishingDestinations servizio di guardia: ListThreatEntitySets servizio di guardia: ListThreatIntelSets servizio di guardia: ListTrustedEntitySets servizio di guardia: StartMalwareScan servizio di guardia: StartMonitoringMembers servizio di guardia: StopMonitoringMembers servizio di guardia: UnarchiveFindings servizio di guardia: UpdateDetector servizio di guardia: UpdateFilter servizio di guardia: UpdateFindingsFeedback guardduty:UpdateIPSet servizio di guardia: UpdateMalwareProtectionPlan servizio di guardia: UpdateMalwareScanSettings servizio di guardia: UpdateMemberDetectors servizio di guardia: UpdateOrganizationConfiguration servizio di guardia: UpdatePublishingDestination servizio di guardia: UpdateThreatEntitySet

Prefisso del servizio	Azioni
	servizio di guardia: UpdateThreatIntelSet servizio di guardia: UpdateTrustedEntitySet

Prefisso del servizio	Azioni
healthlake	lago sanitario: CancelFHIRExportJobWithDelete
	lago sanitario: CreateDataTransformationProfile
	healthlake:CreateFHIRDatastore
	lago sanitario: CreateResource
	lago sanitario: DeleteDataTransformationProfile
	healthlake:DeleteFHIRDatastore
	lago sanitario: DeleteResource
	lago sanitario: DescribeDataTransformationJob
	healthlake:DescribeFHIRDatastore
	lago sanitario: DescribeFHIRExportJob
	lago sanitario: DescribeFHIRExportJobWithGet
	lago sanitario: DescribeFHIRImportJob
	lago sanitario: GetCapabilities
	lago sanitario: GetDataTransformationProfile
	lago sanitario: ListDataTransformationJobs
	lago sanitario: ListDataTransformationProfileVersions
	lago sanitario: ListDataTransformationProfiles
	healthlake:ListFHIRDatastores
	lago sanitario: ListFHIRExportJobs
	lago sanitario: ListFHIRImportJobs
	lago sanitario: PublishDataTransformationProfile

Prefisso del servizio	Azioni
	lago sanitario: ReadResource
	lago sanitario: SearchEverything
	lago sanitario: SearchWithGet
	lago sanitario: SearchWithPost
	lago sanitario: StartFHIRExportJob
	lago sanitario: StartFHIRExportJobWithPost
	lago sanitario: StartFHIRImportJob
	lago sanitario: UpdateDataTransformationProfile
	Healthlake: aggiorna il datastore Fhir
	Healthlake: UpdateProfileWithAgent
	lago sanitario: UpdateResource

Prefisso del servizio	Azioni
honeycode	codice del miele: BatchCreateTableRows codice del miele: BatchDeleteTableRows codice del miele: BatchUpdateTableRows codice del miele: BatchUpsertTableRows codice del miele: DescribeTableDataImportJob codice del miele: GetScreenData codice del miele: InvokeScreenAutomation codice del miele: ListTableColumns codice del miele: ListTableRows codice del miele: ListTables codice del miele: QueryTableRows codice del miele: StartTableDataImportJob

Prefisso del servizio	Azioni
iam	obiettivo: AddClientIDToOpenIDConnectProvider
	Io sono: AddRoleToInstanceProfile
	Io sono: AddUserToGroup
	Io sono: AttachGroupPolicy
	Io sono: AttachRolePolicy
	Io sono: AttachUserPolicy
	Io sono: ChangePassword
	Io sono: CreateAccessKey
	Io sono: CreateAccountAlias
	Io sono: CreateGroup
	Io sono: CreateInstanceProfile
	Io sono: CreateLoginProfile
	Io sono: CreateOpenIDConnectProvider
	Io sono: CreatePolicy
	Io sono: CreatePolicyVersion
	Io sono: CreateRole
	iam:CreateSAMLProvider
	Io sono: CreateServiceLinkedRole
	Io sono: CreateServiceSpecificCredential
	Io sono: CreateUser
	Io sono: CreateVirtualMFADevice

Prefisso del servizio	Azioni
	iam:DeactivateMFADevice
	Io sono: DeleteAccessKey
	Io sono: DeleteAccountAlias
	Io sono: DeleteAccountPasswordPolicy
	Io sono: DeleteCloudFrontPublicKey
	Io sono: DeleteGroup
	Io sono: DeleteGroupPolicy
	Io sono: DeleteInstanceProfile
	Io sono: DeleteLoginProfile
	Io sono: DeleteOpenIDConnectProvider
	Io sono: DeletePolicy
	Io sono: DeletePolicyVersion
	Io sono: DeleteRole
	Io sono: DeleteRolePermissionsBoundary
	Io sono: DeleteRolePolicy
	iam:DeleteSAMLProvider
	Io sono: DeleteSSHPublicKey
	Io sono: DeleteServerCertificate
	Io sono: DeleteServiceLinkedRole
	Io sono: DeleteServiceSpecificCredential
	Io sono: DeleteSigningCertificate

Prefisso del servizio	Azioni
	Io sono: DeleteUser
	Io sono: DeleteUserPermissionsBoundary
	Io sono: DeleteUserPolicy
	Io sono: DeleteVirtualMFADevice
	Io sono: DetachGroupPolicy
	Io sono: DetachRolePolicy
	Io sono: DetachUserPolicy
	Io sono: DisableOrganizationsRootCredentialsManagement
	sono: DisableOrganizationsRootSessions
	sono: DisableOutboundWebIdentityFederation
	iam:EnableMFADevice
	sono: EnableOrganizationsRootCredentialsManagement
	sono: EnableOrganizationsRootSessions
	sono: EnableOutboundWebIdentityFederation
	sono: GenerateCredentialReport
	sono: GenerateOrganizationsAccessReport
	sono: GenerateServiceLastAccessedDetails
	sono: GetAccessKeyLastUsed
	sono: GetAccountAuthorizationDetails
	sono: GetAccountEmailAddress
	sono: GetAccountName

Prefisso del servizio	Azioni
	sono: GetAccountPasswordPolicy
	sono: GetAccountSummary
	sono: GetCloudFrontPublicKey
	sono: GetContextKeysForCustomPolicy
	sono: GetContextKeysForPrincipalPolicy
	sono: GetCredentialReport
	sono: GetGroup
	sono: GetGroupPolicy
	sono: GetInstanceProfile
	sono: GetLoginProfile
	iam:GetMFADevice
	sono: GetOpenIDConnectProvider
	sono: GetOrganizationsAccessReport
	sono: GetOutboundWebIdentityFederationInfo
	sono: GetPolicy
	sono: GetPolicyVersion
	sono: GetRole
	sono: GetRolePolicy
	iam:GetSAMLProvider
	sono: GetSSHPublicKey
	sono: GetServerCertificate

Prefisso del servizio	Azioni
	sono: GetServiceLastAccessedDetails
	sono: GetServiceLastAccessedDetailsWithEntities
	sono: GetServiceLinkedRoleDeletionStatus
	sono: GetUser
	sono: GetUserPolicy
	sono: ListAccessKeys
	sono: ListAccountAliases
	sono: ListAttachedGroupPolicies
	sono: ListAttachedRolePolicies
	sono: ListAttachedUserPolicies
	sono: ListCloudFrontPublicKeys
	sono: ListDelegationRequests
	sono: ListEntitiesForPolicy
	sono: ListGroupPolicies
	sono: ListGroups
	sono: ListGroupsForUser
	sono: ListInstanceProfiles
	sono: ListInstanceProfilesForRole
	iam:ListMFADevices
	sono: ListOpenIDConnectProviders
	sono: ListOrganizationsFeatures

Prefisso del servizio	Azioni
	sono: ListPolicies
	sono: ListPoliciesGrantingServiceAccess
	sono: ListPolicyVersions
	sono: ListRolePolicies
	sono: ListRoles
	iam:ListSAMLProviders
	sono: ListSSHPublicKeys
	sono: ListSTSRegionalEndpointsStatus
	sono: ListServerCertificates
	sono: ListServiceSpecificCredentials
	sono: ListSigningCertificates
	sono: ListUserPolicies
	sono: ListUsers
	sono: ListVirtualMFADevices
	Io sono: PutGroupPolicy
	Io sono: PutRolePermissionsBoundary
	Io sono: PutRolePolicy
	Io sono: PutUserPermissionsBoundary
	Io sono: PutUserPolicy
	Io sono: RemoveClientIDFromOpenIDConnectProvider
	Io sono: RemoveRoleFromInstanceProfile

Prefisso del servizio	Azioni
	Io sono: RemoveUserFromGroup
	Io sono: ResetServiceSpecificCredential
	iam:ResyncMFADevice
	Io sono: SetDefaultPolicyVersion
	Io sono: SetSTSRegionalEndpointStatus
	Io sono: SetSecurityTokenServicePreferences
	Io sono: SimulateCustomPolicy
	Io sono: SimulatePrincipalPolicy
	Io sono: UpdateAccessKey
	Io sono: UpdateAccountEmailAddress
	Io sono: UpdateAccountName
	Io sono: UpdateAccountPasswordPolicy
	Io sono: UpdateAssumeRolePolicy
	Io sono: UpdateCloudFrontPublicKey
	Io sono: UpdateGroup
	Io sono: UpdateLoginProfile
	Io sono: UpdateOpenIDConnectProviderThumbprint
	Io sono: UpdateRole
	Io sono: UpdateRoleDescription
	iam:UpdateSAMLProvider
	Io sono: UpdateSSHPublicKey

Prefisso del servizio	Azioni
	Io sono: UpdateServerCertificate
	Io sono: UpdateServiceSpecificCredential
	Io sono: UpdateSigningCertificate
	Io sono: UpdateUser
	Io sono: UploadCloudFrontPublicKey
	Io sono: UploadSSHPublicKey
	Io sono: UploadServerCertificate
	Io sono: UploadSigningCertificate

Prefisso del servizio	Azioni
identitystore	archivio di identità: CreateGroup archivio di identità: CreateGroupMembership archivio di identità: CreateUser archivio di identità: DeleteGroup archivio di identità: DeleteGroupMembership archivio di identità: DeleteUser archivio di identità: DescribeGroup archivio di identità: DescribeGroupMembership archivio di identità: DescribeUser archivio di identità: GetGroupId archivio di identità: GetGroupMembershipId archivio di identità: GetUserId archivio di identità: IsMemberInGroups archivio di identità: ListGroupMemberships archivio di identità: ListGroupMembershipsForMember archivio di identità: ListGroups archivio di identità: ListUsers archivio di identità: UpdateGroup archivio di identità: UpdateUser

Prefisso del servizio	Azioni
imagebuilder	generatore di immagini: CancelImageCreation
	generatore di immagini: CancelLifecycleExecution
	generatore di immagini: CreateComponent
	generatore di immagini: CreateContainerRecipe
	generatore di immagini: CreateDistributionConfiguration
	generatore di immagini: CreateImage
	generatore di immagini: CreateImagePipeline
	generatore di immagini: CreateImageRecipe
	generatore di immagini: CreateInfrastructureConfiguration
	generatore di immagini: CreateLifecyclePolicy
	generatore di immagini: CreateWorkflow
	generatore di immagini: DeleteComponent
	generatore di immagini: DeleteContainerRecipe
	generatore di immagini: DeleteDistributionConfiguration
	generatore di immagini: DeleteImage
	generatore di immagini: DeleteImagePipeline
	generatore di immagini: DeleteImageRecipe
	generatore di immagini: DeleteInfrastructureConfiguration
	generatore di immagini: DeleteLifecyclePolicy
	generatore di immagini: DeleteWorkflow
	generatore di immagini: DistributeImage

Prefisso del servizio	Azioni
	generatore di immagini: GetComponentPolicy
	generatore di immagini: GetContainerRecipePolicy
	generatore di immagini: GetImagePolicy
	generatore di immagini: GetImageRecipePolicy
	generatore di immagini: GetLifecycleExecution
	generatore di immagini: GetLifecyclePolicy
	generatore di immagini: GetMarketplaceResource
	generatore di immagini: GetWorkflowExecution
	generatore di immagini: GetWorkflowStepExecution
	generatore di immagini: ImportComponent
	generatore di immagini: ImportDiskImage
	generatore di immagini: ImportVmImage
	generatore di immagini: ListComponentBuildVersions
	generatore di immagini: ListComponents
	generatore di immagini: ListContainerRecipes
	generatore di immagini: ListDistributionConfigurations
	generatore di immagini: ListImageBuildVersions
	generatore di immagini: ListImagePackages
	generatore di immagini: ListImagePipelineImages
	generatore di immagini: ListImagePipelines
	generatore di immagini: ListImageRecipes

Prefisso del servizio	Azioni
	generatore di immagini: ListImageScanFindingAggregations
	generatore di immagini: ListImageScanFindings
	generatore di immagini: ListImages
	generatore di immagini: ListInfrastructureConfigurations
	generatore di immagini: ListLifecycleExecutionResources
	generatore di immagini: ListLifecycleExecutions
	generatore di immagini: ListLifecyclePolicies
	generatore di immagini: ListWaitingWorkflowSteps
	generatore di immagini: ListWorkflowExecutions
	generatore di immagini: ListWorkflowStepExecutions
	generatore di immagini: ListWorkflows
	generatore di immagini: PutComponentPolicy
	generatore di immagini: PutContainerRecipePolicy
	generatore di immagini: PutImagePolicy
	generatore di immagini: PutImageRecipePolicy
	generatore di immagini: RetryImage
	generatore di immagini: SendWorkflowStepAction
	generatore di immagini: StartImagePipelineExecution
	generatore di immagini: StartResourceStateUpdate
	generatore di immagini: UpdateDistributionConfiguration
	generatore di immagini: UpdateImagePipeline

Prefisso del servizio	Azioni
	generatore di immagini: UpdateInfrastructureConfiguration

Prefisso del servizio	Azioni
inspector	ispettore: AddAttributesToFindings ispettore: CreateAssessmentTarget ispettore: CreateAssessmentTemplate ispettore: CreateExclusionsPreview ispettore: CreateResourceGroup ispettore: DeleteAssessmentRun ispettore: DeleteAssessmentTarget ispettore: DeleteAssessmentTemplate ispettore: DescribeAssessmentRuns ispettore: DescribeAssessmentTargets ispettore: DescribeAssessmentTemplates ispettore: DescribeCrossAccountAccessRole ispettore: DescribeExclusions ispettore: DescribeFindings ispettore: DescribeResourceGroups ispettore: DescribeRulesPackages ispettore: GetAssessmentReport ispettore: GetExclusionsPreview ispettore: GetTelemetryMetadata ispettore: ListAssessmentRunAgents ispettore: ListAssessmentRuns

Prefisso del servizio	Azioni
	ispettore: ListAssessmentTargets ispettore: ListAssessmentTemplates ispettore: ListEventSubscriptions ispettore: ListExclusions ispettore: ListFindings ispettore: ListRulesPackages ispettore: PreviewAgents ispettore: RegisterCrossAccountAccessRole ispettore: RemoveAttributesFromFindings ispettore: StartAssessmentRun ispettore: StopAssessmentRun ispettore: SubscribeToEvent ispettore: UnsubscribeFromEvent ispettore: UpdateAssessmentTarget

Prefisso del servizio	Azioni
inspector2	ispettore 2: AssociateMember
	ispettore 2: BatchGetAccountStatus
	ispettore 2: BatchGetCodeSnippet
	ispettore 2: BatchGetFindingDetails
	ispettore 2: BatchGetFreeTrialInfo
	ispettore 2: BatchGetMemberEc2DeepInspectionStatus
	ispettore 2: BatchUpdateMemberEc2DeepInspectionStatus
	ispettore 2: CancelFindingsReport
	ispettore 2: CancelSbomExport
	ispettore 2: CreateCisScanConfiguration
	ispettore 2: CreateCodeSecurityIntegration
	ispettore 2: CreateConnector
	ispettore 2: CreateFilter
	ispettore 2: CreateFindingsReport
	ispettore 2: CreateSbomExport
	ispettore 2: DeleteCisScanConfiguration
	ispettore 2: DeleteCodeSecurityIntegration
	ispettore 2: DeleteConnector
	ispettore 2: DeleteFilter
	ispettore 2: DescribeOrganizationConfiguration
	inspector2:Disable

Prefisso del servizio	Azioni
	ispettore 2: DisableDelegatedAdminAccount
	ispettore 2: DisassociateMember
	inspector2:Enable
	ispettore 2: EnableDelegatedAdminAccount
	ispettore 2: GetCisScanReport
	ispettore 2: GetCisScanResultDetails
	ispettore 2: GetClustersForImage
	ispettore 2: GetCodeSecurityIntegration
	ispettore 2: GetCodeSecurityScan
	ispettore 2: GetConfiguration
	ispettore 2: GetDelegatedAdminAccount
	ispettore 2: GetEc2DeepInspectionConfiguration
	ispettore 2: GetEncryptionKey
	ispettore 2: GetFindingsReportStatus
	ispettore 2: GetMember
	ispettore 2: GetSbomExport
	ispettore 2: ListAccountPermissions
	ispettore 2: ListCisScanConfigurations
	ispettore 2: ListCisScanResultsAggregatedByChecks
	ispettore 2: ListCisScanResultsAggregatedByTargetResource
	ispettore 2: ListCisScans

Prefisso del servizio	Azioni
	ispettore 2: ListCodeSecurityIntegrations
	ispettore 2: ListCodeSecurityScanConfigurations
	ispettore 2: ListConnectorScanConfigurations
	ispettore 2: ListConnectors
	ispettore 2: ListCoverage
	ispettore 2: ListCoverageStatistics
	ispettore 2: ListDelegatedAdminAccounts
	ispettore 2: ListFilters
	ispettore 2: ListFindingAggregations
	ispettore 2: ListFindings
	ispettore 2: ListMembers
	ispettore 2: ListUsageTotals
	ispettore 2: ResetEncryptionKey
	ispettore 2: SearchVulnerabilities
	ispettore 2: SendCisSessionHealth
	ispettore 2: SendCisSessionTelemetry
	ispettore 2: StartCisSession
	ispettore 2: StartCodeSecurityScan
	ispettore 2: StopCisSession
	ispettore 2: UpdateCisScanConfiguration
	ispettore 2: UpdateCodeSecurityIntegration

Prefisso del servizio	Azioni
	ispettore 2: UpdateConfiguration ispettore 2: UpdateConnector ispettore 2: UpdateEc2DeepInspectionConfiguration ispettore 2: UpdateEncryptionKey ispettore 2: UpdateFilter ispettore 2: UpdateOrgEc2DeepInspectionConfiguration ispettore 2: UpdateOrganizationConfiguration

Prefisso del servizio	Azioni
iot	IoT: AcceptCertificateTransfer
	IoT: AddThingToBillingGroup
	IoT: AddThingToThingGroup
	IoT: AssociateSbomWithPackageVersion
	IoT: AssociateTargetsWithJob
	IoT: AttachPolicy
	IoT: AttachPrincipalPolicy
	IoT: AttachSecurityProfile
	IoT: AttachThingPrincipal
	IoT: CancelAuditMitigationActionsTask
	IoT: CancelAuditTask
	IoT: CancelCertificateTransfer
	IoT: CancelDetectMitigationActionsTask
	IoT: CancelJob
	IoT: CancelJobExecution
	IoT: ClearDefaultAuthorizer
	IoT: ConfirmTopicRuleDestination
	IoT: CreateAuditSuppression
	IoT: CreateAuthorizer
	IoT: CreateBillingGroup
	IoT: CreateCertificateFromCsr

Prefisso del servizio	Azioni
	IoT: CreateCertificateProvider IoT: CreateCommand IoT: CreateCustomMetric IoT: CreateDimension IoT: CreateDomainConfiguration IoT: CreateDynamicThingGroup IoT: CreateFleetMetric IoT: CreateJob IoT: CreateJobTemplate IoT: CreateKeysAndCertificate IoT: CreateMitigationAction iot:CreateOTAUpdate IoT: CreatePackage IoT: CreatePackageVersion IoT: CreatePolicy IoT: CreatePolicyVersion IoT: CreateProvisioningClaim IoT: CreateProvisioningTemplate IoT: CreateProvisioningTemplateVersion IoT: CreateRoleAlias IoT: CreateScheduledAudit

Prefisso del servizio	Azioni
	IoT: CreateSecurityProfile IoT: CreateStream IoT: CreateThing IoT: CreateThingGroup IoT: CreateThingType IoT: CreateTopicRule IoT: CreateTopicRuleDestination IoT: DeleteAccountAuditConfiguration IoT: DeleteAuditSuppression IoT: DeleteAuthorizer IoT: DeleteBillingGroup iot:DeleteCACertificate IoT: DeleteCertificate IoT: DeleteCertificateProvider IoT: DeleteCommand IoT: DeleteCustomMetric IoT: DeleteDimension IoT: DeleteDomainConfiguration IoT: DeleteDynamicThingGroup IoT: DeleteFleetMetric IoT: DeleteJob

Prefisso del servizio	Azioni
	IoT: DeleteJobExecution
	IoT: DeleteJobTemplate
	IoT: DeleteMitigationAction
	iot:DeleteOTAUpdate
	IoT: DeletePackage
	IoT: DeletePackageVersion
	IoT: DeletePolicy
	IoT: DeletePolicyVersion
	IoT: DeleteProvisioningTemplate
	IoT: DeleteProvisioningTemplateVersion
	IoT: DeleteRegistrationCode
	IoT: DeleteRoleAlias
	IoT: DeleteScheduledAudit
	IoT: DeleteSecurityProfile
	IoT: DeleteStream
	IoT: DeleteThing
	IoT: DeleteThingGroup
	IoT: DeleteThingType
	IoT: DeleteTopicRule
	IoT: DeleteTopicRuleDestination
	IoT: DeleteV2LoggingLevel

Prefisso del servizio	Azioni
	IoT: DeprecateThingType IoT: DescribeAccountAuditConfiguration IoT: DescribeAuditFinding IoT: DescribeAuditMitigationActionsTask IoT: DescribeAuditSuppression IoT: DescribeAuditTask IoT: DescribeAuthorizer IoT: DescribeBillingGroup iot:DescribeCACertificate IoT: DescribeCertificate IoT: DescribeCertificateProvider IoT: DescribeCustomMetric IoT: DescribeDefaultAuthorizer IoT: DescribeDetectMitigationActionsTask IoT: DescribeDimension IoT: DescribeDomainConfiguration IoT: DescribeEncryptionConfiguration IoT: DescribeEndpoint IoT: DescribeEventConfigurations IoT: DescribeFleetMetric IoT: DescribeIndex

Prefisso del servizio	Azioni
	IoT: DescribeJob IoT: DescribeJobExecution IoT: DescribeJobTemplate IoT: DescribeManagedJobTemplate IoT: DescribeMitigationAction IoT: DescribeProvisioningTemplate IoT: DescribeProvisioningTemplateVersion IoT: DescribeRoleAlias IoT: DescribeScheduledAudit IoT: DescribeSecurityProfile IoT: DescribeStream IoT: DescribeThing IoT: DescribeThingGroup IoT: DescribeThingRegistrationTask IoT: DescribeThingType IoT: DetachPolicy IoT: DetachPrincipalPolicy IoT: DetachSecurityProfile IoT: DetachThingPrincipal IoT: DisableTopicRule IoT: DisassociateSbomFromPackageVersion

Prefisso del servizio	Azioni
	IoT: EnableTopicRule
	IoT: GetBehaviorModelTrainingSummaries
	IoT: GetBucketsAggregation
	IoT: GetCardinality
	IoT: GetCommand
	IoT: GetEffectivePolicies
	IoT: GetJobDocument
	IoT: GetLoggingOptions
	iot:GetOTAUpdate
	IoT: GetPackage
	IoT: GetPackageConfiguration
	IoT: GetPackageVersion
	IoT: GetPercentiles
	IoT: GetPolicy
	IoT: GetPolicyVersion
	IoT: GetRegistrationCode
	IoT: GetStatistics
	IoT: GetThingConnectivityData
	IoT: GetTopicRule
	IoT: GetTopicRuleDestination
	IoT: GetV2LoggingOptions

Prefisso del servizio	Azioni
	IoT: ListActiveViolations
	IoT: ListAttachedPolicies
	IoT: ListAuditFindings
	IoT: ListAuditMitigationActionsExecutions
	IoT: ListAuditMitigationActionsTasks
	IoT: ListAuditSuppressions
	IoT: ListAuditTasks
	IoT: ListAuthorizers
	IoT: ListBillingGroups
	iot:ListCACertificates
	IoT: ListCertificateProviders
	IoT: ListCertificates
	IoT: ListCertificatesByCA
	IoT: ListCommands
	IoT: ListCustomMetrics
	IoT: ListDetectMitigationActionsExecutions
	IoT: ListDetectMitigationActionsTasks
	IoT: ListDimensions
	IoT: ListDomainConfigurations
	IoT: ListFleetMetrics
	IoT: ListIndices

Prefisso del servizio	Azioni
	IoT: ListJobExecutionsForJob IoT: ListJobExecutionsForThing IoT: ListJobTemplates IoT: ListJobs IoT: ListManagedJobTemplates IoT: ListMetricValues IoT: ListMitigationActions iot:ListOTAUpdates IoT: ListOutgoingCertificates IoT: ListPackageVersions IoT: ListPackages IoT: ListPolicies IoT: ListPolicyPrincipals IoT: ListPolicyVersions IoT: ListPrincipalPolicies IoT: ListPrincipalThings IoT: ListPrincipalThingsV2 IoT: ListProvisioningTemplateVersions IoT: ListProvisioningTemplates IoT: ListRelatedResourcesForAuditFinding IoT: ListRoleAliases

Prefisso del servizio	Azioni
	IoT: ListSbomValidationResults
	IoT: ListScheduledAudits
	IoT: ListSecurityProfiles
	IoT: ListSecurityProfilesForTarget
	IoT: ListStreams
	IoT: ListTargetsForPolicy
	IoT: ListTargetsForSecurityProfile
	IoT: ListThingGroups
	IoT: ListThingGroupsForThing
	IoT: ListThingPrincipals
	IoT: ListThingPrincipalsV2
	IoT: ListThingRegistrationTaskReports
	IoT: ListThingRegistrationTasks
	IoT: ListThingTypes
	IoT: ListThings
	IoT: ListThingsInBillingGroup
	IoT: ListThingsInThingGroup
	IoT: ListTopicRuleDestinations
	IoT: ListTopicRules
	IoT: ListV2LoggingLevels
	IoT: ListViolationEvents

Prefisso del servizio	Azioni
	IoT: PutVerificationStateOnViolation iot:RegisterCACertificate IoT: RegisterCertificate IoT: RegisterCertificateWithoutCA IoT: RegisterThing IoT: RejectCertificateTransfer IoT: RemoveThingFromBillingGroup IoT: RemoveThingFromThingGroup IoT: ReplaceTopicRule IoT: SearchIndex IoT: SetDefaultAuthorizer IoT: SetDefaultPolicyVersion IoT: SetLoggingOptions IoT: SetV2LoggingLevel IoT: SetV2LoggingOptions IoT: StartAuditMitigationActionsTask IoT: StartDetectMitigationActionsTask IoT: StartOnDemandAuditTask IoT: StartThingRegistrationTask IoT: StopThingRegistrationTask IoT: TestAuthorization

Prefisso del servizio	Azioni
	IoT: TestInvokeAuthorizer IoT: TransferCertificate IoT: UpdateAccountAuditConfiguration IoT: UpdateAuditSuppression IoT: UpdateAuthorizer IoT: UpdateBillingGroup iot:UpdateCACertificate IoT: UpdateCertificate IoT: UpdateCertificateProvider IoT: UpdateCommand IoT: UpdateCustomMetric IoT: UpdateDimension IoT: UpdateDomainConfiguration IoT: UpdateDynamicThingGroup IoT: UpdateEncryptionConfiguration IoT: UpdateEventConfigurations IoT: UpdateFleetMetric IoT: UpdateIndexingConfiguration IoT: UpdateJob IoT: UpdateMitigationAction IoT: UpdatePackage

Prefisso del servizio	Azioni
	IoT: UpdatePackageConfiguration IoT: UpdatePackageVersion IoT: UpdateProvisioningTemplate IoT: UpdateRoleAlias IoT: UpdateScheduledAudit IoT: UpdateSecurityProfile IoT: UpdateStream IoT: UpdateThing IoT: UpdateThingGroup IoT: UpdateThingGroupsForThing IoT: UpdateThingType IoT: UpdateTopicRuleDestination IoT: ValidateSecurityProfileBehaviors

Prefisso del servizio	Azioni
iotanalytics	analisi IoT: CancelPipelineReprocessing
	analisi IoT: CreateChannel
	analisi IoT: CreateDataset
	analisi IoT: CreateDatasetContent
	analisi IoT: CreateDatastore
	analisi IoT: CreatePipeline
	analisi IoT: DeleteChannel
	analisi IoT: DeleteDataset
	analisi IoT: DeleteDatasetContent
	analisi IoT: DeleteDatastore
	analisi IoT: DeletePipeline
	analisi IoT: DescribeChannel
	analisi IoT: DescribeDataset
	analisi IoT: DescribeDatastore
	analisi IoT: DescribeLoggingOptions
	analisi IoT: DescribePipeline
	analisi IoT: GetDatasetContent
	analisi IoT: ListChannels
	analisi IoT: ListDatasetContents
	analisi IoT: ListDatasets
	analisi IoT: ListDatastores

Prefisso del servizio	Azioni
	analisi IoT: ListPipelines analisi IoT: PutLoggingOptions analisi IoT: RunPipelineActivity analisi IoT: SampleChannelData analisi IoT: StartPipelineReprocessing analisi IoT: UpdateChannel analisi IoT: UpdateDataset analisi IoT: UpdateDatastore analisi IoT: UpdatePipeline
iotdeviceadvisor	consulente per dispositivi IoT: CreateSuiteDefinition consulente per dispositivi iot: DeleteSuiteDefinition consulente per dispositivi iot: GetEndpoint consulente per dispositivi iot: GetSuiteDefinition consulente per dispositivi iot: GetSuiteRun consulente per dispositivi iot: GetSuiteRunReport consulente per dispositivi iot: ListSuiteDefinitions consulente per dispositivi iot: ListSuiteRuns consulente per dispositivi iot: StartSuiteRun consulente per dispositivi iot: StopSuiteRun consulente per dispositivi iot: UpdateSuiteDefinition

Prefisso del servizio	Azioni
iotevents	eventi IoT: BatchAcknowledgeAlarm
	eventi IoT: BatchDeleteDetector
	eventi IoT: BatchDisableAlarm
	eventi IoT: BatchEnableAlarm
	eventi IoT: BatchResetAlarm
	eventi IoT: BatchSnoozeAlarm
	eventi IoT: BatchUpdateDetector
	eventi IoT: CreateAlarmModel
	eventi IoT: CreateDetectorModel
	eventi IoT: CreateInput
	eventi IoT: DeleteAlarmModel
	eventi IoT: DeleteDetectorModel
	eventi IoT: DeleteInput
	eventi IoT: DescribeAlarm
	eventi IoT: DescribeAlarmModel
	eventi IoT: DescribeDetector
	eventi IoT: DescribeDetectorModel
	eventi IoT: DescribeDetectorModelAnalysis
	eventi IoT: DescribeInput
	eventi IoT: DescribeLoggingOptions
	eventi IoT: GetDetectorModelAnalysisResults

Prefisso del servizio	Azioni
	eventi IoT: ListAlarmModelVersions eventi IoT: ListAlarmModels eventi IoT: ListAlarms eventi IoT: ListDetectorModelVersions eventi IoT: ListDetectorModels eventi IoT: ListDetectors eventi IoT: ListInputRoutings eventi IoT: ListInputs eventi IoT: PutLoggingOptions eventi IoT: StartDetectorModelAnalysis eventi IoT: UpdateAlarmModel eventi IoT: UpdateDetectorModel eventi IoT: UpdateInput
iotfleethub	hub iotfleet: CreateApplication hub iotfleet: DeleteApplication hub iotfleet: DescribeApplication hub iotfleet: ListApplications hub iotfleet: UpdateApplication

Prefisso del servizio	Azioni
iotsitewise	per quanto riguarda il sito IoT: AssociateAssets
	per quanto riguarda il sito IoT: AssociateTimeSeriesToAssetProperty
	per quanto riguarda il sito IoT: BatchAssociateProjectAssets
	per quanto riguarda il sito IoT: BatchDisassociateProjectAssets
	per quanto riguarda il sito IoT: CreateAccessPolicy
	per quanto riguarda il sito IoT: CreateAsset
	per quanto riguarda il sito IoT: CreateAssetModel
	per quanto riguarda il sito IoT: CreateAssetModelCompositeModel
	a livello di sito IoT: CreateBulkImportJob
	a livello di sito IoT: CreateComputationModel
	a livello di sito IoT: CreateDashboard
	a livello di sito IoT: CreateDataset
	a livello di sito IoT: CreateGateway
	a livello di sito IoT: CreatePortal
	a livello di sito IoT: CreateProject
	a livello di sito IoT: DeleteAccessPolicy
	a livello di sito IoT: DeleteAsset
	a livello di sito IoT: DeleteAssetModel
	a livello di sito IoT: DeleteAssetModelCompositeModel
	a livello di sito IoT: DeleteComputationModel

Prefisso del servizio	Azioni
	a livello di sito IoT: DeleteDashboard
	a livello di sito IoT: DeleteDataset
	a livello di sito IoT: DeleteGateway
	a livello di sito IoT: DeletePortal
	a livello di sito IoT: DeleteProject
	a livello di sito IoT: DeleteTimeSeries
	a livello di sito IoT: DescribeAccessPolicy
	a livello di sito IoT: DescribeAsset
	a livello di sito IoT: DescribeAssetCompositeModel
	a livello di sito IoT: DescribeAssetModel
	a livello di sito IoT: DescribeAssetModelCompositeModel
	a livello di sito IoT: DescribeAssetModelInterfaceRelationship
	a livello di sito IoT: DescribeAssetProperty
	a livello di sito IoT: DescribeBulkImportJob
	a livello di sito IoT: DescribeComputationModel
	a livello di sito IoT: DescribeComputationModelExecutionSummary
	a livello di sito IoT: DescribeDashboard
	a livello di sito IoT: DescribeDataset
	a livello di sito IoT: DescribeDefaultEncryptionConfiguration
	a livello di sito IoT: DescribeExecution
	a livello di sito IoT: DescribeGateway

Prefisso del servizio	Azioni
	a livello di sito IoT: DescribeGatewayCapabilityConfiguration
	a livello di sito IoT: DescribeLoggingOptions
	a livello di sito IoT: DescribePortal
	a livello di sito IoT: DescribeProject
	a livello di sito IoT: DescribeStorageConfiguration
	a livello di sito IoT: DescribeTimeSeries
	a livello di sito IoT: DisassociateAssets
	a livello di sito IoT: DisassociateTimeSeriesFromAssetProperty
	a livello di sito IoT: ExecuteAction
	a livello di sito IoT: ExecuteQuery
	a livello di sito IoT: ListAccessPolicies
	a livello di sito IoT: ListActions
	a livello di sito IoT: ListAssetModelCompositeModels
	a livello di sito IoT: ListAssetModelProperties
	a livello di sito IoT: ListAssetModels
	a livello di sito IoT: ListAssetProperties
	a livello di sito IoT: ListAssetRelationships
	a livello di sito IoT: ListAssets
	a livello di sito IoT: ListAssociatedAssets
	a livello di sito IoT: ListBulkImportJobs
	a livello di sito IoT: ListCompositionRelationships

Prefisso del servizio	Azioni
	a livello di sito IoT: ListComputationModelDataBindingUsages
	a livello di sito IoT: ListComputationModelResolveToResources
	a livello di sito IoT: ListComputationModels
	a livello di sito IoT: ListDashboards
	a livello di sito IoT: ListDatasets
	a livello di sito IoT: ListExecutions
	a livello di sito IoT: ListGateways
	a livello di sito IoT: ListInterfaceRelationships
	a livello di sito IoT: ListPortals
	a livello di sito IoT: ListProjectAssets
	a livello di sito IoT: ListProjects
	a livello di sito IoT: ListTimeSeries
	a livello di sito IoT: PutDefaultEncryptionConfiguration
	a livello di sito IoT: PutLoggingOptions
	a livello di sito IoT: PutStorageConfiguration
	a livello di sito IoT: UpdateAccessPolicy
	a livello di sito IoT: UpdateAsset
	a livello di sito IoT: UpdateAssetModel
	a livello di sito IoT: UpdateAssetModelCompositeModel
	a livello di sito IoT: UpdateAssetProperty
	a livello di sito IoT: UpdateComputationModel

Prefisso del servizio	Azioni
	a livello di sito IoT: UpdateDashboard
	a livello di sito IoT: UpdateDataset
	a livello di sito IoT: UpdateGateway
	a livello di sito IoT: UpdateGatewayCapabilityConfiguration
	a livello di sito IoT: UpdatePortal
	a livello di sito IoT: UpdateProject

Prefisso del servizio	Azioni
iottwinmaker	iottwinmaker: CancelMetadataTransferJob iottwinmaker: CreateComponentType iottwinmaker: CreateEntity iottwinmaker: CreateMetadataTransferJob iottwinmaker: CreateScene iottwinmaker: CreateSyncJob iottwinmaker: CreateWorkspace iottwinmaker: DeleteComponentType iottwinmaker: DeleteEntity iottwinmaker: DeleteScene iottwinmaker: DeleteSyncJob iottwinmaker: DeleteWorkspace iottwinmaker: ExecuteQuery iottwinmaker: GetMetadataTransferJob iottwinmaker: GetPricingPlan iottwinmaker: GetScene iottwinmaker: GetSyncJob iottwinmaker: ListComponentTypes iottwinmaker: ListComponents iottwinmaker: ListEntities iottwinmaker: ListMetadataTransferJobs

Prefisso del servizio	Azioni
	iottwinmaker: ListProperties iottwinmaker: ListScenes iottwinmaker: ListSyncJobs iottwinmaker: ListSyncResources iottwinmaker: ListWorkspaces iottwinmaker: UpdateComponentType iottwinmaker: UpdateEntity iottwinmaker: UpdatePricingPlan iottwinmaker: UpdateScene iottwinmaker: UpdateWorkspace

Prefisso del servizio	Azioni
iotwireless	IoT senza fili: AssociateAwsAccountWithPartnerAccount
	IoT wireless: AssociateMulticastGroupWithFuotaTask
	IoT wireless: AssociateWirelessDeviceWithFuotaTask
	IoT wireless: AssociateWirelessDeviceWithMulticastGroup
	IoT wireless: AssociateWirelessDeviceWithThing
	IoT wireless: AssociateWirelessGatewayWithCertificate
	IoT wireless: AssociateWirelessGatewayWithThing
	IoT wireless: CancelMulticastGroupSession
	IoT wireless: CreateDestination
	IoT wireless: CreateDeviceProfile
	IoT wireless: CreateFuotaTask
	IoT wireless: CreateMulticastGroup
	IoT wireless: CreateNetworkAnalyzerConfiguration
	IoT wireless: CreateServiceProfile
	IoT wireless: CreateWirelessDevice
	IoT wireless: CreateWirelessGateway
	IoT wireless: CreateWirelessGatewayTask
	IoT wireless: CreateWirelessGatewayTaskDefinition
	IoT wireless: DeleteDestination
	IoT wireless: DeleteDeviceProfile
	IoT wireless: DeleteFuotaTask

Prefisso del servizio	Azioni
	IoT wireless: DeleteMulticastGroup
	IoT wireless: DeleteNetworkAnalyzerConfiguration
	IoT wireless: DeleteQueuedMessages
	IoT wireless: DeleteServiceProfile
	IoT wireless: DeleteWirelessDevice
	IoT wireless: DeleteWirelessDeviceImportTask
	IoT wireless: DeleteWirelessGateway
	IoT wireless: DeleteWirelessGatewayTask
	IoT wireless: DeleteWirelessGatewayTaskDefinition
	IoT wireless: DeregisterWirelessDevice
	IoT wireless: DisassociateAwsAccountFromPartnerAccount
	IoT wireless: DisassociateMulticastGroupFromFuotaTask
	IoT wireless: DisassociateWirelessDeviceFromFuotaTask
	IoT wireless: DisassociateWirelessDeviceFromMulticastGroup
	IoT wireless: DisassociateWirelessDeviceFromThing
	IoT wireless: DisassociateWirelessGatewayFromCertificate
	IoT wireless: DisassociateWirelessGatewayFromThing
	IoT wireless: GetDestination
	IoT wireless: GetDeviceProfile
	IoT wireless: GetEventConfigurationByResourceTypes
	IoT wireless: GetFuotaTask

Prefisso del servizio	Azioni
	IoT wireless: GetLogLevelsByResourceTypes
	IoT wireless: GetMetricConfiguration
	IoT wireless: GetMetrics
	IoT wireless: GetMulticastGroup
	IoT wireless: GetMulticastGroupSession
	IoT wireless: GetNetworkAnalyzerConfiguration
	IoT wireless: GetPartnerAccount
	IoT wireless: GetPosition
	IoT wireless: GetPositionConfiguration
	IoT wireless: GetPositionEstimate
	IoT wireless: GetResourceEventConfiguration
	IoT wireless: GetResourceLogLevel
	IoT wireless: GetResourcePosition
	IoT wireless: GetServiceEndpoint
	IoT wireless: GetServiceProfile
	IoT wireless: GetWirelessDevice
	IoT wireless: GetWirelessDeviceImportTask
	IoT wireless: GetWirelessDeviceStatistics
	IoT wireless: GetWirelessGateway
	IoT wireless: GetWirelessGatewayCertificate
	IoT wireless: GetWirelessGatewayFirmwareInformation

Prefisso del servizio	Azioni
	IoT wireless: GetWirelessGatewayStatistics
	IoT wireless: GetWirelessGatewayTask
	IoT wireless: GetWirelessGatewayTaskDefinition
	IoT wireless: ListDestinations
	IoT wireless: ListDeviceProfiles
	IoT wireless: ListDevicesForWirelessDeviceImportTask
	IoT wireless: ListEventConfigurations
	IoT wireless: ListFuotaTasks
	IoT wireless: ListMulticastGroups
	IoT wireless: ListMulticastGroupsByFuotaTask
	IoT wireless: ListNetworkAnalyzerConfigurations
	IoT wireless: ListPartnerAccounts
	IoT wireless: ListPositionConfigurations
	IoT wireless: ListQueuedMessages
	IoT wireless: ListServiceProfiles
	IoT wireless: ListWirelessDeviceImportTasks
	IoT wireless: ListWirelessDevices
	IoT wireless: ListWirelessGatewayTaskDefinitions
	IoT wireless: ListWirelessGateways
	IoT wireless: PutPositionConfiguration
	IoT wireless: PutResourceLogLevel

Prefisso del servizio	Azioni
	IoT wireless: ResetAllResourceLogLevels
	IoT wireless: ResetResourceLogLevel
	IoT wireless: SendDataToMulticastGroup
	IoT wireless: SendDataToWirelessDevice
	IoT wireless: StartBulkAssociateWirelessDeviceWithMulticastGroup
	IoT wireless: StartBulkDisassociateWirelessDeviceFromMulticastGroup
	IoT wireless: StartFuotaTask
	IoT wireless: StartMulticastGroupSession
	IoT wireless: StartNetworkAnalyzerStream
	IoT wireless: StartSingleWirelessDeviceImportTask
	IoT wireless: StartWirelessDeviceImportTask
	IoT wireless: TestWirelessDevice
	IoT wireless: UpdateDestination
	IoT wireless: UpdateEventConfigurationByResourceTypes
	IoT wireless: UpdateFuotaTask
	IoT wireless: UpdateLogLevelsByResourceTypes
	IoT wireless: UpdateMetricConfiguration
	IoT wireless: UpdateMulticastGroup
	IoT wireless: UpdateNetworkAnalyzerConfiguration
	IoT wireless: UpdatePartnerAccount

Prefisso del servizio	Azioni
	IoT wireless: UpdatePosition
	IoT wireless: UpdateResourceEventConfiguration
	IoT wireless: UpdateResourcePosition
	IoT wireless: UpdateWirelessDevice
	IoT wireless: UpdateWirelessDeviceImportTask
	IoT wireless: UpdateWirelessGateway

Prefisso del servizio	Azioni
ivs	è: BatchGetChannel è: BatchGetStreamKey è: BatchStartViewerSessionRevocation è: CreateAdConfiguration è: CreateChannel è: CreateEncoderConfiguration è: CreateIngestConfiguration è: CreateParticipantToken è: CreatePlaybackRestrictionPolicy è: CreateRecordingConfiguration è: CreateStorageConfiguration è: CreateStreamKey è: DeleteAdConfiguration è: DeleteChannel è: DeleteEncoderConfiguration è: DeleteIngestConfiguration è: DeletePlaybackKeyPair è: DeletePlaybackRestrictionPolicy è: DeletePublicKey è: DeleteRecordingConfiguration è: DeleteStorageConfiguration

Prefisso del servizio	Azioni
	è: DeleteStreamKey è: DisconnectParticipant è: GetAdConfiguration è: GetChannel è: GetComposition è: GetEncoderConfiguration è: GetIngestConfiguration è: GetParticipant è: GetPlaybackKeyPair è: GetPlaybackRestrictionPolicy è: GetPublicKey è: GetRecordingConfiguration è: GetStorageConfiguration è: GetStream è: GetStreamKey è: GetStreamSession è: ImportPlaybackKeyPair è: ImportPublicKey è: InsertAdBreak è: ListAdConfigurations è: ListChannels

Prefisso del servizio	Azioni
	è: ListCompositions è: ListEncoderConfigurations è: ListIngestConfigurations è: ListParticipantEvents è: ListParticipantReplicas è: ListParticipants è: ListPlaybackKeyPairs è: ListPlaybackRestrictionPolicies è: ListPublicKeys è: ListRecordingConfigurations è: ListStorageConfigurations è: ListStreamKeys è: ListStreamSessions è: ListStreams è: PutMetadata è: StartComposition è: StartViewerSessionRevocation è: StopComposition è: StopStream è: UpdateAdConfiguration è: UpdateChannel

Prefisso del servizio	Azioni
	è: UpdateIngestConfiguration è: UpdatePlaybackRestrictionPolicy
ivschat	visto: CreateChatToken vista chat: CreateLoggingConfiguration vista chat: CreateRoom vista chat: DeleteLoggingConfiguration vista chat: DeleteMessage vista chat: DeleteRoom vista chat: DisconnectUser vista chat: GetLoggingConfiguration vista chat: GetRoom vista chat: ListLoggingConfigurations vista chat: ListRooms vista chat: SendEvent vista chat: UpdateLoggingConfiguration vista chat: UpdateRoom

Prefisso del servizio	Azioni
kafka	kafka: BatchAssociateScramSecret caffè: BatchDisassociateScramSecret caffè: CreateCluster kafka: V2 CreateCluster kafka: CreateConfiguration caffè: CreateReplicator caffè: CreateVpcConnection caffè: DeleteCluster caffè: DeleteClusterPolicy caffè: DeleteConfiguration caffè: DeleteReplicator caffè: DeleteVpcConnection caffè: DescribeCluster caffè: DescribeClusterOperation caffè: DescribeClusterOperationV2 kafka: V2 DescribeCluster kafka: DescribeConfiguration caffè: DescribeConfigurationRevision caffè: DescribeVpcConnection caffè: GetBootstrapBrokers caffè: GetClusterPolicy

Prefisso del servizio	Azioni
	caffè: GetCompatibleKafkaVersions caffè: ListClientVpcConnections caffè: ListClusterOperations caffè: ListClusterOperationsV2 caffè: ListClusters kafka: V2 ListClusters kafka: ListConfigurationRevisions caffè: ListConfigurations caffè: ListKafkaVersions caffè: ListNodes caffè: ListReplicators caffè: ListScramSecrets caffè: ListVpcConnections caffè: PutClusterPolicy caffè: RebootBroker caffè: RejectClientVpcConnection caffè: UpdateBrokerCount caffè: UpdateBrokerStorage caffè: UpdateBrokerType caffè: UpdateClusterConfiguration caffè: UpdateClusterKafkaVersion

Prefisso del servizio	Azioni
	caffè: UpdateConfiguration caffè: UpdateConnectivity caffè: UpdateMonitoring caffè: UpdateRebalancing caffè: UpdateReplicationInfo caffè: UpdateSecurity caffè: UpdateStorage
kafkaconnect	connessione kafka: CreateConnector connessione kafka: CreateCustomPlugin connessione kafka: CreateWorkerConfiguration connessione kafka: DeleteConnector connessione kafka: DeleteCustomPlugin connessione kafka: DeleteWorkerConfiguration connessione kafka: DescribeConnector connessione kafka: DescribeCustomPlugin connessione kafka: DescribeWorkerConfiguration connessione kafka: ListConnectorOperations connessione kafka: ListConnectors connessione kafka: ListCustomPlugins connessione kafka: ListWorkerConfigurations connessione kafka: UpdateConnector

Prefisso del servizio	Azioni
kendra	kendra: AssociateEntitiesToExperience kendra: AssociatePersonasToEntities kendra: BatchDeleteDocument kendra: BatchDeleteFeaturedResultsSet kendra: BatchGetDocumentStatus kendra: BatchPutDocument kendra: ClearQuerySuggestions kendra: CreateAccessControlConfiguration kendra: CreateDataSource kendra: CreateExperience kendra: CreateFaq kendra: CreateFeaturedResultsSet kendra: CreateIndex kendra: CreateQuerySuggestionsBlockList kendra: CreateThesaurus kendra: DeleteDataSource kendra: DeleteExperience kendra: DeleteFaq kendra: DeleteIndex kendra: DeletePrincipalMapping kendra: DeleteQuerySuggestionsBlockList

Prefisso del servizio	Azioni
	kendra: DeleteThesaurus kendra: DescribeAccessControlConfiguration kendra: DescribeDataSource kendra: DescribeExperience kendra: DescribeFaq kendra: DescribeFeaturedResultsSet kendra: DescribeIndex kendra: DescribePrincipalMapping kendra: DescribeQuerySuggestionsBlockList kendra: DescribeQuerySuggestionsConfig kendra: DescribeThesaurus kendra: DisassociateEntitiesFromExperience kendra: DisassociatePersonasFromEntities kendra: GetQuerySuggestions kendra: GetSnapshots kendra: ListAccessControlConfigurations kendra: ListDataSourceSyncJobs kendra: ListDataSources kendra: ListEntityPersonas kendra: ListExperienceEntities kendra: ListExperiences

Prefisso del servizio	Azioni
	kendra: ListFaqs kendra: ListFeaturedResultsSets kendra: ListGroupsOlderThanOrderingId kendra: ListIndices kendra: ListQuerySuggestionsBlockLists kendra: ListThesauri kendra: PutPrincipalMapping kendra:Query kendra:Retrieve kendra: StartDataSourceSyncJob kendra: StopDataSourceSyncJob kendra: SubmitFeedback kendra: UpdateDataSource kendra: UpdateExperience kendra: UpdateFeaturedResultsSet kendra: UpdateIndex kendra: UpdateQuerySuggestionsBlockList kendra: UpdateQuerySuggestionsConfig kendra: UpdateThesaurus

Prefisso del servizio	Azioni
kinesis	cinesi: CreateStream
	cinesi: DecreaseStreamRetentionPeriod
	cinesi: DeleteStream
	cinesi: DeregisterStreamConsumer
	cinesi: DescribeAccountSettings
	cinesi: DescribeLimits
	cinesi: DescribeStream
	cinesi: DescribeStreamConsumer
	cinesi: DescribeStreamSummary
	cinesi: DisableEnhancedMonitoring
	cinesi: EnableEnhancedMonitoring
	cinesi: IncreaseStreamRetentionPeriod
	cinesi: ListShards
	cinesi: ListStreamConsumers
	cinesi: ListStreams
	cinesi: MergeShards
	cinesi: RegisterStreamConsumer
	cinesi: SplitShard
	cinesi: StartStreamEncryption
	cinesi: StopStreamEncryption
	cinesi: UpdateAccountSettings

Prefisso del servizio	Azioni
	cinesi: UpdateShardCount cinesi: UpdateStreamMode

Prefisso del servizio	Azioni
kinesisanalytics	analisi della cinesi: AddApplicationCloudWatchLoggingOption kinesisanalytics: AddApplicationInput kinesisanalytics: AddApplicationInputProcessingConfiguration kinesisanalytics: AddApplicationOutput kinesisanalytics: AddApplicationReferenceDataSource kinesisanalytics: AddApplicationVpcConfiguration kinesisanalytics: CreateApplication kinesisanalytics: CreateApplicationPresignedUrl kinesisanalytics: CreateApplicationSnapshot kinesisanalytics: DeleteApplication kinesisanalytics: DeleteApplicationCloudWatchLoggingOption kinesisanalytics: DeleteApplicationInputProcessingConfiguration kinesisanalytics: DeleteApplicationOutput kinesisanalytics: DeleteApplicationReferenceDataSource kinesisanalytics: DeleteApplicationSnapshot kinesisanalytics: DeleteApplicationVpcConfiguration kinesisanalytics: DescribeApplication kinesisanalytics: DescribeApplicationOperation kinesisanalytics: DescribeApplicationSnapshot kinesisanalytics: DescribeApplicationVersion kinesisanalytics: DiscoverInputSchema

Prefisso del servizio	Azioni
	kinesisanalytics: ListApplicationOperations kinesisanalytics: ListApplicationSnapshots kinesisanalytics: ListApplicationVersions kinesisanalytics: ListApplications kinesisanalytics: RollbackApplication kinesisanalytics: StartApplication kinesisanalytics: StopApplication kinesisanalytics: UpdateApplication kinesisanalytics: UpdateApplicationMaintenanceConfiguration

Prefisso del servizio	Azioni
kms	kms:CancelKeyDeletion
	kms:ConnectCustomKeyStore
	kms:CreateAlias
	kms:CreateCustomKeyStore
	kms:CreateGrant
	kms:CreateKey
	kms:Decrypt
	kms:DeleteAlias
	kms:DeleteCustomKeyStore
	kms:DeleteImportedKeyMaterial
	kms:DeriveSharedSecret
	kms:DescribeCustomKeyStores
	kms:DescribeKey
	kms:DisableKey
	kms:DisableKeyRotation
	kms:DisconnectCustomKeyStore
	kms:EnableKey
	kms:EnableKeyRotation
	kms:Encrypt
	kms:GenerateDataKey
	kms:GenerateDataKeyPair

Prefisso del servizio	Azioni
	kms:GenerateDataKeyPairWithoutPlaintext
	kms:GenerateDataKeyWithoutPlaintext
	kms:GenerateMac
	kms:GenerateRandom
	kms:GetKeyLastUsage
	kms:GetKeyPolicy
	kms:GetKeyRotationStatus
	kms:GetParametersForImport
	kms:GetPublicKey
	kms:ImportKeyMaterial
	kms:ListAliases
	kms:ListGrants
	kms:ListKeyPolicies
	kms:ListKeyRotations
	kms:ListKeys
	kms:ListRetirableGrants
	kms:ReplicateKey
	kms:RetireGrant
	kms:RevokeGrant
	kms:RotateKeyOnDemand
	kms:ScheduleKeyDeletion

Prefisso del servizio	Azioni
	kms:Sign kms:UpdateAlias kms:UpdateCustomKeyStore kms:UpdateKeyDescription kms:UpdatePrimaryRegion kms:Verify kms:VerifyMac

Prefisso del servizio	Azioni
lambda	lambda: AddLayerVersionPermission lambda: AddPermission lambda: CreateAlias lambda: CreateCodeSigningConfig lambda: CreateEventSourceMapping lambda: CreateFunction lambda: CreateFunctionUrlConfig lambda: DeleteAlias lambda: DeleteCapacityProvider lambda: DeleteCodeSigningConfig lambda: DeleteEventSourceMapping lambda: DeleteFunction lambda: DeleteFunctionCodeSigningConfig lambda: DeleteFunctionConcurrency lambda: DeleteFunctionEventInvokeConfig lambda: DeleteFunctionUrlConfig lambda: DeleteLayerVersion lambda: DeleteProvisionedConcurrencyConfig lambda: GetAccountSettings lambda: GetAlias lambda: GetCapacityProvider

Prefisso del servizio	Azioni
	lambda: GetCodeSigningConfig lambda: GetEventSourceMapping lambda: GetFunction lambda: GetFunctionCodeSigningConfig lambda: GetFunctionConcurrency lambda: GetFunctionConfiguration lambda: GetFunctionEventInvokeConfig lambda: GetFunctionRecursionConfig lambda: GetFunctionScalingConfig lambda: GetFunctionUrlConfig lambda: GetLayerVersion lambda: GetLayerVersionPolicy lambda: GetPolicy lambda: GetProvisionedConcurrencyConfig lambda: GetRuntimeManagementConfig lambda: ListAliases lambda: ListCapacityProviders lambda: ListCodeSigningConfigs lambda: ListDurableExecutionsByFunction lambda: ListEventSourceMappings lambda: ListFunctionEventInvokeConfigs

Prefisso del servizio	Azioni
	lambda: ListFunctionUrlConfigs lambda: ListFunctions lambda: ListFunctionsByCodeSigningConfig lambda: ListLayerVersions lambda: ListLayers lambda: ListProvisionedConcurrencyConfigs lambda: ListVersionsByFunction lambda: PublishLayerVersion lambda: PublishVersion lambda: PutFunctionCodeSigningConfig lambda: PutFunctionConcurrency lambda: PutFunctionEventInvokeConfig lambda: PutFunctionRecursionConfig lambda: PutFunctionScalingConfig lambda: PutProvisionedConcurrencyConfig lambda: PutRuntimeManagementConfig lambda: RemoveLayerVersionPermission lambda: RemovePermission lambda: UpdateAlias lambda: UpdateCapacityProvider lambda: UpdateCodeSigningConfig

Prefisso del servizio	Azioni
	lambda: UpdateEventSourceMapping lambda: UpdateFunctionCode lambda: UpdateFunctionConfiguration lambda: UpdateFunctionEventInvokeConfig lambda: UpdateFunctionUrlConfig

Prefisso del servizio	Azioni
lex	lex: BatchCreateCustomVocabularyItem
	lex: BatchDeleteCustomVocabularyItem
	lex: BatchUpdateCustomVocabularyItem
	lex: BuildBotLocale
	lex: CreateBotAlias
	lex: CreateBotReplica
	lex: CreateBotVersion
	lex: CreateExport
	lex: CreateIntentVersion
	lex: CreateResourcePolicy
	lex: CreateSlotTypeVersion
	lex: CreateTestSetDiscrepancyReport
	lex: CreateUploadUrl
	lex: DeleteBot
	lex: DeleteBotAnalyzerRecommendation
	lex: DeleteBotChannelAssociation
	lex: DeleteBotReplica
	lex: DeleteExport
	lex: DeleteImport
	lex: DeleteIntentVersion
	lex: DeleteResourcePolicy

Prefisso del servizio	Azioni
	lex: DeleteSlotTypeVersion
	lex: DeleteTestSet
	lex: DeleteUtterances
	lex: DescribeBotAlias
	lex: DescribeBotAnalyzerRecommendation
	lex: DescribeBotRecommendation
	lex: DescribeBotReplica
	lex: DescribeBotResourceGeneration
	lex: DescribeBotVersion
	lex: DescribeCustomVocabularyMetadata
	lex: DescribeExport
	lex: DescribeImport
	lex: DescribeResourcePolicy
	lex: DescribeTestExecution
	lex: DescribeTestSet
	lex: DescribeTestSetDiscrepancyReport
	lex: DescribeTestSetGeneration
	lex: GenerateBotElement
	lex: GetBot
	lex: GetBotAlias
	lex: GetBotAliases

Prefisso del servizio	Azioni
	lex: GetBotChannelAssociation
	lex: GetBotChannelAssociations
	lex: GetBotVersions
	lex: GetBots
	lex: GetBuiltinIntent
	lex: GetBuiltinIntents
	lex: GetBuiltinSlotTypes
	lex: GetExport
	lex: GetImport
	lex: GetIntent
	lex: GetIntentVersions
	lex: GetIntents
	lex: GetMigration
	lex: GetMigrations
	lex: GetSlotType
	lex: GetSlotTypeVersions
	lex: GetSlotTypes
	lex: GetTestExecutionArtifactsUrl
	lex: GetUtterancesView
	lex: ListBotAliasReplicas
	lex: ListBotAliases

Prefisso del servizio	Azioni
	lex: ListBotRecommendations
	lex: ListBotReplicas
	lex: ListBotResourceGenerations
	lex: ListBotVersionReplicas
	lex: ListBotVersions
	lex: ListBots
	lex: ListBuiltInIntents
	lex: ListBuiltInSlotTypes
	lex: ListCustomVocabularyItems
	lex: ListExports
	lex: ListImports
	lex: ListIntentMetrics
	lex: ListIntentPaths
	lex: ListRecommendedIntents
	lex: ListSessionAnalyticsData
	lex: ListSessionMetrics
	lex: ListTestExecutionResultItems
	lex: ListTestExecutions
	lex: ListTestSets
	lex: PutBot
	lex: PutBotAlias

Prefisso del servizio	Azioni
	lex: PutIntent
	lex: PutSlotType
	lex: SearchAssociatedTranscripts
	lex: StartBotAnalyzer
	lex: StartBotRecommendation
	lex: StartImport
	lex: StartMigration
	lex: StartTestExecution
	lex: StartTestSetGeneration
	lex: StopBotAnalyzer
	lex: StopBotRecommendation
	lex: UpdateBotAlias
	lex: UpdateBotRecommendation
	lex: UpdateExport
	lex: UpdateResourcePolicy

Prefisso del servizio	Azioni
license-manager-linux-subscriptions	abbonamenti license-manager-linux-: DeregisterSubscriptionProvider abbonamenti license-manager-linux-: GetRegisteredSubscriptionProvider abbonamenti license-manager-linux-: GetServiceSettings abbonamenti license-manager-linux-: ListLinuxSubscriptionInstances abbonamenti license-manager-linux-: ListLinuxSubscriptions abbonamenti license-manager-linux-: ListRegisteredSubscriptionProviders abbonamenti license-manager-linux-: RegisterSubscriptionProvider abbonamenti license-manager-linux-: UpdateServiceSettings

Prefisso del servizio	Azioni
lightsail	vela leggera: AllocateStaticIp
	vela leggera: AttachCertificateToDistribution
	vela leggera: AttachDisk
	vela leggera: AttachInstancesToLoadBalancer
	vela leggera: AttachLoadBalancerTlsCertificate
	vela leggera: AttachStaticIp
	vela leggera: CloseInstancePublicPorts
	vela leggera: CopySnapshot
	vela leggera: CreateBucket
	vela leggera: CreateBucketAccessKey
	vela leggera: CreateCertificate
	vela leggera: CreateCloudFormationStack
	vela leggera: CreateContactMethod
	vela leggera: CreateContainerService
	vela leggera: CreateContainerServiceDeployment
	vela leggera: CreateContainerServiceRegistryLogin
	vela leggera: CreateDisk
	vela leggera: CreateDiskFromSnapshot
	vela leggera: CreateDiskSnapshot
	vela leggera: CreateDistribution
	vela leggera: CreateDomain

Prefisso del servizio	Azioni
	vela leggera: CreateGUISessionAccessDetails
	vela leggera: CreateInstanceSnapshot
	vela leggera: CreateInstances
	vela leggera: CreateInstancesFromSnapshot
	vela leggera: CreateKeyPair
	vela leggera: CreateLoadBalancer
	vela leggera: CreateLoadBalancerTlsCertificate
	vela leggera: CreateRelationalDatabase
	vela leggera: CreateRelationalDatabaseFromSnapshot
	vela leggera: CreateRelationalDatabaseSnapshot
	vela leggera: DeleteAlarm
	vela leggera: DeleteAutoSnapshot
	vela leggera: DeleteBucket
	vela leggera: DeleteBucketAccessKey
	vela leggera: DeleteCertificate
	vela leggera: DeleteContactMethod
	vela leggera: DeleteContainerImage
	vela leggera: DeleteContainerService
	vela leggera: DeleteDisk
	vela leggera: DeleteDiskSnapshot
	vela leggera: DeleteDistribution

Prefisso del servizio	Azioni
	vela leggera: DeleteDomain
	vela leggera: DeleteDomainEntry
	vela leggera: DeleteInstance
	vela leggera: DeleteInstanceSnapshot
	vela leggera: DeleteKeyPair
	vela leggera: DeleteKnownHostKeys
	vela leggera: DeleteLoadBalancer
	vela leggera: DeleteLoadBalancerTlsCertificate
	vela leggera: DeleteRelationalDatabase
	vela leggera: DeleteRelationalDatabaseSnapshot
	vela leggera: DetachCertificateFromDistribution
	vela leggera: DetachDisk
	vela leggera: DetachInstancesFromLoadBalancer
	vela leggera: DetachStaticIp
	vela leggera: DisableAddOn
	vela leggera: DownloadDefaultKeyPair
	vela leggera: EnableAddOn
	vela leggera: ExportSnapshot
	vela leggera: GetActiveNames
	vela leggera: GetAlarms
	vela leggera: GetAutoSnapshots

Prefisso del servizio	Azioni
	vela leggera: GetBlueprints
	vela leggera: GetBucketAccessKeys
	vela leggera: GetBucketBundles
	vela leggera: GetBucketMetricData
	vela leggera: GetBuckets
	vela leggera: GetBundles
	vela leggera: GetCertificates
	vela leggera: GetCloudFormationStackRecords
	vela leggera: GetContactMethods
	vela leggera: GetContainerAPIMetadata
	vela leggera: GetContainerImages
	vela leggera: GetContainerLog
	vela leggera: GetContainerServiceDeployments
	vela leggera: GetContainerServiceMetricData
	vela leggera: GetContainerServicePowers
	vela leggera: GetContainerServices
	vela leggera: GetCostEstimate
	vela leggera: GetDisk
	vela leggera: GetDiskSnapshot
	vela leggera: GetDiskSnapshots
	vela leggera: GetDisks

Prefisso del servizio	Azioni
	vela leggera: GetDistributionBundles
	vela leggera: GetDistributionLatestCacheReset
	vela leggera: GetDistributionMetricData
	vela leggera: GetDistributions
	vela leggera: GetDomain
	vela leggera: GetExportSnapshotRecords
	vela leggera: GetInstance
	vela leggera: GetInstanceMetricData
	vela leggera: GetInstancePortStates
	vela leggera: GetInstanceSnapshot
	vela leggera: GetInstanceSnapshots
	vela leggera: GetInstanceState
	vela leggera: GetInstances
	vela leggera: GetKeyPair
	vela leggera: GetKeyPairs
	vela leggera: GetLoadBalancer
	vela leggera: GetLoadBalancerMetricData
	vela leggera: GetLoadBalancerTlsCertificates
	vela leggera: GetLoadBalancerTlsPolicies
	vela leggera: GetLoadBalancers
	vela leggera: GetOperation

Prefisso del servizio	Azioni
	vela leggera: GetOperations
	vela leggera: GetOperationsForResource
	vela leggera: GetRegions
	vela leggera: GetRelationalDatabase
	vela leggera: GetRelationalDatabaseBlueprints
	vela leggera: GetRelationalDatabaseBundles
	vela leggera: GetRelationalDatabaseEvents
	vela leggera: GetRelationalDatabaseLogEvents
	vela leggera: GetRelationalDatabaseLogStreams
	vela leggera: GetRelationalDatabaseMasterUserPassword
	vela leggera: GetRelationalDatabaseMetricData
	vela leggera: GetRelationalDatabaseParameters
	vela leggera: GetRelationalDatabaseSnapshot
	vela leggera: GetRelationalDatabaseSnapshots
	vela leggera: GetRelationalDatabases
	vela leggera: GetSetupHistory
	vela leggera: GetStaticIp
	vela leggera: GetStaticIps
	vela leggera: ImportKeyPair
	vela leggera: IsVpcPeered
	vela leggera: OpenInstancePublicPorts

Prefisso del servizio	Azioni
	vela leggera: PeerVpc
	vela leggera: PutAlarm
	vela leggera: PutInstancePublicPorts
	vela leggera: RebootInstance
	vela leggera: RebootRelationalDatabase
	vela leggera: RegisterContainerImage
	vela leggera: ReleaseStaticIp
	vela leggera: ResetDistributionCache
	vela leggera: SendContactMethodVerification
	vela leggera: SetIpAddressType
	vela leggera: SetResourceAccessForBucket
	vela leggera: SetupInstanceHttps
	lightsail:StartGUISession
	vela leggera: StartInstance
	vela leggera: StartRelationalDatabase
	lightsail:StopGUISession
	vela leggera: StopInstance
	vela leggera: StopRelationalDatabase
	vela leggera: TestAlarm
	vela leggera: UnpeerVpc
	vela leggera: UpdateBucket

Prefisso del servizio	Azioni
	vela leggera: UpdateBucketBundle
	vela leggera: UpdateContainerService
	vela leggera: UpdateDistribution
	vela leggera: UpdateDistributionBundle
	vela leggera: UpdateDomainEntry
	vela leggera: UpdateInstanceMetadataOptions
	vela leggera: UpdateLoadBalancerAttribute
	vela leggera: UpdateRelationalDatabase
	vela leggera: UpdateRelationalDatabaseParameters

Prefisso del servizio	Azioni
log	registri: AssociateKmsKey registri: AssociateSourceToS3TableIntegration registri: CancelExportTask registri: CancelImportTask registri: CreateDelivery registri: CreateExportTask registri: CreateLogAnomalyDetector registri: CreateLogGroup registri: CreateLogStream registri: CreateLookupTable registri: DeleteDataProtectionPolicy registri: DeleteDelivery registri: DeleteDeliveryDestination registri: DeleteDeliveryDestinationPolicy registri: DeleteDeliverySource registri: DeleteDestination registri: DeleteIndexPolicy registri: DeleteIntegration registri: DeleteLogAnomalyDetector registri: DeleteLogGroup registri: DeleteLogStream

Prefisso del servizio	Azioni
	registri: DeleteLookupTable
	registri: DeleteMetricFilter
	registri: DeleteQueryDefinition
	registri: DeleteResourcePolicy
	registri: DeleteRetentionPolicy
	registri: DeleteScheduledQuery
	registri: DeleteSubscriptionFilter
	registri: DeleteSyslogConfiguration
	registri: DeleteTransformer
	registri: DescribeAccountPolicies
	registri: DescribeConfigurationTemplates
	registri: DescribeDeliveries
	registri: DescribeDeliveryDestinations
	registri: DescribeDeliverySources
	registri: DescribeDestinations
	registri: DescribeExportTasks
	registri: DescribeFieldIndexes
	registri: DescribeImportTaskBatches
	registri: DescribeImportTasks
	registri: DescribeIndexPolicies
	registri: DescribeLogGroups

Prefisso del servizio	Azioni
	registri: DescribeLogStreams registri: DescribeLookupTables registri: DescribeMetricFilters registri: DescribeQueries registri: DescribeQueryDefinitions registri: DescribeResourcePolicies registri: DescribeSubscriptionFilters registri: DisassociateKmsKey registri: DisassociateSourceFromS3TableIntegration registri: GetDataProtectionPolicy registri: GetDelivery registri: GetDeliveryDestination registri: GetDeliveryDestinationPolicy registri: GetDeliverySource registri: GetIntegration registri: GetLogAnomalyDetector registri: GetLogFields registri: GetLogGroupFields registri: GetLogRecord registri: GetLookupTable registri: GetQueryResults

Prefisso del servizio	Azioni
	registri: GetScheduledQuery registri: GetScheduledQueryHistory registri: GetStorageTierPolicy registri: GetTransformer registri: ListAnomalies registri: ListIntegrations registri: ListLogAnomalyDetectors registri: ListLogGroupsForQuery registri: ListScheduledQueries registri: ListSourcesForS3TableIntegration registri: ListSyslogConfigurations registri: PutBearerTokenAuthentication registri: PutDataProtectionPolicy registri: PutDeliveryDestination registri: PutDeliveryDestinationPolicy registri: PutDeliverySource registri: PutDestination registri: PutDestinationPolicy registri: PutIndexPolicy registri: PutIntegration registri: PutLogGroupDeletionProtection

Prefisso del servizio	Azioni
	registri: PutMetricFilter registri: PutQueryDefinition registri: PutResourcePolicy registri: PutRetentionPolicy registri: PutStorageTierPolicy registri: PutSubscriptionFilter registri: PutSyslogConfiguration registri: PutTransformer registri: StartLiveTail registri: StartQuery registri: StopQuery registri: TestMetricFilter registri: TestTransformer registri: UpdateAnomaly registri: UpdateDeliveryConfiguration registri: UpdateLogAnomalyDetector registri: UpdateLookupTable

Prefisso del servizio	Azioni
lookoutequipment	attrezzatura di avvistamento: CreateDataset
	attrezzatura di avvistamento: CreateInferenceScheduler
	attrezzatura di avvistamento: CreateLabel
	attrezzatura di avvistamento: CreateLabelGroup
	attrezzatura di avvistamento: CreateModel
	attrezzatura di avvistamento: DeleteDataset
	attrezzatura di avvistamento: DeleteInferenceScheduler
	attrezzatura di avvistamento: DeleteLabel
	attrezzatura di avvistamento: DeleteLabelGroup
	attrezzatura di avvistamento: DeleteModel
	attrezzatura di avvistamento: DeleteResourcePolicy
	attrezzatura di avvistamento: DeleteRetrainingScheduler
	attrezzatura di avvistamento: DescribeDataIngestionJob
	attrezzatura di avvistamento: DescribeDataset
	attrezzatura di avvistamento: DescribeInferenceScheduler
	attrezzatura di avvistamento: DescribeLabelGroup
	attrezzatura di avvistamento: DescribeModel
	attrezzatura di avvistamento: DescribeModelVersion
	attrezzatura di avvistamento: DescribeResourcePolicy
	attrezzatura di avvistamento: DescribeRetrainingScheduler
	lookoutequipment:Describelabel

Prefisso del servizio	Azioni
	attrezzatura di avvistamento: ImportDataset
	attrezzatura di avvistamento: ImportModelVersion
	attrezzatura di avvistamento: ListDataIngestionJobs
	attrezzatura di avvistamento: ListDatasets
	attrezzatura di avvistamento: ListInferenceEvents
	attrezzatura di avvistamento: ListInferenceExecutions
	attrezzatura di avvistamento: ListInferenceSchedulers
	attrezzatura di avvistamento: ListLabelGroups
	attrezzatura di avvistamento: ListLabels
	attrezzatura di avvistamento: ListModelVersions
	attrezzatura di avvistamento: ListModels
	attrezzatura di avvistamento: ListRetrainingSchedulers
	attrezzatura di avvistamento: ListSensorStatistics
	attrezzatura di avvistamento: PutResourcePolicy
	attrezzatura di avvistamento: StartDataIngestionJob
	attrezzatura di avvistamento: StartInferenceScheduler
	attrezzatura di avvistamento: StartRetrainingScheduler
	attrezzatura di avvistamento: StopInferenceScheduler
	attrezzatura di avvistamento: StopRetrainingScheduler
	attrezzatura di avvistamento: UpdateActiveModelVersion
	attrezzatura di avvistamento: UpdateInferenceScheduler

Prefisso del servizio	Azioni
	attrezzatura di avvistamento: UpdateLabelGroup attrezzatura di avvistamento: UpdateModel attrezzatura di avvistamento: UpdateRetrainingScheduler

Prefisso del servizio	Azioni
lookoutmetrics	metriche di osservazione: ActivateAnomalyDetector metriche di attenzione: BackTestAnomalyDetector metriche di attenzione: CreateAlert metriche di attenzione: CreateAnomalyDetector metriche di attenzione: CreateMetricSet metriche di attenzione: DeactivateAnomalyDetector metriche di attenzione: DeleteAlert metriche di attenzione: DeleteAnomalyDetector metriche di attenzione: DescribeAlert metriche di attenzione: DescribeAnomalyDetectionExecutions metriche di attenzione: DescribeAnomalyDetector metriche di attenzione: DescribeMetricSet metriche di attenzione: DetectMetricSetConfig metriche di attenzione: GetAnomalyGroup metriche di attenzione: GetDataQualityMetrics metriche di attenzione: GetFeedback metriche di attenzione: GetSampleData metriche di attenzione: ListAlerts metriche di attenzione: ListAnomalyDetectors metriche di attenzione: ListAnomalyGroupRelatedMetrics metriche di attenzione: ListAnomalyGroupSummaries

Prefisso del servizio	Azioni
	metriche di attenzione: ListAnomalyGroupTimeSeries metriche di attenzione: ListMetricSets metriche di attenzione: PutFeedback metriche di attenzione: UpdateAlert metriche di attenzione: UpdateAnomalyDetector metriche di attenzione: UpdateMetricSet

Prefisso del servizio	Azioni
m2	m2: CancelBatchJobExecution
	m2: CreateApplication
	m2: CreateDataSetExportTask
	m2: CreateDataSetImportTask
	m2: CreateDeployment
	m2: CreateEnvironment
	m2: DeleteApplication
	m2: DeleteApplicationFromEnvironment
	m2: DeleteEnvironment
	m2: GetApplication
	m2: GetApplicationVersion
	m2: GetBatchJobExecution
	m2: GetDataSetDetails
	m2: GetDataSetExportTask
	m2: GetDataSetImportTask
	m2: GetDeployment
	m2: GetEnvironment
	m2: GetSignedBluinsightsUrl
	m2: ListApplicationVersions
	m2: ListApplications
	m2: ListBatchJobDefinitions

Prefisso del servizio	Azioni
	m2: ListBatchJobExecutions
	m2: ListBatchJobRestartPoints
	m2: ListDataSetExportHistory
	m2: ListDataSetImportHistory
	m2: ListDataSets
	m2: ListDeployments
	m2: ListEngineVersions
	m2: ListEnvironments
	m2: StartApplication
	m2: StartBatchJob
	m2: StopApplication
	m2: UpdateApplication
	m2: UpdateEnvironment

Prefisso del servizio	Azioni
managedblockchain	blockchain gestita: CreateAccessor
	blockchain gestita: CreateMember
	blockchain gestita: CreateNetwork
	blockchain gestita: CreateNode
	blockchain gestita: CreateProposal
	blockchain gestita: DeleteAccessor
	blockchain gestita: DeleteMember
	blockchain gestita: DeleteNode
	blockchain gestita: GetAccessor
	blockchain gestita: GetMember
	blockchain gestita: GetNetwork
	blockchain gestita: GetNode
	blockchain gestita: GetProposal
	blockchain gestita: InvokeRpcPolygonMainnet
	blockchain gestita: InvokeRpcPolygonMumbaiTestnet
	blockchain gestita: ListAccessors
	blockchain gestita: ListInvitations
	blockchain gestita: ListMembers
	blockchain gestita: ListNetworks
	blockchain gestita: ListNodes
	blockchain gestita: ListProposalVotes

Prefisso del servizio	Azioni
	blockchain gestita: ListProposals blockchain gestita: RejectInvitation blockchain gestita: UpdateMember blockchain gestita: UpdateNode blockchain gestita: VoteOnProposal

Prefisso del servizio	Azioni
mediaconnect	connessione multimediale: AddBridgeOutputs
	connessione multimediale: AddBridgeSources
	connessione multimediale: AddFlowMediaStreams
	connessione multimediale: AddFlowOutputs
	connessione multimediale: AddFlowSources
	connessione multimediale: AddFlowVpcInterfaces
	connessione multimediale: CreateBridge
	connessione multimediale: CreateFlow
	connessione multimediale: CreateGateway
	connessione multimediale: DeleteBridge
	connessione multimediale: DeleteFlow
	connessione multimediale: DeleteGateway
	connessione multimediale: DeleteRouterInput
	connessione multimediale: DeleteRouterNetworkInterface
	connessione multimediale: DeleteRouterOutput
	connessione multimediale: DeregisterGatewayInstance
	connessione multimediale: DescribeBridge
	connessione multimediale: DescribeFlow
	connessione multimediale: DescribeFlowSourceMetadata
	connessione multimediale: DescribeFlowSourceThumbnail
	connessione multimediale: DescribeGateway

Prefisso del servizio	Azioni
	connessione multimediale: DescribeGatewayInstance
	connessione multimediale: DescribeOffering
	connessione multimediale: DescribeReservation
	connessione multimediale: GetRouterInput
	connessione multimediale: GetRouterInputSourceMetadata
	connessione multimediale: GetRouterInputThumbnail
	connessione multimediale: GetRouterNetworkInterface
	connessione multimediale: GetRouterOutput
	connessione multimediale: GrantFlowEntitlements
	connessione multimediale: ListBridges
	connessione multimediale: ListEntitlements
	connessione multimediale: ListFlows
	connessione multimediale: ListGatewayInstances
	connessione multimediale: ListGateways
	connessione multimediale: ListOfferings
	connessione multimediale: ListReservations
	connessione multimediale: ListRouterInputs
	connessione multimediale: ListRouterNetworkInterfaces
	connessione multimediale: ListRouterOutputs
	connessione multimediale: PurchaseOffering
	connessione multimediale: RemoveBridgeOutput

Prefisso del servizio	Azioni
	connessione multimediale: RemoveBridgeSource
	connessione multimediale: RemoveFlowMediaStream
	connessione multimediale: RemoveFlowOutput
	connessione multimediale: RemoveFlowSource
	connessione multimediale: RemoveFlowVpcInterface
	connessione multimediale: RestartRouterInput
	connessione multimediale: RestartRouterOutput
	connessione multimediale: RevokeFlowEntitlement
	connessione multimediale: StartFlow
	connessione multimediale: StartRouterInput
	connessione multimediale: StartRouterOutput
	connessione multimediale: StopFlow
	connessione multimediale: StopRouterInput
	connessione multimediale: StopRouterOutput
	connessione multimediale: TakeRouterInput
	connessione multimediale: UpdateBridge
	connessione multimediale: UpdateBridgeOutput
	connessione multimediale: UpdateBridgeSource
	connessione multimediale: UpdateBridgeState
	connessione multimediale: UpdateFlow
	connessione multimediale: UpdateFlowEntitlement

Prefisso del servizio	Azioni
	connessione multimediale: UpdateFlowMediaStream connessione multimediale: UpdateGatewayInstance

Prefisso del servizio	Azioni
mediaconvert	conversione multimediale: AssociateCertificate conversione multimediale: CancelJob conversione multimediale: CreateJob conversione multimediale: CreateJobTemplate conversione multimediale: CreatePreset conversione multimediale: CreateQueue conversione multimediale: CreateResourceShare conversione multimediale: DeleteJobTemplate conversione multimediale: DeletePolicy conversione multimediale: DeletePreset conversione multimediale: DeleteQueue conversione multimediale: DescribeEndpoints conversione multimediale: DisassociateCertificate conversione multimediale: GetJob conversione multimediale: GetJobTemplate conversione multimediale: GetPolicy conversione multimediale: GetPreset conversione multimediale: GetQueue conversione multimediale: ListJobTemplates conversione multimediale: ListJobs conversione multimediale: ListPresets

Prefisso del servizio	Azioni
	conversione multimediale: ListQueues conversione multimediale: ListVersions mediaconvert:Probe conversione multimediale: PutPolicy conversione multimediale: SearchJobs conversione di file multimediali: UpdateJobTemplate conversione di file multimediali: UpdatePreset conversione di file multimediali: UpdateQueue

Prefisso del servizio	Azioni
medialive	media live: AcceptInputDeviceTransfer
	media in diretta: BatchDelete
	media in diretta: BatchStart
	media in diretta: BatchStop
	media in diretta: BatchUpdateSchedule
	media in diretta: CancelInputDeviceTransfer
	media in diretta: ClaimDevice
	media in diretta: CreateChannel
	media in diretta: CreateChannelPlacementGroup
	media in diretta: CreateCloudWatchAlarmTemplate
	media in diretta: CreateCloudWatchAlarmTemplateGroup
	media in diretta: CreateCluster
	media in diretta: CreateEventBridgeRuleTemplate
	media in diretta: CreateEventBridgeRuleTemplateGroup
	media in diretta: CreateInput
	media in diretta: CreateInputSecurityGroup
	media in diretta: CreateMultiplex
	media in diretta: CreateMultiplexProgram
	media in diretta: CreateNetwork
	media in diretta: CreateNode
	media in diretta: CreateNodeRegistrationScript

Prefisso del servizio	Azioni
	media in diretta: CreatePartnerInput
	media in diretta: CreateSdiSource
	media in diretta: CreateSignalMap
	media in diretta: DeleteChannel
	media in diretta: DeleteChannelPlacementGroup
	media in diretta: DeleteCloudWatchAlarmTemplate
	media in diretta: DeleteCloudWatchAlarmTemplateGroup
	media in diretta: DeleteCluster
	media in diretta: DeleteEventBridgeRuleTemplate
	media in diretta: DeleteEventBridgeRuleTemplateGroup
	media in diretta: DeleteInput
	media in diretta: DeleteInputSecurityGroup
	media in diretta: DeleteMultiplex
	media in diretta: DeleteMultiplexProgram
	media in diretta: DeleteNetwork
	media in diretta: DeleteNode
	media in diretta: DeleteReservation
	media in diretta: DeleteSchedule
	media in diretta: DeleteSdiSource
	media in diretta: DeleteSignalMap
	media in diretta: DescribeAccountConfiguration

Prefisso del servizio	Azioni
	media in diretta: DescribeChannel
	media in diretta: DescribeChannelPlacementGroup
	media in diretta: DescribeCluster
	media in diretta: DescribeInput
	media in diretta: DescribeInputDevice
	media in diretta: DescribeInputDeviceThumbnail
	media in diretta: DescribeInputSecurityGroup
	media in diretta: DescribeMultiplex
	media in diretta: DescribeMultiplexProgram
	media in diretta: DescribeNetwork
	media in diretta: DescribeNode
	media in diretta: DescribeOffering
	media in diretta: DescribeReservation
	media in diretta: DescribeSchedule
	media in diretta: DescribeSdiSource
	media in diretta: DescribeThumbnails
	media in diretta: GetCloudWatchAlarmTemplate
	media in diretta: GetCloudWatchAlarmTemplateGroup
	media in diretta: GetEventBridgeRuleTemplate
	media in diretta: GetEventBridgeRuleTemplateGroup
	media in diretta: GetSignalMap

Prefisso del servizio	Azioni
	media in diretta: ListAlerts
	media in diretta: ListChannelPlacementGroups
	media in diretta: ListChannels
	media in diretta: ListCloudWatchAlarmTemplateGroups
	media in diretta: ListCloudWatchAlarmTemplates
	media in diretta: ListClusterAlerts
	media in diretta: ListClusters
	media in diretta: ListEventBridgeRuleTemplateGroups
	media in diretta: ListEventBridgeRuleTemplates
	media in diretta: ListInputDeviceTransfers
	media in diretta: ListInputDevices
	media in diretta: ListInputSecurityGroups
	media in diretta: ListInputs
	media in diretta: ListMultiplexAlerts
	media in diretta: ListMultiplexPrograms
	media in diretta: ListMultiplexes
	media in diretta: ListNetworks
	media in diretta: ListNodes
	media in diretta: ListOfferings
	media in diretta: ListReservations
	media in diretta: ListSdiSources

Prefisso del servizio	Azioni
	media in diretta: ListSignalMaps
	media in diretta: ListVersions
	media in diretta: PurchaseOffering
	media in diretta: RebootInputDevice
	media in diretta: RejectInputDeviceTransfer
	media in diretta: RestartChannelPipelines
	media in diretta: StartChannel
	media in diretta: StartDeleteMonitorDeployment
	media in diretta: StartInputDevice
	media in diretta: StartInputDeviceMaintenanceWindow
	media in diretta: StartMonitorDeployment
	media in diretta: StartMultiplex
	media in diretta: StartUpdateSignalMap
	media in diretta: StopChannel
	media in diretta: StopInputDevice
	media in diretta: StopMultiplex
	media in diretta: TransferInputDevice
	media in diretta: UpdateAccountConfiguration
	media in diretta: UpdateChannel
	media in diretta: UpdateChannelClass
	media in diretta: UpdateChannelPlacementGroup

Prefisso del servizio	Azioni
	media in diretta: UpdateCloudWatchAlarmTemplate
	media in diretta: UpdateCloudWatchAlarmTemplateGroup
	media in diretta: UpdateCluster
	media in diretta: UpdateEventBridgeRuleTemplate
	media in diretta: UpdateEventBridgeRuleTemplateGroup
	media in diretta: UpdateInput
	media in diretta: UpdateInputDevice
	media in diretta: UpdateInputSecurityGroup
	media in diretta: UpdateMultiplex
	media in diretta: UpdateMultiplexProgram
	media in diretta: UpdateNetwork
	media in diretta: UpdateNode
	media in diretta: UpdateNodeState
	media in diretta: UpdateReservation
	media in diretta: UpdateSdiSource

Prefisso del servizio	Azioni
mediatailor	media tailor: ConfigureLogsForPlaybackConfiguration mediatailor: CreateChannel mediatailor: CreateLiveSource mediatailor: CreatePrefetchSchedule mediatailor: CreateProgram mediatailor: CreateSourceLocation mediatailor: CreateVodSource mediatailor: DeleteChannel mediatailor: DeleteChannelPolicy mediatailor: DeleteLiveSource mediatailor: DeletePlaybackConfiguration mediatailor: DeletePrefetchSchedule mediatailor: DeleteProgram mediatailor: DeleteSourceLocation mediatailor: DeleteVodSource mediatailor: DescribeChannel mediatailor: DescribeLiveSource mediatailor: DescribeProgram mediatailor: DescribeSourceLocation mediatailor: DescribeVodSource mediatailor: GetChannelPolicy

Prefisso del servizio	Azioni
	mediatailor: GetChannelSchedule mediatailor: GetPlaybackConfiguration mediatailor: GetPrefetchSchedule mediatailor: ListAlerts mediatailor: ListChannels mediatailor: ListLiveSources mediatailor: ListPlaybackConfigurations mediatailor: ListPrefetchSchedules mediatailor: ListSourceLocations mediatailor: ListVodSources mediatailor: PutChannelPolicy mediatailor: PutPlaybackConfiguration mediatailor: StartChannel mediatailor: StopChannel mediatailor: UpdateChannel mediatailor: UpdateLiveSource mediatailor: UpdateProgram mediatailor: UpdateSourceLocation mediatailor: UpdateVodSource

Prefisso del servizio	Azioni
db di memoria	db di memoria: BatchUpdateCluster
	db di memoria: CopySnapshot
	db di memoria: CreateAcl
	db di memoria: CreateCluster
	db di memoria: CreateMultiRegionCluster
	db di memoria: CreateParameterGroup
	db di memoria: CreateSnapshot
	db di memoria: CreateSubnetGroup
	db di memoria: CreateUser
	db di memoria: DeleteAcl
	db di memoria: DeleteCluster
	db di memoria: DeleteMultiRegionCluster
	db di memoria: DeleteParameterGroup
	db di memoria: DeleteSnapshot
	db di memoria: DeleteSubnetGroup
	db di memoria: DeleteUser
	db di memoria: DescribeAcls
	db di memoria: DescribeClusters
	db di memoria: DescribeEngineVersions
	db di memoria: DescribeEvents
	db di memoria: DescribeMultiRegionClusters

Prefisso del servizio	Azioni
	db di memoria: DescribeMultiRegionParameterGroups
	db di memoria: DescribeMultiRegionParameters
	db di memoria: DescribeParameterGroups
	db di memoria: DescribeParameters
	db di memoria: DescribeReservedNodes
	db di memoria: DescribeReservedNodesOfferings
	db di memoria: DescribeServiceUpdates
	db di memoria: DescribeSnapshots
	db di memoria: DescribeSubnetGroups
	db di memoria: DescribeUsers
	db di memoria: FailoverShard
	db di memoria: ListAllowedMultiRegionClusterUpdates
	db di memoria: ListAllowedNodeTypeUpdates
	db di memoria: PurchaseReservedNodesOffering
	db di memoria: ResetParameterGroup
	db di memoria: UpdateAcl
	db di memoria: UpdateCluster
	db di memoria: UpdateMultiRegionCluster
	db di memoria: UpdateParameterGroup
	db di memoria: UpdateSubnetGroup
	db di memoria: UpdateUser

Prefisso del servizio	Azioni
mgh	mg: AssociateCreatedArtifact mg: AssociateDiscoveredResource mg: AssociateSourceResource mg: CreateHomeRegionControl mg: CreateProgressUpdateStream mg: DeleteHomeRegionControl mg: DeleteProgressUpdateStream mg: DescribeApplicationState mg: DescribeHomeRegionControls mg: DescribeMigrationTask mg: DisassociateCreatedArtifact mg: DisassociateDiscoveredResource mg: DisassociateSourceResource mg: GetHomeRegion mg: ImportMigrationTask mg: ListApplicationStates mg: ListCreatedArtifacts mg: ListDiscoveredResources mg: ListMigrationTaskUpdates mg: ListMigrationTasks mg: ListProgressUpdateStreams

Prefisso del servizio	Azioni
	mg: ListSourceResources mg: NotifyApplicationState mg: NotifyMigrationTaskState mg: PutResourceAttributes

Prefisso del servizio	Azioni
mgn	mgn: ArchiveApplication
	mgn: ArchiveWave
	mgn: AssociateApplications
	mgn: AssociateSourceServers
	mgn: ChangeServerLifeCycleState
	mgn: CreateApplication
	mgn: CreateConnector
	mgn: CreateLaunchConfigurationTemplate
	mgn: CreateNetworkMigrationDefinition
	mgn: CreateReplicationConfigurationTemplate
	mgn: CreateWave
	mgn: DeleteApplication
	mgn: DeleteConnector
	mgn: DeleteJob
	mgn: DeleteLaunchConfigurationTemplate
	mgn: DeleteNetworkMigrationDefinition
	mgn: DeleteReplicationConfigurationTemplate
	mgn: DeleteSourceServer
	mgn: DeleteVcenterClient
	mgn: DeleteWave
	mgn: DescribeJobLogItems

Prefisso del servizio	Azioni
	mgn: DescribeJobs
	mgn: DescribeLaunchConfigurationTemplates
	mgn: DescribeReplicationConfigurationTemplates
	mgn: DescribeVcenterClients
	mgn: DisassociateApplications
	mgn: DisassociateSourceServers
	mgn: DisconnectFromService
	mgn: FinalizeCutover
	mgn: GetNetworkMigrationDefinition
	mgn: GetNetworkMigrationMapperSegmentConstruct
	mgn: GetReplicationConfiguration
	mgn: InitializeService
	mgn: ListConnectors
	mgn: ListExportErrors
	mgn: ListExports
	mgn: ListImportErrors
	mgn: ListImportFileEnrichments
	mgn: ListImports
	mgn: ListManagedAccounts
	mgn: ListNetworkMigrationAnalyses
	mgn: ListNetworkMigrationAnalysisResults

Prefisso del servizio	Azioni
	mgn: ListNetworkMigrationCodeGenerationSegments
	mgn: ListNetworkMigrationCodeGenerations
	mgn: ListNetworkMigrationDefinitions
	mgn: ListNetworkMigrationDeployedStacks
	mgn: ListNetworkMigrationDeployments
	mgn: ListNetworkMigrationExecutions
	mgn: ListNetworkMigrationMapperSegmentConstructs
	mgn: ListNetworkMigrationMapperSegments
	mgn: ListNetworkMigrationMappingUpdates
	mgn: ListNetworkMigrationMappings
	mgn: ListSourceServerActions
	mgn: ListTemplateActions
	mgn: MarkAsArchived
	mgn: PauseReplication
	mgn: PutSourceServerAction
	mgn: PutTemplateAction
	mgn: RemoveSourceServerAction
	mgn: RemoveTemplateAction
	mgn: ResumeReplication
	mgn: RetryDataReplication
	mgn: StartCutover

Prefisso del servizio	Azioni
	mgn: StartExport
	mgn: StartImport
	mgn: StartImportFileEnrichment
	mgn: StartNetworkMigrationCodeGeneration
	mgn: StartNetworkMigrationMapping
	mgn: StartNetworkMigrationMappingUpdate
	mgn: StartReplication
	mgn: StartTest
	mgn: StopReplication
	mgn: TerminateTargetInstances
	mgn: UnarchiveApplication
	mgn: UnarchiveWave
	mgn: UpdateApplication
	mgn: UpdateConnector
	mgn: UpdateLaunchConfigurationTemplate
	mgn: UpdateNetworkMigrationDefinition
	mgn: UpdateNetworkMigrationMapperSegment
	mgn: UpdateReplicationConfiguration
	mgn: UpdateReplicationConfigurationTemplate
	mgn: UpdateSourceServer
	mgn: UpdateSourceServerReplicationType

Prefisso del servizio	Azioni
	mgn: UpdateWave

Prefisso del servizio	Azioni
migrationhub-strategy	strategia migrationhub: GetAntiPattern migrationhub-strategy: GetApplicationComponentDetails migrationhub-strategy: GetApplicationComponentStrategies migrationhub-strategy: GetAssessment migrationhub-strategy: GetImportFileTask migrationhub-strategy: GetLatestAssessmentId migrationhub-strategy: GetMessage migrationhub-strategy: GetPortfolioPreferences migrationhub-strategy: GetPortfolioSummary migrationhub-strategy: GetRecommendationReportDetails migrationhub-strategy: GetServerDetails migrationhub-strategy: GetServerStrategies migrationhub-strategy: ListAnalyzableServers migrationhub-strategy: ListAntiPatterns migrationhub-strategy: ListApplicationComponents migrationhub-strategy: ListCollectors migrationhub-strategy: ListImportFileTask migrationhub-strategy: ListJarArtifacts migrationhub-strategy: ListServers migrationhub-strategy: PutLogData migrationhub-strategy: PutMetricData

Prefisso del servizio	Azioni
	migrationhub-strategy: PutPortfolioPreferences migrationhub-strategy: RegisterCollector migrationhub-strategy: SendMessage migrationhub-strategy: StartAssessment migrationhub-strategy: StartImportFileTask migrationhub-strategy: StartRecommendationReportGeneration migrationhub-strategy: StopAssessment migrationhub-strategy: UpdateApplicationComponentConfig migrationhub-strategy: UpdateCollectorConfiguration migrationhub-strategy: UpdateServerConfig

Prefisso del servizio	Azioni
mobiletargeting	targeting mobile: CreateApp
	targeting mobile: CreateCampaign
	targeting mobile: CreateEmailTemplate
	targeting mobile: CreateExportJob
	targeting mobile: CreateImportJob
	targeting mobile: CreateInAppTemplate
	targeting mobile: CreateJourney
	targeting mobile: CreatePushTemplate
	targeting mobile: CreateRecommenderConfiguration
	targeting mobile: CreateSegment
	targeting mobile: CreateSmsTemplate
	targeting mobile: CreateVoiceTemplate
	targeting mobile: DeleteAdmChannel
	targeting mobile: DeleteApnsChannel
	targeting mobile: DeleteApnsSandboxChannel
	targeting mobile: DeleteApnsVoipChannel
	targeting mobile: DeleteApnsVoipSandboxChannel
	targeting mobile: DeleteApp
	targeting mobile: DeleteBaiduChannel
	targeting mobile: DeleteCampaign
	targeting mobile: DeleteEmailChannel

Prefisso del servizio	Azioni
	targeting mobile: DeleteEmailTemplate
	targeting mobile: DeleteEndpoint
	targeting mobile: DeleteEventStream
	targeting mobile: DeleteGcmChannel
	targeting mobile: DeleteInAppTemplate
	targeting mobile: DeleteJourney
	targeting mobile: DeletePushTemplate
	targeting mobile: DeleteRecommenderConfiguration
	targeting mobile: DeleteSegment
	targeting mobile: DeleteSmsChannel
	targeting mobile: DeleteSmsTemplate
	targeting mobile: DeleteUserEndpoints
	targeting mobile: DeleteVoiceChannel
	targeting mobile: DeleteVoiceTemplate
	targeting mobile: GetAdmChannel
	targeting mobile: GetApnsChannel
	targeting mobile: GetApnsSandboxChannel
	targeting mobile: GetApnsVoipChannel
	targeting mobile: GetApnsVoipSandboxChannel
	targeting mobile: GetApp
	targeting mobile: GetApplicationDateRangeKpi

Prefisso del servizio	Azioni
	targeting mobile: GetApplicationSettings
	targeting mobile: GetApps
	targeting mobile: GetBaiduChannel
	targeting mobile: GetCampaign
	targeting mobile: GetCampaignActivities
	targeting mobile: GetCampaignDateRangeKpi
	targeting mobile: GetCampaignVersion
	targeting mobile: GetCampaignVersions
	targeting mobile: GetCampaigns
	targeting mobile: GetChannels
	targeting mobile: GetEmailChannel
	targeting mobile: GetEmailTemplate
	targeting mobile: GetEndpoint
	targeting mobile: GetEventStream
	targeting mobile: GetExportJob
	targeting mobile: GetExportJobs
	targeting mobile: GetGcmChannel
	targeting mobile: GetImportJob
	targeting mobile: GetImportJobs
	targeting mobile: GetInAppMessages
	targeting mobile: GetInAppTemplate

Prefisso del servizio	Azioni
	targeting mobile: GetJourney
	targeting mobile: GetJourneyDateRangeKpi
	targeting mobile: GetJourneyExecutionActivityMetrics
	targeting mobile: GetJourneyExecutionMetrics
	targeting mobile: GetJourneyRunExecutionActivityMetrics
	targeting mobile: GetJourneyRunExecutionMetrics
	targeting mobile: GetJourneyRuns
	targeting mobile: GetPushTemplate
	targeting mobile: GetRecommenderConfiguration
	targeting mobile: GetRecommenderConfigurations
	targeting mobile: GetSegment
	targeting mobile: GetSegmentExportJobs
	targeting mobile: GetSegmentImportJobs
	targeting mobile: GetSegmentVersion
	targeting mobile: GetSegmentVersions
	targeting mobile: GetSegments
	targeting mobile: GetSmsChannel
	targeting mobile: GetSmsTemplate
	targeting mobile: GetUserEndpoints
	targeting mobile: GetVoiceChannel
	targeting mobile: GetVoiceTemplate

Prefisso del servizio	Azioni
	targeting mobile: ListJourneys
	targeting mobile: ListTemplateVersions
	targeting mobile: ListTemplates
	targeting mobile: PhoneNumberValidate
	targeting mobile: PutEventStream
	targeting mobile: RemoveAttributes
	targeting mobile: UpdateAdmChannel
	targeting mobile: UpdateApnsChannel
	targeting mobile: UpdateApnsSandboxChannel
	targeting mobile: UpdateApnsVoipChannel
	targeting mobile: UpdateApnsVoipSandboxChannel
	targeting mobile: UpdateApplicationSettings
	targeting mobile: UpdateBaiduChannel
	targeting mobile: UpdateCampaign
	targeting mobile: UpdateEmailChannel
	targeting mobile: UpdateEmailTemplate
	targeting mobile: UpdateEndpoint
	targeting mobile: UpdateEndpointsBatch
	targeting mobile: UpdateGcmChannel
	targeting mobile: UpdateInAppTemplate
	targeting mobile: UpdateJourney

Prefisso del servizio	Azioni
	targeting mobile: UpdateJourneyState
	targeting mobile: UpdatePushTemplate
	targeting mobile: UpdateRecommenderConfiguration
	targeting mobile: UpdateSegment
	targeting mobile: UpdateSmsChannel
	targeting mobile: UpdateSmsTemplate
	targeting mobile: UpdateTemplateActiveVersion
	targeting mobile: UpdateVoiceChannel
	targeting mobile: UpdateVoiceTemplate
	mobiletargeting:VerifyOTPMessage

Prefisso del servizio	Azioni
mq	mq: CreateBroker mq: CreateConfiguration mq: CreateUser mq: DeleteBroker mq: DeleteConfiguration mq: DeleteUser mq: DescribeBroker mq: DescribeBrokerEngineTypes mq: DescribeBrokerInstanceOptions mq: DescribeConfiguration mq: DescribeConfigurationRevision mq: DescribeSharedResources mq: DescribeUser mq: ListBrokers mq: ListConfigurationRevisions mq: ListConfigurations mq: ListUsers mq:Promote mq: RebootBroker mq: UpdateBroker mq: UpdateConfiguration

Prefisso del servizio	Azioni
	mq: UpdateUser

Prefisso del servizio	Azioni
networkmanager	gestore di rete: AcceptAttachment
	gestore di rete: AssociateConnectPeer
	gestore di rete: AssociateCustomerGateway
	gestore di rete: AssociateLink
	gestore di rete: AssociateTransitGatewayConnectPeer
	gestore di rete: CreateConnectAttachment
	gestore di rete: CreateConnectPeer
	gestore di rete: CreateConnection
	gestore di rete: CreateCoreNetwork
	gestore di rete: CreateDevice
	gestore di rete: CreateDirectConnectGatewayAttachment
	gestore di rete: CreateGlobalNetwork
	gestore di rete: CreateLink
	gestore di rete: CreateSite
	gestore di rete: CreateSiteToSiteVpnAttachment
	gestore di rete: CreateTransitGatewayPeering
	gestore di rete: CreateTransitGatewayRouteTableAttachment
	gestore di rete: CreateVpcAttachment
	gestore di rete: DeleteAttachment
	gestore di rete: DeleteConnectPeer
	gestore di rete: DeleteConnection

Prefisso del servizio	Azioni
	gestore di rete: DeleteCoreNetwork
	gestore di rete: DeleteCoreNetworkPolicyVersion
	gestore di rete: DeleteDevice
	gestore di rete: DeleteGlobalNetwork
	gestore di rete: DeleteLink
	gestore di rete: DeletePeering
	gestore di rete: DeleteResourcePolicy
	gestore di rete: DeleteSite
	gestore di rete: DeregisterTransitGateway
	gestore di rete: DescribeGlobalNetworks
	gestore di rete: DisassociateConnectPeer
	gestore di rete: DisassociateCustomerGateway
	gestore di rete: DisassociateLink
	gestore di rete: DisassociateTransitGatewayConnectPeer
	gestore di rete: ExecuteCoreNetworkChangeSet
	gestore di rete: GetConnectAttachment
	gestore di rete: GetConnectPeer
	gestore di rete: GetConnectPeerAssociations
	gestore di rete: GetConnections
	gestore di rete: GetCoreNetwork
	gestore di rete: GetCoreNetworkChangeEvents

Prefisso del servizio	Azioni
	gestore di rete: GetCoreNetworkChangeSet
	gestore di rete: GetCoreNetworkPolicy
	gestore di rete: GetCustomerGatewayAssociations
	gestore di rete: GetDevices
	gestore di rete: GetLinkAssociations
	gestore di rete: GetLinks
	gestore di rete: GetNetworkResourceCounts
	gestore di rete: GetNetworkResourceRelationships
	gestore di rete: GetNetworkResources
	gestore di rete: GetNetworkRoutes
	gestore di rete: GetNetworkTelemetry
	gestore di rete: GetResourcePolicy
	gestore di rete: GetRouteAnalysis
	gestore di rete: GetSiteToSiteVpnAttachment
	gestore di rete: GetSites
	gestore di rete: GetTransitGatewayConnectPeerAssociations
	gestore di rete: GetTransitGatewayPeering
	gestore di rete: GetTransitGatewayRegistrations
	gestore di rete: GetTransitGatewayRouteTableAttachment
	gestore di rete: GetVpcAttachment
	gestore di rete: ListAttachmentRoutingPolicyAssociations

Prefisso del servizio	Azioni
	gestore di rete: ListAttachments
	gestore di rete: ListConnectPeers
	gestore di rete: ListCoreNetworkPolicyVersions
	gestore di rete: ListCoreNetworkPrefixListAssociations
	gestore di rete: ListCoreNetworkRoutingInformation
	gestore di rete: ListCoreNetworks
	gestore di rete: ListOrganizationServiceAccessStatus
	gestore di rete: ListPeerings
	gestore di rete: PutAttachmentRoutingPolicyLabel
	gestore di rete: PutCoreNetworkPolicy
	gestore di rete: PutResourcePolicy
	gestore di rete: RegisterTransitGateway
	gestore di rete: RejectAttachment
	gestore di rete: RemoveAttachmentRoutingPolicyLabel
	gestore di rete: RestoreCoreNetworkPolicyVersion
	gestore di rete: StartOrganizationServiceAccessUpdate
	gestore di rete: StartRouteAnalysis
	gestore di rete: UpdateConnection
	gestore di rete: UpdateCoreNetwork
	gestore di rete: UpdateDevice
	gestore di rete: UpdateDirectConnectGatewayAttachment

Prefisso del servizio	Azioni
	gestore di rete: UpdateGlobalNetwork gestore di rete: UpdateLink gestore di rete: UpdateNetworkResourceMetadata gestore di rete: UpdateSite gestore di rete: UpdateVpcAttachment

Prefisso del servizio	Azioni
nimble	agile: AcceptEulas agile: CreateLaunchProfile agile: CreateStreamingImage agile: CreateStreamingSession agile: CreateStreamingSessionStream agile: CreateStudio agile: CreateStudioComponent agile: DeleteLaunchProfile agile: DeleteLaunchProfileMember agile: DeleteStreamingImage agile: DeleteStreamingSession agile: DeleteStudio agile: DeleteStudioComponent agile: DeleteStudioMember agile: GetEula agile: GetLaunchProfileDetails agile: GetStreamingImage agile: GetStreamingSession agile: GetStreamingSessionBackup agile: GetStreamingSessionStream agile: GetStudio

Prefisso del servizio	Azioni
	agile: GetStudioComponent agile: GetStudioMember agile: ListEulas agile: ListLaunchProfileMembers agile: ListLaunchProfiles agile: ListStreamingImages agile: ListStreamingSessionBackups agile: ListStreamingSessions agile: ListStudioComponents agile: ListStudioMembers agile: ListStudios agile: PutLaunchProfileMembers agile: PutStudioMembers agile: StartStreamingSession agile: StartStudioSSOConfigurationRepair agile: StopStreamingSession agile: UpdateLaunchProfile agile: UpdateLaunchProfileMember agile: UpdateStreamingImage agile: UpdateStudio agile: UpdateStudioComponent

Prefisso del servizio	Azioni
omics	fumetti: AbortMultipartReadSetUpload
	fumetti: AcceptShare
	fumetti: BatchDeleteReadSet
	fumetti: CancelAnnotationImportJob
	fumetti: CancelRun
	fumetti: CancelRunBatch
	fumetti: CancelVariantImportJob
	fumetti: CompleteMultipartReadSetUpload
	fumetti: CreateAnnotationStore
	fumetti: CreateAnnotationStoreVersion
	fumetti: CreateConfiguration
	fumetti: CreateMultipartReadSetUpload
	fumetti: CreateReferenceStore
	fumetti: CreateRunGroup
	fumetti: CreateSequenceStore
	fumetti: CreateShare
	fumetti: CreateVariantStore
	fumetti: CreateWorkflow
	fumetti: CreateWorkflowVersion
	fumetti: DeleteAnnotationStore
	fumetti: DeleteAnnotationStoreVersions

Prefisso del servizio	Azioni
	fumetti: DeleteBatch
	fumetti: DeleteConfiguration
	fumetti: DeleteReference
	fumetti: DeleteReferenceStore
	fumetti: DeleteRun
	fumetti: DeleteRunBatch
	fumetti: DeleteRunGroup
	fumetti: DeleteSequenceStore
	fumetti: DeleteShare
	fumetti: DeleteVariantStore
	fumetti: DeleteWorkflow
	fumetti: DeleteWorkflowVersion
	fumetti: GetAnnotationImportJob
	fumetti: GetAnnotationStore
	fumetti: GetAnnotationStoreVersion
	fumetti: GetBatch
	fumetti: GetConfiguration
	fumetti: GetReadSet
	fumetti: GetReadSetActivationJob
	fumetti: GetReadSetExportJob
	fumetti: GetReadSetImportJob

Prefisso del servizio	Azioni
	fumetti: GetReadSetMetadata
	fumetti: GetReference
	fumetti: GetReferenceImportJob
	fumetti: GetReferenceMetadata
	fumetti: GetReferenceStore
	fumetti: GetRun
	fumetti: GetRunGroup
	fumetti: GetRunTask
	fumetti: GetSequenceStore
	fumetti: GetShare
	fumetti: GetVariantImportJob
	fumetti: GetVariantStore
	fumetti: GetWorkflow
	fumetti: GetWorkflowVersion
	fumetti: ListAnnotationImportJobs
	fumetti: ListAnnotationStoreVersions
	fumetti: ListAnnotationStores
	fumetti: ListBatch
	fumetti: ListConfigurations
	fumetti: ListMultipartReadSetUploads
	fumetti: ListReadSetActivationJobs

Prefisso del servizio	Azioni
	fumetti: ListReadSetExportJobs
	fumetti: ListReadSetImportJobs
	fumetti: ListReadSetUploadParts
	fumetti: ListReadSets
	fumetti: ListReferenceImportJobs
	fumetti: ListReferenceStores
	fumetti: ListReferences
	fumetti: ListRunGroups
	fumetti: ListRunTasks
	fumetti: ListRuns
	fumetti: ListRunsInBatch
	fumetti: ListSequenceStores
	fumetti: ListShares
	fumetti: ListVariantImportJobs
	fumetti: ListVariantStores
	fumetti: ListWorkflowVersions
	fumetti: ListWorkflows
	fumetti: StartAnnotationImportJob
	fumetti: StartReadSetActivationJob
	fumetti: StartReadSetExportJob
	fumetti: StartReadSetImportJob

Prefisso del servizio	Azioni
	fumetti: StartReferenceImportJob
	fumetti: StartRun
	fumetti: StartRunBatch
	fumetti: StartVariantImportJob
	fumetti: UpdateAnnotationStore
	fumetti: UpdateAnnotationStoreVersion
	fumetti: UpdateRunGroup
	fumetti: UpdateVariantStore
	fumetti: UpdateWorkflow
	fumetti: UpdateWorkflowVersion
	fumetti: UploadReadSetPart

Prefisso del servizio	Azioni
opsworks	Ops funziona: AssignInstance opsworks: AssignVolume opsworks: AssociateElasticIp opsworks: AttachElasticLoadBalancer opsworks: CloneStack opsworks: CreateApp opsworks: CreateDeployment opsworks: CreateInstance opsworks: CreateLayer opsworks: CreateStack opsworks: CreateUserProfile opsworks: DeleteApp opsworks: DeleteInstance opsworks: DeleteLayer opsworks: DeleteStack opsworks: DeleteUserProfile opsworks: DeregisterEcsCluster opsworks: DeregisterElasticIp opsworks: DeregisterInstance opsworks: DeregisterRdsDbInstance opsworks: DeregisterVolume

Prefisso del servizio	Azioni
	opsworks: DescribeAgentVersions opsworks: DescribeApps opsworks: DescribeCommands opsworks: DescribeDeployments opsworks: DescribeEcsClusters opsworks: DescribeElasticIps opsworks: DescribeElasticLoadBalancers opsworks: DescribeInstances opsworks: DescribeLayers opsworks: DescribeLoadBasedAutoScaling opsworks: DescribeMyUserProfile opsworks: DescribeOperatingSystems opsworks: DescribePermissions opsworks: DescribeRaidArrays opsworks: DescribeRdsDbInstances opsworks: DescribeServiceErrors opsworks: DescribeStackProvisioningParameters opsworks: DescribeStackSummary opsworks: DescribeStacks opsworks: DescribeTimeBasedAutoScaling opsworks: DescribeUserProfiles

Prefisso del servizio	Azioni
	opsworks: DescribeVolumes opsworks: DetachElasticLoadBalancer opsworks: DisassociateElasticIp opsworks: GetHostnameSuggestion opsworks: GrantAccess opsworks: RebootInstance opsworks: RegisterEcsCluster opsworks: RegisterElasticIp opsworks: RegisterInstance opsworks: RegisterRdsDbInstance opsworks: RegisterVolume opsworks: SetLoadBasedAutoScaling opsworks: SetPermission opsworks: SetTimeBasedAutoScaling opsworks: StartInstance opsworks: StartStack opsworks: StopInstance opsworks: StopStack opsworks: UnassignInstance opsworks: UnassignVolume opsworks: UpdateApp

Prefisso del servizio	Azioni
	opsworks: UpdateElasticIp opsworks: UpdateInstance opsworks: UpdateLayer opsworks: UpdateMyUserProfile opsworks: UpdateRdsDbInstance opsworks: UpdateStack opsworks: UpdateUserProfile opsworks: UpdateVolume

Prefisso del servizio	Azioni
opsworks-cm	opsworks-cm: AssociateNode
	opsworks-cm: CreateBackup
	opsworks-cm: CreateServer
	opsworks-cm: DeleteBackup
	opsworks-cm: DeleteServer
	opsworks-cm: DescribeAccountAttributes
	opsworks-cm: DescribeBackups
	opsworks-cm: DescribeEvents
	opsworks-cm: DescribeNodeAssociationStatus
	opsworks-cm: DescribeServers
	opsworks-cm: DisassociateNode
	opsworks-cm: ExportServerEngineAttribute
	opsworks-cm: RestoreServer
	opsworks-cm: StartMaintenance
	opsworks-cm: UpdateServer
	opsworks-cm: UpdateServerEngineAttributes

Prefisso del servizio	Azioni
organizations	organizzazioni: AcceptHandshake organizzazioni: AttachPolicy organizzazioni: CancelHandshake organizzazioni: CloseAccount organizzazioni: CreateAccount organizzazioni: CreateGovCloudAccount organizzazioni: CreateOrganization organizzazioni: CreateOrganizationalUnit organizzazioni: CreatePolicy organizzazioni: DeclineHandshake organizzazioni: DeleteOrganization organizzazioni: DeleteOrganizationalUnit organizzazioni: DeletePolicy organizzazioni: DeleteResourcePolicy organizzazioni: DeregisterDelegatedAdministrator organizzazioni: DescribeAccount organizzazioni: DescribeCreateAccountStatus organizzazioni: DescribeEffectivePolicy organizzazioni: DescribeHandshake organizzazioni: DescribeOrganization organizzazioni: DescribeOrganizationalUnit

Prefisso del servizio	Azioni
	organizzazioni: DescribePolicy organizzazioni: DescribeResourcePolicy organizzazioni: DescribeResponsibilityTransfer organizzazioni: DetachPolicy organizzazioni: DisableAWSServiceAccess organizzazioni: DisablePolicyType organizzazioni: EnableAWSServiceAccess organizzazioni: EnableAllFeatures organizzazioni: EnablePolicyType organizzazioni: InviteAccountToOrganization organizzazioni: LeaveOrganization organizzazioni: ListAWSServiceAccessForOrganization organizzazioni: ListAccounts organizzazioni: ListAccountsForParent organizzazioni: ListAccountsWithInvalidEffectivePolicy organizzazioni: ListChildren organizzazioni: ListCreateAccountStatus organizzazioni: ListDelegatedAdministrators organizzazioni: ListDelegatedServicesForAccount organizzazioni: ListHandshakesForAccount organizzazioni: ListHandshakesForOrganization

Prefisso del servizio	Azioni
	organizzazioni: ListInboundResponsibilityTransfers organizzazioni: ListOrganizationalUnitsForParent organizzazioni: ListOutboundResponsibilityTransfers organizzazioni: ListParents organizzazioni: ListPolicies organizzazioni: ListPoliciesForTarget organizzazioni: ListRoots organizzazioni: ListTargetsForPolicy organizzazioni: MoveAccount organizzazioni: PutResourcePolicy organizzazioni: RegisterDelegatedAdministrator organizzazioni: RemoveAccountFromOrganization organizzazioni: TerminateResponsibilityTransfer organizzazioni: UpdateOrganizationalUnit organizzazioni: UpdatePolicy organizzazioni: UpdateResponsibilityTransfer

Prefisso del servizio	Azioni
outposts	avamposti: CancelCapacityTask
	avamposti: CancelOrder
	avamposti: CreateOrder
	avamposti: CreateOutpost
	avamposti: CreatePrivateConnectivityConfig
	avamposti: CreateQuote
	avamposti: CreateRenewal
	avamposti: CreateSite
	avamposti: DeleteOutpost
	avamposti: DeleteQuote
	avamposti: DeleteSite
	avamposti: GetCapacityTask
	avamposti: GetCatalogItem
	avamposti: GetConnection
	avamposti: GetOrder
	avamposti: GetOutpost
	avamposti: GetOutpostBillingInformation
	avamposti: GetOutpostInstanceTypes
	avamposti: GetOutpostSupportedInstanceTypes
	avamposti: GetPrivateConnectivityConfig
	avamposti: GetQuote

Prefisso del servizio	Azioni
	avamposti: GetRenewalPricing
	avamposti: GetSite
	avamposti: GetSiteAddress
	avamposti: ListAssetInstances
	avamposti: ListAssets
	avamposti: ListBlockingInstancesForCapacityTask
	avamposti: ListCapacityTasks
	avamposti: ListCatalogItems
	avamposti: ListOrderableInstanceTypes
	avamposti: ListOrders
	avamposti: ListOutposts
	avamposti: ListQuotes
	avamposti: ListSites
	avamposti: StartCapacityTask
	avamposti: StartConnection
	avamposti: UpdateOutpost
	avamposti: UpdateQuote
	avamposti: UpdateSite
	avamposti: UpdateSiteAddress
	avamposti: UpdateSiteRackPhysicalProperties

Prefisso del servizio	Azioni
pi	pipì: CreatePerformanceAnalysisReport pipì: DeletePerformanceAnalysisReport pipì: DescribeDimensionKeys pipì: GetDimensionKeyDetails pipì: GetPerformanceAnalysisReport pipì: GetResourceMetadata pipì: GetResourceMetrics pipì: ListAvailableResourceDimensions pipì: ListAvailableResourceMetrics pipì: ListPerformanceAnalysisReports
pipes	tubi: CreatePipe tubi: DeletePipe tubi: DescribePipe tubi: ListPipes tubi: StartPipe tubi: StopPipe tubi: UpdatePipe

Prefisso del servizio	Azioni
polly	polly: DeleteLexicon polly: DescribeVoices polly: GetLexicon polly: GetSpeechSynthesisTask polly: ListLexicons polly: ListSpeechSynthesisTasks polly: PutLexicon polly: StartSpeechSynthesisTask polly: SynthesizeSpeech

Prefisso del servizio	Azioni
profile	profilo: AddProfileKey
	profilo: BatchGetCalculatedAttributeForProfile
	profilo: BatchGetProfile
	profilo: CreateCalculatedAttributeDefinition
	profilo: CreateDomain
	profilo: CreateEventStream
	profilo: CreateProfile
	profilo: CreateRecommender
	profilo: CreateRecommenderFilter
	profilo: CreateRecommenderSchema
	profilo: CreateSegmentDefinition
	profilo: CreateSegmentEstimate
	profilo: CreateSegmentSnapshot
	profilo: CreateUploadJob
	profilo: DeleteCalculatedAttributeDefinition
	profilo: DeleteDomain
	profilo: DeleteDomainObjectType
	profilo: DeleteEventStream
	profilo: DeleteIntegration
	profilo: DeleteProfile
	profilo: DeleteProfileKey

Prefisso del servizio	Azioni
	profilo: DeleteProfileObject
	profilo: DeleteProfileObjectType
	profilo: DeleteRecommender
	profilo: DeleteRecommenderFilter
	profilo: DeleteRecommenderSchema
	profilo: DeleteSegmentDefinition
	profilo: DeleteWorkflow
	profilo: DetectProfileObjectType
	profilo: GetAutoMergingPreview
	profilo: GetCalculatedAttributeDefinition
	profilo: GetCalculatedAttributeForProfile
	profilo: GetDomain
	profilo: GetDomainObjectType
	profilo: GetEventStream
	profilo: GetIdentityResolutionJob
	profilo: GetIntegration
	profilo: GetMatches
	profilo: GetObjectTypeAttributeStatistics
	profilo: GetProfileObjectType
	profilo: GetProfileObjectTypeTemplate
	profilo: GetProfileRecommendations

Prefisso del servizio	Azioni
	profilo: GetRecommender
	profilo: GetRecommenderFilter
	profilo: GetRecommenderSchema
	profilo: GetSegmentDefinition
	profilo: GetSegmentEstimate
	profilo: GetSegmentMembership
	profilo: GetSegmentSnapshot
	profilo: GetSimilarProfiles
	profilo: GetUploadJob
	profilo: GetUploadJobPath
	profilo: GetWorkflow
	profilo: GetWorkflowSteps
	profilo: ListAccountIntegrations
	profilo: ListCalculatedAttributeDefinitions
	profilo: ListCalculatedAttributesForProfile
	profilo: ListDomainLayouts
	profilo: ListDomainObjectTypes
	profilo: ListDomains
	profilo: ListEventStreams
	profilo: ListIdentityResolutionJobs
	profilo: ListIntegrations

Prefisso del servizio	Azioni
	profilo: ListObjectTypeAttributeValues
	profilo: ListObjectTypeAttributes
	profilo: ListProfileAttributeValues
	profilo: ListProfileObjectTypeTemplates
	profilo: ListProfileObjectTypes
	profilo: ListProfileObjects
	profilo: ListRecommenderFilters
	profilo: ListRecommenderRecipes
	profilo: ListRecommenderSchemas
	profilo: ListRecommenders
	profilo: ListRuleBasedMatches
	profilo: ListSegmentDefinitions
	profilo: ListUploadJobs
	profilo: ListWorkflows
	profilo: MergeProfiles
	profilo: PutDomainObjectType
	profilo: PutIntegration
	profilo: PutProfileObject
	profilo: PutProfileObjectType
	profilo: SearchProfiles
	profilo: StartRecommender

Prefisso del servizio	Azioni
	profilo: StartUploadJob profilo: StopRecommender profilo: StopUploadJob profilo: UpdateCalculatedAttributeDefinition profilo: UpdateDomain profilo: UpdateProfile profilo: UpdateRecommender

Prefisso del servizio	Azioni
qldb	qldb: CancelJournalKinesisStream qldb: CreateLedger qldb: DeleteLedger qldb: DescribeJournalKinesisStream qldb: DescribeJournalS3Export qldb: DescribeLedger qldb: ExportJournalToS3 qldb: GetBlock qldb: GetDigest qldb: GetRevision qldb: ListJournalKinesisStreamsForLedger qldb: ListJournalS3Exports qldb: ListJournalS3ExportsForLedger qldb: ListLedgers qldb: StreamJournalToKinesis qldb: UpdateLedger qldb: UpdateLedgerPermissionsMode

Prefisso del servizio	Azioni
ram	ram: AcceptResourceShareInvitation
	ram: AssociateResourceShare
	ram: AssociateResourceSharePermission
	ram: CreatePermission
	ram: CreatePermissionVersion
	ram: CreateResourceShare
	ram: DeletePermission
	ram: DeletePermissionVersion
	ram: DeleteResourceShare
	ram: DisassociateResourceShare
	ram: DisassociateResourceSharePermission
	ram: EnableSharingWithAwsOrganization
	ram: GetPermission
	ram: GetResourcePolicies
	ram: GetResourceShareAssociations
	ram: GetResourceShareInvitations
	ram: GetResourceShares
	ram: ListPendingInvitationResources
	ram: ListPermissionAssociations
	ram: ListPermissionVersions
	ram: ListPermissions

Prefisso del servizio	Azioni
	ram: ListPrincipals ram: ListReplacePermissionAssociationsWork ram: ListResourceSharePermissions ram: ListResourceTypes ram: ListResources ram: ListSourceAssociations ram: PromotePermissionCreatedFromPolicy ram: PromoteResourceShareCreatedFromPolicy ram: RejectResourceShareInvitation ram: ReplacePermissionAssociations ram: SetDefaultPermissionVersion ram: UpdateResourceShare
rbin	pettine: CreateRule rbin: DeleteRule rbin: GetRule rbin: ListRules rbin: LockRule rbin: UnlockRule rbin: UpdateRule

Prefisso del servizio	Azioni
rds	rds: AddRoleToDBCluster
	rds: AddRoleToDBInstance
	rds: AddSourceIdentifierToSubscription
	rds: ApplyPendingMaintenanceAction
	rds: AuthorizeDBSecurityGroupIngress
	rds:BacktrackDBCluster
	rds: CancelExportTask
	rds: CopyDBClusterParameterGroup
	rds: CopyDBClusterSnapshot
	rds: CopyDBParameterGroup
	rds:CopyDBSnapshot
	rds: CopyOptionGroup
	rds: CreateCustomDBEngineVersion
	rds: CreateDBClusterParameterGroup
	rds: CreateDBParameterGroup
	rds:CreateDBProxy
	rds: CreateDBProxyEndpoint
	rds: CreateDBSecurityGroup
	rds: CreateDBSubnetGroup
	rds: CreateEventSubscription
	rds: CreateGlobalCluster

Prefisso del servizio	Azioni
	rds: CreateOptionGroup
	rds: DeleteBlueGreenDeployment
	rds: DeleteDBClusterAutomatedBackup
	rds: DeleteDBClusterParameterGroup
	rds: DeleteDBClusterSnapshot
	rds: DeleteDBInstanceAutomatedBackup
	rds: DeleteDBParameterGroup
	rds:DeleteDBProxy
	rds: DeleteDBProxyEndpoint
	rds: DeleteDBSecurityGroup
	rds:DeleteDBSnapshot
	rds: DeleteDBSubnetGroup
	rds: DeleteEventSubscription
	rds: DeleteGlobalCluster
	rds: DeleteOptionGroup
	rds: DeregisterDBProxyTargets
	rds: DescribeAccountAttributes
	rds: DescribeBlueGreenDeployments
	rds: DescribeCertificates
	rds: DescribeDBClusterAutomatedBackups
	rds: DescribeDBClusterBacktracks

Prefisso del servizio	Azioni
	rds: DescribeDBClusterEndpoints
	rds: DescribeDBClusterParameterGroups
	rds: DescribeDBClusterParameters
	rds: DescribeDBClusterSnapshotAttributes
	rds: DescribeDBClusterSnapshots
	rds:DescribeDBClusters
	rds: DescribeDBEngineVersions
	rds: DescribeDBInstanceAutomatedBackups
	rds:DescribeDBInstances
	rds: DescribeDBLogFiles
	rds: DescribeDBMajorEngineVersions
	rds: DescribeDBParameterGroups
	rds:DescribeDBParameters
	rds:DescribeDBProxies
	rds: DescribeDBProxyEndpoints
	rds: DescribeDBProxyTargetGroups
	rds: DescribeDBProxyTargets
	rds:DescribeDBRecommendations
	rds: DescribeDBSecurityGroups
	rds: DescribeDBSnapshotAttributes
	rds: DescribeDBSnapshotTenantDatabases

Prefisso del servizio	Azioni
	rds:DescribeDBSnapshots
	rds: DescribeDBSubnetGroups
	rds: DescribeEngineDefaultClusterParameters
	rds: DescribeEngineDefaultParameters
	rds: DescribeEventCategories
	rds: DescribeEventSubscriptions
	rds: DescribeEvents
	rds: DescribeExportTasks
	rds: DescribeGlobalClusters
	rds: DescribeIntegrations
	rds: DescribeOptionGroupOptions
	rds: DescribeOptionGroups
	rds: DescribeOrderableDBInstanceOptions
	rds: DescribePendingMaintenanceActions
	rds: DescribeReservedDBInstances
	rds: DescribeReservedDBInstancesOfferings
	rds: DescribeSourceRegions
	rds: DescribeTenantDatabases
	rds: DescribeValidDBInstanceModifications
	rds: DownloadCompleteDBLogFile
	rds: DownloadDBLogFilePortion

Prefisso del servizio	Azioni
	rds:FailoverDBCluster
	rds: FailoverGlobalCluster
	rds: ModifyActivityStream
	rds: ModifyCertificates
	rds: ModifyCurrentDBClusterCapacity
	rds: ModifyDBClusterEndpoint
	rds: ModifyDBClusterParameterGroup
	rds: ModifyDBClusterSnapshotAttribute
	rds: ModifyDBParameterGroup
	rds:ModifyDBProxy
	rds: ModifyDBProxyEndpoint
	rds: ModifyDBProxyTargetGroup
	rds:ModifyDBRecommendation
	rds:ModifyDBSnapshot
	rds: ModifyDBSnapshotAttribute
	rds: ModifyDBSubnetGroup
	rds: ModifyEventSubscription
	rds: ModifyGlobalCluster
	rds: ModifyOptionGroup
	rds: ModifyTenantDatabase
	rds: PurchaseReservedDBInstancesOffering

Prefisso del servizio	Azioni
	rds:RebootDBCluster
	rds: RegisterDBProxyTargets
	rds: RemoveFromGlobalCluster
	rds: RemoveRoleFromDBCluster
	rds: RemoveRoleFromDBInstance
	rds: RemoveSourceIdentifierFromSubscription
	rds: ResetDBClusterParameterGroup
	rds: ResetDBParameterGroup
	rds: RestoreDBClusterFromS3
	rds: RestoreDBClusterFromSnapshot
	rds: RestoreDBClusterToPointInTime
	rds: RestoreDBInstanceFromDBSnapshot
	rds: RestoreDBInstanceFromS3
	rds: RestoreDBInstanceToPointInTime
	rds: RevokeDBSecurityGroupIngress
	rds: StartActivityStream
	rds:StartDBCluster
	rds:StartDBInstance
	rds: StartDBInstanceAutomatedBackupsReplication
	rds: StartExportTask
	rds: StopActivityStream

Prefisso del servizio	Azioni
	rds:StopDBCluster rds:StopDBInstance rds: StopDBInstanceAutomatedBackupsReplication rds: SwitchoverBlueGreenDeployment rds: SwitchoverGlobalCluster rds: SwitchoverReadReplica

Prefisso del servizio	Azioni
redshift	spostamento verso il rosso: AcceptReservedNodeExchange spostamento verso il rosso: AddPartner spostamento verso il rosso: AssociateDataShareConsumer spostamento verso il rosso: AuthorizeClusterSecurityGroupIngress spostamento verso il rosso: AuthorizeDataShare spostamento verso il rosso: AuthorizeEndpointAccess spostamento verso il rosso: AuthorizeSnapshotAccess spostamento verso il rosso: BatchDeleteClusterSnapshots spostamento verso il rosso: BatchModifyClusterSnapshots spostamento verso il rosso: CancelResize spostamento verso il rosso: CopyClusterSnapshot spostamento verso il rosso: CreateAuthenticationProfile spostamento verso il rosso: CreateCluster spostamento verso il rosso: CreateClusterParameterGroup spostamento verso il rosso: CreateClusterSecurityGroup spostamento verso il rosso: CreateClusterSnapshot spostamento verso il rosso: CreateClusterSubnetGroup spostamento verso il rosso: CreateCustomDomainAssociation spostamento verso il rosso: CreateEndpointAccess spostamento verso il rosso: CreateEventSubscription spostamento verso il rosso: CreateHsmClientCertificate

Prefisso del servizio	Azioni
	spostamento verso il rosso: CreateHsmConfiguration spostamento verso il rosso: CreateIntegration spostamento verso il rosso: CreateRedshiftIdcApplication spostamento verso il rosso: CreateScheduledAction spostamento verso il rosso: CreateSnapshotCopyGrant spostamento verso il rosso: CreateSnapshotSchedule spostamento verso il rosso: CreateUsageLimit spostamento verso il rosso: DeauthorizeDataShare spostamento verso il rosso: DeleteAuthenticationProfile spostamento verso il rosso: DeleteCluster spostamento verso il rosso: DeleteClusterParameterGroup spostamento verso il rosso: DeleteClusterSecurityGroup spostamento verso il rosso: DeleteClusterSnapshot spostamento verso il rosso: DeleteClusterSubnetGroup spostamento verso il rosso: DeleteCustomDomainAssociation spostamento verso il rosso: DeleteEndpointAccess spostamento verso il rosso: DeleteEventSubscription spostamento verso il rosso: DeleteHsmClientCertificate spostamento verso il rosso: DeleteHsmConfiguration spostamento verso il rosso: DeletePartner spostamento verso il rosso: DeleteRedshiftIdcApplication

Prefisso del servizio	Azioni
	spostamento verso il rosso: DeleteResourcePolicy spostamento verso il rosso: DeleteScheduledAction spostamento verso il rosso: DeleteSnapshotCopyGrant spostamento verso il rosso: DeleteSnapshotSchedule spostamento verso il rosso: DeleteUsageLimit spostamento verso il rosso: DeregisterNamespace spostamento verso il rosso: DescribeAccountAttributes spostamento verso il rosso: DescribeAuthenticationProfiles spostamento verso il rosso: DescribeClusterDbRevisions spostamento verso il rosso: DescribeClusterParameterGroups spostamento verso il rosso: DescribeClusterParameters spostamento verso il rosso: DescribeClusterSecurityGroups spostamento verso il rosso: DescribeClusterSnapshots spostamento verso il rosso: DescribeClusterSubnetGroups spostamento verso il rosso: DescribeClusterTracks spostamento verso il rosso: DescribeClusterVersions spostamento verso il rosso: DescribeClusters spostamento verso il rosso: DescribeCustomDomainAssociations spostamento verso il rosso: DescribeDataShares spostamento verso il rosso: DescribeDataSharesForConsumer spostamento verso il rosso: DescribeDataSharesForProducer

Prefisso del servizio	Azioni
	spostamento verso il rosso: DescribeDefaultClusterParameters spostamento verso il rosso: DescribeEndpointAccess spostamento verso il rosso: DescribeEndpointAuthorization spostamento verso il rosso: DescribeEventCategories spostamento verso il rosso: DescribeEventSubscriptions spostamento verso il rosso: DescribeEvents spostamento verso il rosso: DescribeHsmClientCertificates spostamento verso il rosso: DescribeHsmConfigurations spostamento verso il rosso: DescribeInboundIntegrations spostamento verso il rosso: DescribeIntegrations spostamento verso il rosso: DescribeLoggingStatus spostamento verso il rosso: DescribeNodeConfigurationOptions spostamento verso il rosso: DescribeOrderableClusterOptions spostamento verso il rosso: DescribePartners spostamento verso il rosso: DescribeRedshiftIdcApplications spostamento verso il rosso: DescribeReservedNodeExchangeStatus spostamento verso il rosso: DescribeReservedNodeOfferings spostamento verso il rosso: DescribeReservedNodes spostamento verso il rosso: DescribeResize spostamento verso il rosso: DescribeScheduledActions

Prefisso del servizio	Azioni
	spostamento verso il rosso: DescribeSnapshotCopyGrants spostamento verso il rosso: DescribeSnapshotSchedules spostamento verso il rosso: DescribeStorage spostamento verso il rosso: DescribeTableRestoreStatus spostamento verso il rosso: DescribeUsageLimits spostamento verso il rosso: DisableLogging spostamento verso il rosso: DisableSnapshotCopy spostamento verso il rosso: DisassociateDataShareConsumer spostamento verso il rosso: EnableLogging spostamento verso il rosso: EnableSnapshotCopy spostamento verso il rosso: FailoverPrimaryCompute spostamento verso il rosso: GetClusterCredentials spostamento verso il rosso: GetClusterCredentialsWithIAM spostamento verso il rosso: GetIdentityCenterAuthToken spostamento verso il rosso: GetReservedNodeExchangeConfigurationOptions spostamento verso il rosso: GetReservedNodeExchangeOfferings spostamento verso il rosso: GetResourcePolicy spostamento verso il rosso: ListRecommendations spostamento verso il rosso: ModifyAquaConfiguration spostamento verso il rosso: ModifyAuthenticationProfile

Prefisso del servizio	Azioni
	spostamento verso il rosso: ModifyCluster spostamento verso il rosso: ModifyClusterDbRevision spostamento verso il rosso: ModifyClusterIamRoles spostamento verso il rosso: ModifyClusterMaintenance spostamento verso il rosso: ModifyClusterParameterGroup spostamento verso il rosso: ModifyClusterSnapshot spostamento verso il rosso: ModifyClusterSnapshotSchedule spostamento verso il rosso: ModifyClusterSubnetGroup spostamento verso il rosso: ModifyCustomDomainAssociation spostamento verso il rosso: ModifyEndpointAccess spostamento verso il rosso: ModifyEventSubscription spostamento verso il rosso: ModifyRedshiftIdcApplication spostamento verso il rosso: ModifyScheduledAction spostamento verso il rosso: ModifySnapshotCopyRetentionPeriod spostamento verso il rosso: ModifySnapshotSchedule spostamento verso il rosso: ModifyUsageLimit spostamento verso il rosso: PauseCluster spostamento verso il rosso: PurchaseReservedNodeOffering spostamento verso il rosso: PutResourcePolicy spostamento verso il rosso: RebootCluster spostamento verso il rosso: RegisterNamespace

Prefisso del servizio	Azioni
	spostamento verso il rosso: RejectDataShare spostamento verso il rosso: ResetClusterParameterGroup spostamento verso il rosso: ResizeCluster spostamento verso il rosso: RestoreFromClusterSnapshot spostamento verso il rosso: RestoreTableFromClusterSnapshot spostamento verso il rosso: ResumeCluster spostamento verso il rosso: RevokeClusterSecurityGroupIngress spostamento verso il rosso: RevokeEndpointAccess spostamento verso il rosso: RevokeSnapshotAccess spostamento verso il rosso: RotateEncryptionKey spostamento verso il rosso: UpdatePartnerStatus

Prefisso del servizio	Azioni
redshift-data	dati redshift: BatchExecuteStatement dati redshift: CancelStatement dati redshift: DescribeStatement dati redshift: DescribeTable dati redshift: ExecuteStatement dati redshift: GetStatementResult dati redshift: ListDatabases dati redshift: ListSchemas dati redshift: ListSessions dati redshift: ListStatements dati redshift: ListTables

Prefisso del servizio	Azioni
refactor-spaces	spazi refactorici: CreateApplication spazi di refattore: CreateEnvironment spazi di refattore: CreateRoute spazi di refattore: CreateService spazi di refattore: DeleteApplication spazi di refattore: DeleteEnvironment spazi di refattore: DeleteResourcePolicy spazi di refattore: DeleteRoute spazi di refattore: DeleteService spazi di refattore: GetApplication spazi di refattore: GetEnvironment spazi di refattore: GetResourcePolicy spazi di refattore: GetRoute spazi di refattore: GetService spazi di refattore: ListApplications spazi di refattore: ListEnvironmentVpcs spazi di refattore: ListEnvironments spazi di refattore: ListRoutes spazi di refattore: ListServices spazi di refattore: PutResourcePolicy spazi di refattore: UpdateRoute

Prefisso del servizio	Azioni
rekognition	riconoscimento: AssociateFaces
	riconoscimento: CompareFaces
	riconoscimento: CopyProjectVersion
	riconoscimento: CreateCollection
	riconoscimento: CreateDataset
	riconoscimento: CreateFaceLivenessSession
	riconoscimento: CreateProject
	riconoscimento: CreateProjectVersion
	riconoscimento: CreateStreamProcessor
	riconoscimento: CreateUser
	riconoscimento: DeleteCollection
	riconoscimento: DeleteDataset
	riconoscimento: DeleteFaces
	riconoscimento: DeleteProject
	riconoscimento: DeleteProjectPolicy
	riconoscimento: DeleteProjectVersion
	riconoscimento: DeleteStreamProcessor
	riconoscimento: DeleteUser
	riconoscimento: DescribeCollection
	riconoscimento: DescribeDataset
	riconoscimento: DescribeProjectVersions

Prefisso del servizio	Azioni
	riconoscimento: DescribeProjects
	riconoscimento: DescribeStreamProcessor
	riconoscimento: DetectCustomLabels
	riconoscimento: DetectFaces
	riconoscimento: DetectLabels
	riconoscimento: DetectModerationLabels
	riconoscimento: DetectProtectiveEquipment
	riconoscimento: DetectText
	riconoscimento: DisassociateFaces
	riconoscimento: DistributeDatasetEntries
	riconoscimento: GetCelebrityInfo
	riconoscimento: GetCelebrityRecognition
	riconoscimento: GetContentModeration
	riconoscimento: GetFaceDetection
	riconoscimento: GetFaceLivenessSessionResults
	riconoscimento: GetFaceSearch
	riconoscimento: GetLabelDetection
	riconoscimento: GetMediaAnalysisJob
	riconoscimento: GetPersonTracking
	riconoscimento: GetSegmentDetection
	riconoscimento: GetTextDetection

Prefisso del servizio	Azioni
	riconoscimento: IndexFaces
	riconoscimento: ListCollections
	riconoscimento: ListDatasetEntries
	riconoscimento: ListDatasetLabels
	riconoscimento: ListFaces
	riconoscimento: ListMediaAnalysisJobs
	riconoscimento: ListProjectPolicies
	riconoscimento: ListStreamProcessors
	riconoscimento: ListUsers
	riconoscimento: PutProjectPolicy
	riconoscimento: RecognizeCelebrities
	riconoscimento: SearchFaces
	riconoscimento: SearchFacesByImage
	riconoscimento: SearchUsers
	riconoscimento: SearchUsersByImage
	riconoscimento: StartCelebrityRecognition
	riconoscimento: StartContentModeration
	riconoscimento: StartFaceDetection
	riconoscimento: StartFaceLivenessSession
	riconoscimento: StartFaceSearch
	riconoscimento: StartLabelDetection

Prefisso del servizio	Azioni
	riconoscimento: StartMediaAnalysisJob
	riconoscimento: StartPersonTracking
	riconoscimento: StartProjectVersion
	riconoscimento: StartSegmentDetection
	riconoscimento: StartStreamProcessor
	riconoscimento: StartTextDetection
	riconoscimento: StopProjectVersion
	riconoscimento: StopStreamProcessor
	riconoscimento: UpdateDatasetEntries
	riconoscimento: UpdateStreamProcessor

Prefisso del servizio	Azioni
resiliencehub	hub di resilienza: AcceptResourceGroupingRecommendations hub di resilienza: AddDraftAppVersionResourceMappings hub di resilienza: BatchUpdateRecommendationStatus hub di resilienza: CreateApp hub di resilienza: CreateAppVersionAppComponent hub di resilienza: CreateAppVersionResource hub di resilienza: CreateRecommendationTemplate hub di resilienza: CreateResiliencyPolicy hub di resilienza: DeleteApp hub di resilienza: DeleteAppAssessment hub di resilienza: DeleteAppInputSource hub di resilienza: DeleteAppVersionAppComponent hub di resilienza: DeleteAppVersionResource hub di resilienza: DeleteRecommendationTemplate hub di resilienza: DeleteResiliencyPolicy hub di resilienza: DescribeApp hub di resilienza: DescribeAppAssessment hub di resilienza: DescribeAppVersion hub di resilienza: DescribeAppVersionAppComponent hub di resilienza: DescribeAppVersionResource hub di resilienza: DescribeAppVersionResourcesResolutionStatus

Prefisso del servizio	Azioni
	hub di resilienza: DescribeAppVersionTemplate hub di resilienza: DescribeDraftAppVersionResourcesImportStatus hub di resilienza: DescribeMetricsExport hub di resilienza: DescribeResiliencyPolicy hub di resilienza: DescribeResourceGroupingRecommendationTask hub di resilienza: ImportResourcesToDraftAppVersion hub di resilienza: ListAlarmRecommendations hub di resilienza: ListAppAssessmentComplianceDrifts hub di resilienza: ListAppAssessmentResourceDrifts hub di resilienza: ListAppAssessments hub di resilienza: ListAppComponentCompliances hub di resilienza: ListAppComponentRecommendations hub di resilienza: ListAppInputSources hub di resilienza: ListAppVersionAppComponents hub di resilienza: ListAppVersionResourceMappings hub di resilienza: ListAppVersionResources hub di resilienza: ListAppVersions hub di resilienza: ListApps hub di resilienza: ListMetrics hub di resilienza: ListRecommendationTemplates hub di resilienza: ListResiliencyPolicies

Prefisso del servizio	Azioni
	hub di resilienza: ListResourceGroupingRecommendations hub di resilienza: ListSopRecommendations hub di resilienza: ListSuggestedResiliencyPolicies hub di resilienza: ListTestRecommendations hub di resilienza: ListUnsupportedAppVersionResources hub di resilienza: PublishAppVersion hub di resilienza: PutDraftAppVersionTemplate hub di resilienza: RejectResourceGroupingRecommendations hub di resilienza: RemoveDraftAppVersionResourceMappings hub di resilienza: ResolveAppVersionResources hub di resilienza: StartAppAssessment hub di resilienza: StartResourceGroupingRecommendationTask hub di resilienza: UpdateApp hub di resilienza: UpdateAppVersion hub di resilienza: UpdateAppVersionAppComponent hub di resilienza: UpdateAppVersionResource hub di resilienza: UpdateResiliencyPolicy

Prefisso del servizio	Azioni
resource-explorer-2	resource-explorer-2: AssociateDefaultView
	esploratore-risorsa-2: BatchGetView
	esploratore-risorsa-2: CreateIndex
	esploratore-risorsa-2: CreateResourceExplorerSetup
	esploratore-risorsa-2: CreateView
	esploratore-risorsa-2: DeleteIndex
	esploratore-risorsa-2: DeleteResourceExplorerSetup
	esploratore-risorsa-2: DeleteView
	esploratore-risorsa-2: DisassociateDefaultView
	esploratore-risorsa-2: GetAccountLevelServiceConfiguration
	esploratore-risorsa-2: GetDefaultView
	esploratore-risorsa-2: GetIndex
	esploratore-risorsa-2: GetManagedView
	esploratore-risorsa-2: GetResourceExplorerSetup
	esploratore-risorsa-2: GetServiceIndex
	esploratore-risorsa-2: GetServiceView
	esploratore-risorsa-2: ListIndexes
	esploratore-risorsa-2: ListIndexesForMembers
	esploratore-risorsa-2: ListManagedViews
	esploratore-risorsa-2: ListServiceIndexes
	esploratore-risorsa-2: ListServiceViews

Prefisso del servizio	Azioni
	esploratore-risorsa-2: ListStreamingAccessForServices
	esploratore-risorsa-2: ListSupportedResourceTypes
	esploratore-risorsa-2: ListViews
	resource-explorer-2:Search
	esploratore-risorsa-2: UpdateIndexType
	esploratore-risorsa-2: UpdateView

Prefisso del servizio	Azioni
resource-groups	gruppi di risorse: CancelTagSyncTask gruppi di risorse: GetAccountSettings gruppi di risorse: GetGroup gruppi di risorse: GetGroupConfiguration gruppi di risorse: GetGroupQuery gruppi di risorse: GetTagSyncTask gruppi di risorse: GroupResources gruppi di risorse: ListGroupResources gruppi di risorse: ListGroupingStatuses gruppi di risorse: ListGroups gruppi di risorse: ListTagSyncTasks gruppi di risorse: PutGroupConfiguration gruppi di risorse: SearchResources gruppi di risorse: StartTagSyncTask gruppi di risorse: UngroupResources gruppi di risorse: UpdateAccountSettings gruppi di risorse: UpdateGroup gruppi di risorse: UpdateGroupQuery

Prefisso del servizio	Azioni
robomaker	robomaker: BatchDeleteWorlds
	robomaker: BatchDescribeSimulationJob
	robomaker: CancelDeploymentJob
	robomaker: CancelSimulationJob
	robomaker: CancelSimulationJobBatch
	robomaker: CancelWorldExportJob
	robomaker: CancelWorldGenerationJob
	robomaker: CreateDeploymentJob
	robomaker: CreateFleet
	robomaker: CreateRobot
	robomaker: CreateRobotApplication
	robomaker: CreateRobotApplicationVersion
	robomaker: CreateSimulationApplication
	robomaker: CreateSimulationApplicationVersion
	robomaker: CreateSimulationJob
	robomaker: CreateWorldExportJob
	robomaker: CreateWorldGenerationJob
	robomaker: CreateWorldTemplate
	robomaker: DeleteFleet
	robomaker: DeleteRobot
	robomaker: DeleteRobotApplication

Prefisso del servizio	Azioni
	robomaker: DeleteSimulationApplication
	robomaker: DeleteWorldTemplate
	robomaker: DeregisterRobot
	robomaker: DescribeDeploymentJob
	robomaker: DescribeFleet
	robomaker: DescribeRobot
	robomaker: DescribeRobotApplication
	robomaker: DescribeSimulationApplication
	robomaker: DescribeSimulationJob
	robomaker: DescribeSimulationJobBatch
	robomaker: DescribeWorld
	robomaker: DescribeWorldExportJob
	robomaker: DescribeWorldGenerationJob
	robomaker: DescribeWorldTemplate
	robomaker: GetWorldTemplateBody
	robomaker: ListDeploymentJobs
	robomaker: ListFleets
	robomaker: ListRobotApplications
	robomaker: ListRobots
	robomaker: ListSimulationApplications
	robomaker: ListSimulationJobBatches

Prefisso del servizio	Azioni
	robomaker: ListSimulationJobs
	robomaker: ListWorldExportJobs
	robomaker: ListWorldGenerationJobs
	robomaker: ListWorldTemplates
	robomaker: ListWorlds
	robomaker: RegisterRobot
	robomaker: RestartSimulationJob
	robomaker: StartSimulationJobBatch
	robomaker: SyncDeploymentJob
	robomaker: UpdateRobotApplication
	robomaker: UpdateSimulationApplication
	robomaker: UpdateWorldTemplate

Prefisso del servizio	Azioni
rolesanywhere	ruoli ovunque: CreateProfile ruoli ovunque: CreateTrustAnchor ruoli ovunque: DeleteAttributeMapping ruoli ovunque: DeleteCrl ruoli ovunque: DeleteProfile ruoli ovunque: DeleteTrustAnchor ruoli ovunque: DisableCrl ruoli ovunque: DisableProfile ruoli ovunque: DisableTrustAnchor ruoli ovunque: EnableCrl ruoli ovunque: EnableProfile ruoli ovunque: EnableTrustAnchor ruoli ovunque: GetCrl ruoli ovunque: GetProfile ruoli ovunque: GetSubject ruoli ovunque: GetTrustAnchor ruoli ovunque: ImportCrl ruoli ovunque: ListCrls ruoli ovunque: ListProfiles ruoli ovunque: ListSubjects ruoli ovunque: ListTrustAnchors

Prefisso del servizio	Azioni
	ruoli ovunque: PutAttributeMapping ruoli ovunque: PutNotificationSettings ruoli ovunque: ResetNotificationSettings ruoli ovunque: UpdateCrl ruoli ovunque: UpdateProfile ruoli ovunque: UpdateTrustAnchor

Prefisso del servizio	Azioni
route53	percorso 53: ActivateKeySigningKey
	percorso 53: AssociateVPCWithHostedZone
	percorso 53: ChangeCidrCollection
	percorso 53: ChangeResourceRecordSets
	percorso 53: CreateCidrCollection
	percorso 53: CreateHealthCheck
	percorso 53: CreateHostedZone
	percorso 53: CreateKeySigningKey
	percorso 53: CreateQueryLoggingConfig
	percorso 53: CreateReusableDelegationSet
	percorso 53: CreateTrafficPolicy
	percorso 53: CreateTrafficPolicyInstance
	percorso 53: CreateTrafficPolicyVersion
	percorso 53: CreateVPCAssociationAuthorization
	percorso 53: DeactivateKeySigningKey
	percorso 53: DeleteCidrCollection
	percorso 53: DeleteHealthCheck
	percorso 53: DeleteHostedZone
	percorso 53: DeleteKeySigningKey
	percorso 53: DeleteQueryLoggingConfig
	percorso 53: DeleteReusableDelegationSet

Prefisso del servizio	Azioni
	percorso 53: DeleteTrafficPolicy
	percorso 53: DeleteTrafficPolicyInstance
	percorso 53: DeleteVPCAssociationAuthorization
	percorso 53: DisableHostedZoneDNSSEC
	percorso 53: DisassociateVPCFromHostedZone
	percorso 53: EnableHostedZoneDNSSEC
	percorso 53: GetAccountLimit
	percorso 53: GetChange
	percorso 53: GetCheckerIpRanges
	route53:GetDNSSEC
	percorso 53: GetGeoLocation
	percorso 53: GetHealthCheck
	percorso 53: GetHealthCheckCount
	percorso 53: GetHealthCheckLastFailureReason
	percorso 53: GetHealthCheckStatus
	percorso 53: GetHostedZone
	percorso 53: GetHostedZoneCount
	percorso 53: GetHostedZoneLimit
	percorso 53: GetQueryLoggingConfig
	percorso 53: GetReusableDelegationSet
	percorso 53: GetReusableDelegationSetLimit

Prefisso del servizio	Azioni
	percorso 53: GetTrafficPolicy
	percorso 53: GetTrafficPolicyInstance
	percorso 53: GetTrafficPolicyInstanceCount
	percorso 53: ListCidrBlocks
	percorso 53: ListCidrCollections
	percorso 53: ListCidrLocations
	percorso 53: ListGeoLocations
	percorso 53: ListHealthChecks
	percorso 53: ListHostedZones
	percorso 53: ListHostedZonesByName
	percorso 53: ListHostedZonesByVPC
	percorso 53: ListQueryLoggingConfigs
	percorso 53: ListResourceRecordSets
	percorso 53: ListReusableDelegationSets
	percorso 53: ListTrafficPolicies
	percorso 53: ListTrafficPolicyInstances
	percorso 53: ListTrafficPolicyInstancesByHostedZone
	percorso 53: ListTrafficPolicyInstancesByPolicy
	percorso 53: ListTrafficPolicyVersions
	percorso 53: ListVPCAssociationAuthorizations
	route53:TestDNSAnswer

Prefisso del servizio	Azioni
	percorso 53: UpdateHealthCheck
	percorso 53: UpdateHostedZoneComment
	percorso 53: UpdateTrafficPolicyComment
	percorso 53: UpdateTrafficPolicyInstance

Prefisso del servizio	Azioni
route53-recovery-control-config	route53-recovery-control-config: CreateCluster route53-recovery-control-config: CreateControlPanel route53-recovery-control-config: CreateRoutingControl route53-recovery-control-config: CreateSafetyRule route53-recovery-control-config: DeleteCluster route53-recovery-control-config: DeleteControlPanel route53-recovery-control-config: DeleteRoutingControl route53-recovery-control-config: DeleteSafetyRule route53-recovery-control-config: DescribeCluster route53-recovery-control-config: DescribeControlPanel route53-recovery-control-config: DescribeRoutingControl route53-recovery-control-config: DescribeSafetyRule route53-recovery-control-config: GetResourcePolicy route53-recovery-control-config: ListAssociatedRoute53HealthChecks route53-recovery-control-config: ListClusters route53-recovery-control-config: ListControlPanels route53-recovery-control-config: ListRoutingControls route53-recovery-control-config: ListSafetyRules route53-recovery-control-config: UpdateCluster route53-recovery-control-config: UpdateControlPanel

Prefisso del servizio	Azioni
	route53-recovery-control-config: UpdateRoutingControl route53-recovery-control-config: UpdateSafetyRule

Prefisso del servizio	Azioni
route53-recovery-readiness	route53-prontezza per il ripristino: CreateCell
	predisposizione al ripristino del route53: CreateCrossAccount Authorization
	predisposizione al ripristino del route53: CreateReadinessCheck
	predisposizione al ripristino del route53: CreateRecoveryGroup
	predisposizione al ripristino del route53: CreateResourceSet
	predisposizione al ripristino del route53: DeleteCell
	predisposizione al ripristino del route53: DeleteCrossAccount Authorization
	predisposizione al ripristino del route53: DeleteReadinessCheck
	predisposizione al ripristino del route53: DeleteRecoveryGroup
	predisposizione al ripristino del route53: DeleteResourceSet
	predisposizione al ripristino del route53: GetArchitectureRecommendations
	predisposizione al ripristino del route53: GetCell
	predisposizione al ripristino del route53: GetCellReadinessSummary
	predisposizione al ripristino del route53: GetReadinessCheck
	predisposizione al ripristino del route53: GetReadinessCheckResourceStatus
	predisposizione al ripristino del route53: GetReadinessCheckStatus
	predisposizione al ripristino del route53: GetRecoveryGroup

Prefisso del servizio	Azioni
	predisposizione al ripristino del route53: GetRecoveryGroupReadinessSummary
	predisposizione al ripristino del route53: GetResourceSet
	predisposizione al ripristino del route53: ListCells
	predisposizione al ripristino del route53: ListCrossAccountAuthorizations
	predisposizione al ripristino del route53: ListReadinessChecks
	predisposizione al ripristino del route53: ListRecoveryGroups
	predisposizione al ripristino del route53: ListResourceSets
	predisposizione al ripristino del route53: ListRules
	predisposizione al ripristino del route53: UpdateCell
	predisposizione al ripristino del route53: UpdateReadinessCheck
	predisposizione al ripristino del route53: UpdateRecoveryGroup
	predisposizione al ripristino del route53: UpdateResourceSet

Prefisso del servizio	Azioni
route53resolver	resolver route53: AssociateFirewallRuleGroup resolver route53: AssociateResolverEndpointIpAddress resolver route53: AssociateResolverQueryLogConfig resolver route53: AssociateResolverRule resolver route53: CreateFirewallDomainList resolver route53: CreateFirewallRule resolver route53: CreateFirewallRuleGroup resolver route53: CreateResolverEndpoint resolver route53: CreateResolverQueryLogConfig resolver route53: CreateResolverRule resolver route53: DeleteFirewallDomainList resolver route53: DeleteFirewallRule resolver route53: DeleteFirewallRuleGroup resolver route53: DeleteOutpostResolver resolver route53: DeleteResolverEndpoint resolver route53: DeleteResolverQueryLogConfig resolver route53: DeleteResolverRule resolver route53: DisassociateFirewallRuleGroup resolver route53: DisassociateResolverEndpointIpAddress resolver route53: DisassociateResolverQueryLogConfig resolver route53: DisassociateResolverRule

Prefisso del servizio	Azioni
	resolver route53: GetFirewallConfig resolver route53: GetFirewallDomainList resolver route53: GetFirewallRuleGroup resolver route53: GetFirewallRuleGroupAssociation resolver route53: GetFirewallRuleGroupPolicy resolver route53: GetOutpostResolver resolver route53: GetResolverConfig resolver route53: GetResolverDnssecConfig resolver route53: GetResolverEndpoint resolver route53: GetResolverQueryLogConfig resolver route53: GetResolverQueryLogConfigAssociation resolver route53: GetResolverQueryLogConfigPolicy resolver route53: GetResolverRule resolver route53: GetResolverRuleAssociation resolver route53: GetResolverRulePolicy resolver route53: ImportFirewallDomains resolver route53: ListFirewallConfigs resolver route53: ListFirewallDomainLists resolver route53: ListFirewallDomains resolver route53: ListFirewallRuleGroupAssociations resolver route53: ListFirewallRuleGroups

Prefisso del servizio	Azioni
	resolver route53: ListFirewallRuleTypes resolver route53: ListFirewallRules resolver route53: ListOutpostResolvers resolver route53: ListResolverConfigs resolver route53: ListResolverDnssecConfigs resolver route53: ListResolverEndpointIpAddresses resolver route53: ListResolverEndpoints resolver route53: ListResolverQueryLogConfigAssociations resolver route53: ListResolverQueryLogConfigs resolver route53: ListResolverRuleAssociations resolver route53: ListResolverRules resolver route53: PutFirewallRuleGroupPolicy resolver route53: PutResolverQueryLogConfigPolicy resolver route53: UpdateFirewallConfig resolver route53: UpdateFirewallDomains resolver route53: UpdateFirewallRule resolver route53: UpdateFirewallRuleGroupAssociation resolver route53: UpdateOutpostResolver resolver route53: UpdateResolverConfig resolver route53: UpdateResolverDnssecConfig resolver route53: UpdateResolverEndpoint

Prefisso del servizio	Azioni
	resolver route53: UpdateResolverRule
rum	tamburo: BatchCreateRumMetricDefinitions rum: BatchDeleteRumMetricDefinitions rum: BatchGetRumMetricDefinitions rum: CreateAppMonitor rum: DeleteAppMonitor rum: DeleteResourcePolicy rum: DeleteRumMetricsDestination rum: GetAppMonitor rum: GetAppMonitorData rum: GetResourcePolicy rum: ListAppMonitors rum: ListRumMetricsDestinations rum: PutResourcePolicy rum: PutRumMetricsDestination rum: UpdateAppMonitor rum: UpdateRumMetricDefinition

Prefisso del servizio	Azioni
s3	s3: AssociateAccessGrantsIdentityCenter s3: CreateAccessGrant s3: CreateAccessGrantsInstance s3: CreateAccessGrantsLocation s3: CreateAccessPoint s3: CreateAccessPointForObjectLambda s3: CreateBucket s3: CreateBucketMetadataTableConfiguration s3: CreateJob s3: CreateMultiRegionAccessPoint s3: DeleteAccessGrant s3: DeleteAccessGrantsInstance s3: DeleteAccessGrantsInstanceResourcePolicy s3: DeleteAccessGrantsLocation s3: DeleteAccessPoint s3: DeleteAccessPointForObjectLambda s3: DeleteAccessPointPolicy s3: DeleteAccessPointPolicyForObjectLambda s3: DeleteBucket s3: DeleteBucketMetadataTableConfiguration s3: DeleteBucketPolicy

Prefisso del servizio	Azioni
	s3: DeleteBucketWebsite s3: DeleteMultiRegionAccessPoint s3: DeleteStorageLensConfiguration s3: DescribeJob s3: DescribeMultiRegionAccessPointOperation s3: DissociateAccessGrantsIdentityCenter s3: GetAccelerateConfiguration s3: GetAccessGrant s3: GetAccessGrantsInstance s3: GetAccessGrantsInstanceForPrefix s3: GetAccessGrantsInstanceResourcePolicy s3: GetAccessGrantsLocation s3: GetAccessPoint s3: GetAccessPointConfigurationForObjectLambda s3: GetAccessPointForObjectLambda s3: GetAccessPointPolicy s3: GetAccessPointPolicyForObjectLambda s3: GetAccessPointPolicyStatus s3: GetAccessPointPolicyStatusForObjectLambda s3: GetAccountPublicAccessBlock s3: GetAnalyticsConfiguration

Prefisso del servizio	Azioni
	s3: GetBucketAbac s3: GetBucketAcl s3: NUCLEO GetBucket s3: GetBucketLocation s3: GetBucketLogging s3: GetBucketNotification s3: GetBucketObjectLockConfiguration s3: GetBucketOwnershipControls s3: GetBucketPolicy s3: GetBucketPolicyStatus s3: GetBucketPublicAccessBlock s3: GetBucketRequestPayment s3: GetBucketVersioning s3: GetBucketWebsite s3: GetDataAccess s3: GetEncryptionConfiguration s3: GetIntelligentTieringConfiguration s3: GetInventoryConfiguration s3: GetLifecycleConfiguration s3: GetMetricsConfiguration s3: GetMultiRegionAccessPoint

Prefisso del servizio	Azioni
	s3: GetMultiRegionAccessPointPolicy s3: GetMultiRegionAccessPointPolicyStatus s3: GetMultiRegionAccessPointRoutes s3: GetReplicationConfiguration s3: GetStorageLensConfiguration s3: GetStorageLensDashboard s3: ListAccessGrants s3: ListAccessGrantsInstances s3: ListAccessGrantsLocations s3: ListAccessPoints s3: ListAccessPointsForObjectLambda s3: ListAllMyBuckets s3: ListBucketMultipartUploads s3: ListCallerAccessGrants s3: ListJobs s3: ListMultiRegionAccessPoints s3: ListStorageLensConfigurations s3: PutAccelerateConfiguration s3: PutAccessGrantsInstanceResourcePolicy s3: PutAccessPointConfigurationForObjectLambda s3: PutAccessPointPolicy

Prefisso del servizio	Azioni
	s3: PutAccessPointPolicyForObjectLambda s3: PutAccountPublicAccessBlock s3: PutAnalyticsConfiguration s3: PutBucketAbac s3: PutBucketAcl s3: NUCLEO PutBucket s3: PutBucketLogging s3: PutBucketNotification s3: PutBucketObjectLockConfiguration s3: PutBucketOwnershipControls s3: PutBucketPolicy s3: PutBucketPublicAccessBlock s3: PutBucketRequestPayment s3: PutBucketVersioning s3: PutBucketWebsite s3: PutEncryptionConfiguration s3: PutIntelligentTieringConfiguration s3: PutInventoryConfiguration s3: PutLifecycleConfiguration s3: PutMetricsConfiguration s3: PutMultiRegionAccessPointPolicy

Prefisso del servizio	Azioni
	s3: PutReplicationConfiguration s3: PutStorageLensConfiguration s3: SubmitMultiRegionAccessPointRoutes s3: UpdateAccessGrantsLocation s3: UpdateBucketMetadataAnnotationTableConfiguration s3: UpdateBucketMetadataJournalTableConfiguration s3: UpdateJobPriority s3: UpdateJobStatus
s3-outposts	avamposti s3: CreateEndpoint avamposti s3: DeleteEndpoint avamposti s3: ListEndpoints avamposti s3: ListOutpostsWithS3 avamposti s3: ListSharedEndpoints

Prefisso del servizio	Azioni
sagemaker-geospatial	sagemaker geospaziale: DeleteEarthObservationJob sagemaker geospaziale: DeleteVectorEnrichmentJob sagemaker geospaziale: ExportEarthObservationJob sagemaker geospaziale: ExportVectorEnrichmentJob sagemaker geospaziale: GetEarthObservationJob sagemaker geospaziale: GetRasterDataCollection sagemaker geospaziale: GetTile sagemaker geospaziale: GetVectorEnrichmentJob sagemaker geospaziale: ListEarthObservationJobs sagemaker geospaziale: ListRasterDataCollections sagemaker geospaziale: ListVectorEnrichmentJobs sagemaker geospaziale: SearchRasterDataCollection sagemaker geospaziale: StartEarthObservationJob sagemaker geospaziale: StartVectorEnrichmentJob sagemaker geospaziale: StopEarthObservationJob sagemaker geospaziale: StopVectorEnrichmentJob

Prefisso del servizio	Azioni
savingsplans	piani di risparmio: CreateSavingsPlan
	piani di risparmio: DeleteQueuedSavingsPlan
	piani di risparmio: DescribeSavingsPlanRates
	piani di risparmio: DescribeSavingsPlans
	piani di risparmio: DescribeSavingsPlansOfferingRates
	piani di risparmio: DescribeSavingsPlansOfferings
	piani di risparmio: ReturnSavingsPlan

Prefisso del servizio	Azioni
schemas	schemi: CreateDiscoverer schemi: CreateRegistry schemi: CreateSchema schemi: DeleteDiscoverer schemi: DeleteRegistry schemi: DeleteResourcePolicy schemi: DeleteSchema schemi: DeleteSchemaVersion schemi: DescribeCodeBinding schemi: DescribeDiscoverer schemi: DescribeRegistry schemi: DescribeSchema schemi: ExportSchema schemi: GetCodeBindingSource schemi: GetDiscoveredSchema schemi: GetResourcePolicy schemi: ListDiscoverers schemi: ListRegistries schemi: ListSchemaVersions schemi: ListSchemas schemi: PutCodeBinding

Prefisso del servizio	Azioni
	schemi: PutResourcePolicy schemi: SearchSchemas schemi: StartDiscoverer schemi: StopDiscoverer schemi: UpdateDiscoverer schemi: UpdateRegistry schemi: UpdateSchema
sdb	sdb: CreateDomain sdb: DeleteDomain sdb: DomainMetadata sdb: ListDomains

Prefisso del servizio	Azioni
secretsmanager	gestore dei segreti: CancelRotateSecret gestore dei segreti: CreateSecret gestore dei segreti: DeleteResourcePolicy gestore dei segreti: DeleteSecret gestore dei segreti: DescribeSecret gestore dei segreti: GetRandomPassword gestore dei segreti: GetResourcePolicy gestore dei segreti: GetSecretValue gestore dei segreti: ListSecretVersionIds gestore dei segreti: ListSecrets gestore dei segreti: PutResourcePolicy gestore dei segreti: PutSecretValue gestore dei segreti: RemoveRegionsFromReplication gestore dei segreti: ReplicateSecretToRegions gestore dei segreti: RestoreSecret gestore dei segreti: RotateSecret gestore dei segreti: StopReplicationToReplica gestore dei segreti: UpdateSecret gestore dei segreti: ValidateResourcePolicy

Prefisso del servizio	Azioni
securityhub	hub di sicurezza: AcceptAdministratorInvitation
	hub di sicurezza: AcceptInvitation
	hub di sicurezza: BatchDeleteAutomationRules
	hub di sicurezza: BatchDisableStandards
	hub di sicurezza: BatchEnableStandards
	hub di sicurezza: BatchGetAutomationRules
	hub di sicurezza: BatchGetConfigurationPolicyAssociations
	hub di sicurezza: BatchGetEnabledRegionsV2
	hub di sicurezza: BatchGetSecurityControls
	hub di sicurezza: BatchGetStandardsControlAssociations
	hub di sicurezza: BatchImportFindings
	hub di sicurezza: BatchUpdateAutomationRules
	hub di sicurezza: BatchUpdateFindings
	hub di sicurezza: BatchUpdateStandardsControlAssociations
	hub di sicurezza: V2 ConnectorRegistrations
	hub di sicurezza: CreateActionTarget
	hub di sicurezza: V2 CreateAggregator
	hub di sicurezza: CreateAutomationRule
	hub di sicurezza: CreateAutomationRuleV2
	hub di sicurezza: CreateConfigurationPolicy
	hub di sicurezza: CreateConnector

Prefisso del servizio	Azioni
	hub di sicurezza: V2 CreateConnector
	hub di sicurezza: CreateFindingAggregator
	hub di sicurezza: CreateInsight
	hub di sicurezza: CreateMembers
	hub di sicurezza: V2 CreateTicket
	hub di sicurezza: DeclineInvitations
	hub di sicurezza: DeleteActionTarget
	hub di sicurezza: V2 DeleteAggregator
	hub di sicurezza: DeleteAutomationRuleV2
	hub di sicurezza: DeleteConfigurationPolicy
	hub di sicurezza: DeleteConnector
	hub di sicurezza: V2 DeleteConnector
	hub di sicurezza: DeleteFindingAggregator
	hub di sicurezza: DeleteInsight
	hub di sicurezza: DeleteInvitations
	hub di sicurezza: DeleteMembers
	hub di sicurezza: DescribeActionTargets
	hub di sicurezza: DescribeHub
	hub di sicurezza: DescribeOrganizationConfiguration
	hub di sicurezza: DescribeProducts
	hub di sicurezza: DescribeSecurityHubV2

Prefisso del servizio	Azioni
	hub di sicurezza: DescribeStandards
	hub di sicurezza: DisableImportFindingsForProduct
	hub di sicurezza: DisableOrganizationAdminAccount
	hub di sicurezza: DisableSecurityHub
	hub di sicurezza: DisableSecurityHubFeatureV2
	hub di sicurezza: DisableSecurityHubV2
	hub di sicurezza: DisassociateFromAdministratorAccount
	hub di sicurezza: DisassociateFromMasterAccount
	hub di sicurezza: DisassociateMembers
	hub di sicurezza: EnableImportFindingsForProduct
	hub di sicurezza: EnableOrganizationAdminAccount
	hub di sicurezza: EnableSecurityHub
	hub di sicurezza: EnableSecurityHubFeatureV2
	hub di sicurezza: GetAdministratorAccount
	hub di sicurezza: V2 GetAggregator
	hub di sicurezza: GetAutomationRuleV2
	hub di sicurezza: GetConfigurationPolicy
	hub di sicurezza: GetConfigurationPolicyAssociation
	hub di sicurezza: GetConnector
	hub di sicurezza: V2 GetConnector
	hub di sicurezza: GetCoverageStatisticsV2

Prefisso del servizio	Azioni
	hub di sicurezza: GetEnabledStandards
	hub di sicurezza: GetFindingAggregator
	hub di sicurezza: GetFindingHistory
	hub di sicurezza: GetFindings
	hub di sicurezza: GetInsightResults
	hub di sicurezza: GetInsights
	hub di sicurezza: GetInvitationsCount
	hub di sicurezza: GetMasterAccount
	hub di sicurezza: GetMembers
	hub di sicurezza: GetSecurityControlDefinition
	hub di sicurezza: InviteMembers
	hub di sicurezza: V2 ListAggregators
	hub di sicurezza: ListAutomationRules
	hub di sicurezza: ListAutomationRulesV2
	hub di sicurezza: ListConfigurationPolicies
	hub di sicurezza: ListConfigurationPolicyAssociations
	hub di sicurezza: ListConnectors
	hub di sicurezza: V2 ListConnectors
	hub di sicurezza: ListEnabledProductsForImport
	hub di sicurezza: ListFindingAggregators
	hub di sicurezza: ListInvitations

Prefisso del servizio	Azioni
	hub di sicurezza: ListMembers
	hub di sicurezza: ListOrganizationAdminAccounts
	hub di sicurezza: ListSecurityControlDefinitions
	hub di sicurezza: ListStandardsControlAssociations
	hub di sicurezza: StartConfigurationPolicyAssociation
	hub di sicurezza: StartConfigurationPolicyDisassociation
	hub di sicurezza: UpdateActionTarget
	hub di sicurezza: V2 UpdateAggregator
	hub di sicurezza: UpdateAutomationRuleV2
	hub di sicurezza: UpdateConfigurationPolicy
	hub di sicurezza: UpdateConnector
	hub di sicurezza: V2 UpdateConnector
	hub di sicurezza: UpdateFindingAggregator
	hub di sicurezza: UpdateFindings
	hub di sicurezza: UpdateInsight
	hub di sicurezza: UpdateOrganizationConfiguration
	hub di sicurezza: UpdateSecurityControl
	hub di sicurezza: UpdateSecurityHubConfiguration

Prefisso del servizio	Azioni
securitylake	lago di sicurezza: CreateAwsLogSource lago di sicurezza: CreateCustomLogSource lago di sicurezza: CreateDataLakeExceptionSubscription lago di sicurezza: CreateDataLakeOrganizationConfiguration lago di sicurezza: CreateSubscriber lago di sicurezza: CreateSubscriberNotification lago di sicurezza: DeleteAwsLogSource lago di sicurezza: DeleteCustomLogSource lago di sicurezza: DeleteDataLakeExceptionSubscription lago di sicurezza: DeleteDataLakeOrganizationConfiguration lago di sicurezza: DeleteSubscriber lago di sicurezza: DeleteSubscriberNotification lago di sicurezza: DeregisterDataLakeDelegatedAdministrator lago di sicurezza: GetDataLakeExceptionSubscription lago di sicurezza: GetDataLakeOrganizationConfiguration lago di sicurezza: GetDataLakeSources lago di sicurezza: GetSubscriber lago di sicurezza: ListDataLakes lago di sicurezza: ListLogSources lago di sicurezza: ListSubscribers lago di sicurezza: RegisterDataLakeDelegatedAdministrator

Prefisso del servizio	Azioni
	lago di sicurezza: UpdateDataLakeExceptionSubscription lago di sicurezza: UpdateSubscriber lago di sicurezza: UpdateSubscriberNotification
serverlessrepo	repository senza server: CreateApplication repository senza server: CreateApplicationVersion repository senza server: CreateCloudFormationChangeSet repository senza server: CreateCloudFormationTemplate repository senza server: DeleteApplication repository senza server: GetApplication repository senza server: GetApplicationPolicy repository senza server: GetCloudFormationTemplate repository senza server: ListApplicationDependencies repository senza server: ListApplicationVersions repository senza server: ListApplications repository senza server: PutApplicationPolicy repository senza server: UnshareApplication repository senza server: UpdateApplication

Prefisso del servizio	Azioni
servicecatalog	catalogo dei servizi: AcceptPortfolioShare catalogo dei servizi: AssociateBudgetWithResource catalogo dei servizi: AssociatePrincipalWithPortfolio catalogo dei servizi: AssociateProductWithPortfolio catalogo dei servizi: AssociateServiceActionWithProvisioningArtifact catalogo dei servizi: BatchAssociateServiceActionWithProvisioningArtifact catalogo dei servizi: BatchDisassociateServiceActionFromProvisioningArtifact catalogo dei servizi: CopyProduct catalogo dei servizi: CreateAttributeGroup catalogo dei servizi: CreateConstraint catalogo dei servizi: CreatePortfolio catalogo dei servizi: CreatePortfolioShare catalogo dei servizi: CreateProduct catalogo dei servizi: CreateProvisionedProductPlan catalogo dei servizi: CreateProvisioningArtifact catalogo dei servizi: CreateServiceAction catalogo dei servizi: DeleteAttributeGroup catalogo dei servizi: DeleteConstraint catalogo dei servizi: DeletePortfolio catalogo dei servizi: DeletePortfolioShare

Prefisso del servizio	Azioni
	catalogo dei servizi: DeleteProduct catalogo dei servizi: DeleteProvisionedProductPlan catalogo dei servizi: DeleteProvisioningArtifact catalogo dei servizi: DeleteServiceAction catalogo dei servizi: DescribeConstraint catalogo dei servizi: DescribeCopyProductStatus catalogo dei servizi: DescribePortfolio catalogo dei servizi: DescribePortfolioShareStatus catalogo dei servizi: DescribePortfolioShares catalogo dei servizi: DescribeProduct catalogo dei servizi: DescribeProductAsAdmin catalogo dei servizi: DescribeProductView catalogo dei servizi: DescribeProvisionedProduct catalogo dei servizi: DescribeProvisionedProductPlan catalogo dei servizi: DescribeProvisioningArtifact catalogo dei servizi: DescribeProvisioningParameters catalogo dei servizi: DescribeRecord catalogo dei servizi: DescribeServiceAction catalogo dei servizi: DescribeServiceActionExecutionParameters catalogo dei servizi: DisableAWSOrganizationsAccess catalogo dei servizi: DisassociateBudgetFromResource

Prefisso del servizio	Azioni
	catalogo dei servizi: DisassociatePrincipalFromPortfolio catalogo dei servizi: DisassociateProductFromPortfolio catalogo dei servizi: DisassociateServiceActionFromProvisioningArtifact catalogo dei servizi: EnableAWSOrganizationsAccess catalogo dei servizi: ExecuteProvisionedProductPlan catalogo dei servizi: ExecuteProvisionedProductServiceAction catalogo dei servizi: GetAWSOrganizationsAccessStatus catalogo dei servizi: GetProvisionedProductOutputs catalogo dei servizi: ImportAsProvisionedProduct catalogo dei servizi: ListAcceptedPortfolioShares catalogo dei servizi: ListAttributeGroups catalogo dei servizi: ListBudgetsForResource catalogo dei servizi: ListConstraintsForPortfolio catalogo dei servizi: ListLaunchPaths catalogo dei servizi: ListOrganizationPortfolioAccess catalogo dei servizi: ListPortfolioAccess catalogo dei servizi: ListPortfolios catalogo dei servizi: ListPortfoliosForProduct catalogo dei servizi: ListPrincipalsForPortfolio catalogo dei servizi: ListProvisionedProductPlans

Prefisso del servizio	Azioni
	catalogo dei servizi: ListProvisioningArtifacts catalogo dei servizi: ListProvisioningArtifactsForServiceAction catalogo dei servizi: ListRecordHistory catalogo dei servizi: ListServiceActions catalogo dei servizi: ListServiceActionsForProvisioningArtifact catalogo dei servizi: ListStackInstancesForProvisionedProduct catalogo dei servizi: NotifyProvisionProductEngineWorkflowResult catalogo dei servizi: NotifyTerminateProvisionedProductEngineWorkflowResult catalogo dei servizi: NotifyUpdateProvisionedProductEngineWorkflowResult catalogo dei servizi: ProvisionProduct catalogo dei servizi: RejectPortfolioShare catalogo dei servizi: ScanProvisionedProducts catalogo dei servizi: SearchProducts catalogo dei servizi: SearchProductsAsAdmin catalogo dei servizi: SearchProvisionedProducts catalogo dei servizi: TerminateProvisionedProduct catalogo dei servizi: UpdateConstraint catalogo dei servizi: UpdatePortfolio catalogo dei servizi: UpdatePortfolioShare catalogo dei servizi: UpdateProduct

Prefisso del servizio	Azioni
	catalogo dei servizi: UpdateProvisionedProduct catalogo dei servizi: UpdateProvisionedProductProperties catalogo dei servizi: UpdateProvisioningArtifact catalogo dei servizi: UpdateServiceAction

Prefisso del servizio	Azioni
servicediscovery	individuazione dei servizi: CreateHttpNamespace
	individuazione del servizio: CreatePrivateDnsNamespace
	individuazione del servizio: CreatePublicDnsNamespace
	individuazione del servizio: CreateService
	individuazione del servizio: DeleteNamespace
	individuazione del servizio: DeleteService
	individuazione del servizio: DeleteServiceAttributes
	individuazione del servizio: DeregisterInstance
	individuazione del servizio: GetInstance
	individuazione del servizio: GetInstancesHealthStatus
	individuazione del servizio: GetNamespace
	individuazione del servizio: GetOperation
	individuazione del servizio: GetService
	individuazione del servizio: ListInstances
	individuazione del servizio: ListNamespaces
	individuazione del servizio: ListOperations
	individuazione del servizio: ListServices
	individuazione del servizio: RegisterInstance
	individuazione del servizio: UpdateHttpNamespace
	individuazione del servizio: UpdateInstanceCustomHealthStatus
	individuazione del servizio: UpdatePrivateDnsNamespace

Prefisso del servizio	Azioni
	individuazione del servizio: UpdatePublicDnsNamespace individuazione del servizio: UpdateService individuazione del servizio: UpdateServiceAttributes

Prefisso del servizio	Azioni
servicequotas	quote di servizio: AssociateServiceQuotaTemplate quote di servizio: CreateSupportCase quote di servizio: DeleteServiceQuotaIncreaseRequestFromTemplate quote di servizio: DisassociateServiceQuotaTemplate quote di servizio: GetAWSDefaultServiceQuota quote di servizio: GetAssociationForServiceQuotaTemplate quote di servizio: GetAutoManagementConfiguration quote di servizio: GetQuotaUtilizationReport quote di servizio: GetRequestedServiceQuotaChange quote di servizio: GetServiceQuota quote di servizio: GetServiceQuotaIncreaseRequestFromTemplate quote di servizio: ListAWSDefaultServiceQuotas quote di servizio: ListRequestedServiceQuotaChangeHistory quote di servizio: ListRequestedServiceQuotaChangeHistoryByQuota quote di servizio: ListServiceQuotaIncreaseRequestsInTemplate quote di servizio: ListServiceQuotas quote di servizio: ListServices quote di servizio: PutServiceQuotaIncreaseRequestIntoTemplate quote di servizio: RequestServiceQuotaIncrease quote di servizio: StartAutoManagement

Prefisso del servizio	Azioni
	quote di servizio: StartQuotaUtilizationReport quote di servizio: StopAutoManagement quote di servizio: UpdateAutoManagement

Prefisso del servizio	Azioni
ses	usi: BatchGetMetricData
	usa: CloneReceiptRuleSet
	usa: CreateAddonInstance
	usa: CreateAddonSubscription
	usa: CreateAddressList
	usa: CreateAddressListImportJob
	usa: CreateArchive
	usa: CreateConfigurationSet
	usa: CreateConfigurationSetEventDestination
	usa: CreateConfigurationSetTrackingOptions
	usa: CreateContact
	usa: CreateContactList
	usa: CreateCustomVerificationEmailTemplate
	usa: CreateDedicatedIpPool
	usa: CreateDeliverabilityTestReport
	usa: CreateEmailIdentity
	usa: CreateEmailIdentityPolicy
	usa: CreateEmailTemplate
	usa: CreateImportJob
	usa: CreateIngressPoint
	usa: CreateMultiRegionEndpoint

Prefisso del servizio	Azioni
	usa: CreateReceiptFilter
	usa: CreateReceiptRule
	usa: CreateReceiptRuleSet
	usa: CreateRelay
	usa: CreateRuleSet
	usa: CreateTemplate
	usa: CreateTenant
	usa: CreateTenantResourceAssociation
	usa: CreateTrafficPolicy
	usa: DeleteAddonInstance
	usa: DeleteAddonSubscription
	usa: DeleteAddressList
	usa: DeleteArchive
	usa: DeleteConfigurationSet
	usa: DeleteConfigurationSetEventDestination
	usa: DeleteConfigurationSetTrackingOptions
	usa: DeleteContact
	usa: DeleteContactList
	usa: DeleteCustomVerificationEmailTemplate
	usa: DeleteDedicatedIpPool
	usa: DeleteEmailIdentity

Prefisso del servizio	Azioni
	usa: DeleteEmailIdentityPolicy
	usa: DeleteEmailTemplate
	usa: DeleteIdentity
	usa: DeleteIdentityPolicy
	usa: DeleteIngressPoint
	usa: DeleteMultiRegionEndpoint
	usa: DeleteReceiptFilter
	usa: DeleteReceiptRule
	usa: DeleteReceiptRuleSet
	usa: DeleteRelay
	usa: DeleteRuleSet
	usa: DeleteSuppressedDestination
	usa: DeleteTemplate
	usa: DeleteTenant
	usa: DeleteTenantResourceAssociation
	usa: DeleteTrafficPolicy
	usa: DeleteVerifiedEmailAddress
	usa: DeregisterMemberFromAddressList
	usa: DescribeActiveReceiptRuleSet
	usa: DescribeConfigurationSet
	usa: DescribeReceiptRule

Prefisso del servizio	Azioni
	usa: DescribeReceiptRuleSet
	usa: GetAccount
	usa: GetAccountSendingEnabled
	usa: GetAddonInstance
	usa: GetAddonSubscription
	usa: GetAddressList
	usa: GetArchive
	usa: GetArchiveExport
	usa: GetArchiveMessage
	usa: GetArchiveMessageContent
	usa: GetArchiveSearch
	usa: GetArchiveSearchResults
	usa: GetBlacklistReports
	usa: GetConfigurationSet
	usa: GetConfigurationSetEventDestinations
	usa: GetContact
	usa: GetContactList
	usa: GetCustomVerificationEmailTemplate
	usa: GetDedicatedIp
	usa: GetDedicatedIpPool
	usa: GetDedicatedIps

Prefisso del servizio	Azioni
	usa: GetDeliverabilityDashboardOptions
	usa: GetDeliverabilityTestReport
	usa: GetDomainDeliverabilityCampaign
	usa: GetDomainStatisticsReport
	usa: GetEmailAddressInsights
	usa: GetEmailIdentity
	usa: GetEmailIdentityPolicies
	usa: GetEmailTemplate
	usa: GetIdentityDkimAttributes
	usa: GetIdentityMailFromDomainAttributes
	usa: GetIdentityNotificationAttributes
	usa: GetIdentityPolicies
	usa: GetIdentityVerificationAttributes
	usa: GetImportJob
	usa: GetIngressPoint
	usa: GetMemberOfAddressList
	usa: GetMessageInsights
	usa: GetMultiRegionEndpoint
	usa: GetRelay
	usa: GetRuleSet
	usa: GetSendQuota

Prefisso del servizio	Azioni
	usa: GetSendStatistics
	usa: GetSuppressedDestination
	usa: GetTemplate
	usa: GetTenant
	usa: GetTrafficPolicy
	usa: ListAddonInstances
	usa: ListAddonSubscriptions
	usa: ListAddressListImportJobs
	usa: ListAddressLists
	usa: ListArchiveExports
	usa: ListArchiveSearches
	usa: ListArchives
	usa: ListConfigurationSets
	usa: ListContactLists
	usa: ListContacts
	usa: ListCustomVerificationEmailTemplates
	usa: ListDedicatedIpPools
	usa: ListDeliverabilityTestReports
	usa: ListDomainDeliverabilityCampaigns
	usa: ListEmailIdentities
	usa: ListEmailTemplates

Prefisso del servizio	Azioni
	usa: ListExportJobs
	usa: ListIdentities
	usa: ListIdentityPolicies
	usa: ListImportJobs
	usa: ListIngressPoints
	usa: ListMembersOfAddressList
	usa: ListMultiRegionEndpoints
	usa: ListReceiptFilters
	usa: ListReceiptRuleSets
	usa: ListRecommendations
	usa: ListRelays
	usa: ListReputationEntities
	usa: ListResourceTenants
	usa: ListRuleSets
	usa: ListSuppressedDestinations
	usa: ListTemplates
	usa: ListTenantResources
	usa: ListTenants
	usa: ListTrafficPolicies
	usa: ListVerifiedEmailAddresses
	usa: PutAccountDedicatedIpWarmupAttributes

Prefisso del servizio	Azioni
	usa: PutAccountDetails
	usa: PutAccountSendingAttributes
	usa: PutAccountSuppressionAttributes
	usa: PutAccountVdmAttributes
	usa: PutConfigurationSetArchivingOptions
	usa: PutConfigurationSetDeliveryOptions
	usa: PutConfigurationSetReputationOptions
	usa: PutConfigurationSetSendingOptions
	usa: PutConfigurationSetSuppressionOptions
	usa: PutConfigurationSetTrackingOptions
	usa: PutConfigurationSetVdmOptions
	usa: PutDedicatedIpInPool
	usa: PutDedicatedIpPoolScalingAttributes
	usa: PutDedicatedIpWarmupAttributes
	usa: PutDeliverabilityDashboardOption
	usa: PutEmailIdentityConfigurationSetAttributes
	usa: PutEmailIdentityDkimAttributes
	usa: PutEmailIdentityDkimSigningAttributes
	usa: PutEmailIdentityFeedbackAttributes
	usa: PutEmailIdentityMailFromAttributes
	usa: PutIdentityPolicy

Prefisso del servizio	Azioni
	usa: PutSuppressedDestination
	usa: PutTenantSuppressionAttributes
	usa: RegisterMemberToAddressList
	usa: ReorderReceiptRuleSet
	usa: SendBounce
	usa: SendCustomVerificationEmail
	usa: SetActiveReceiptRuleSet
	usa: SetIdentityDkimEnabled
	usa: SetIdentityFeedbackForwardingEnabled
	usa: SetIdentityHeadersInNotificationsEnabled
	usa: SetIdentityMailFromDomain
	usa: SetIdentityNotificationTopic
	usa: SetReceiptRulePosition
	usa: StartArchiveExport
	usa: StartArchiveSearch
	usa: StopArchiveExport
	usa: StopArchiveSearch
	usa: TestRenderEmailTemplate
	usa: TestRenderTemplate
	usa: UpdateAccountSendingEnabled
	usa: UpdateArchive

Prefisso del servizio	Azioni
	usa: UpdateConfigurationSetEventDestination
	usa: UpdateConfigurationSetReputationMetricsEnabled
	usa: UpdateConfigurationSetSendingEnabled
	usa: UpdateConfigurationSetTrackingOptions
	usa: UpdateContact
	usa: UpdateContactList
	usa: UpdateCustomVerificationEmailTemplate
	usa: UpdateEmailIdentityPolicy
	usa: UpdateEmailTemplate
	usa: UpdateIngressPoint
	usa: UpdateReceiptRule
	usa: UpdateRelay
	usa: UpdateRuleSet
	usa: UpdateTemplate
	usa: UpdateTrafficPolicy
	usa: VerifyDomainDkim
	usa: VerifyDomainIdentity
	usa: VerifyEmailAddress
	usa: VerifyEmailIdentity

Prefisso del servizio	Azioni
shield	scudo: AssociateDRTLogBucket scudo: AssociateHealthCheck scudo: AssociateProactiveEngagementDetails scudo: CreateProtection scudo: CreateProtectionGroup scudo: CreateSubscription scudo: DeleteProtection scudo: DeleteProtectionGroup scudo: DeleteSubscription scudo: DescribeAttack scudo: DescribeAttackStatistics shield:DescribeDRTAccess scudo: DescribeEmergencyContactSettings scudo: DescribeProtection scudo: DescribeProtectionGroup scudo: DescribeSubscription scudo: DisableApplicationLayerAutomaticResponse scudo: DisableProactiveEngagement scudo: DisassociateDRTLogBucket shield:DisassociateDRTRole scudo: DisassociateHealthCheck

Prefisso del servizio	Azioni
	scudo: EnableApplicationLayerAutomaticResponse scudo: EnableProactiveEngagement scudo: GetSubscriptionState scudo: ListAttacks scudo: ListProtectionGroups scudo: ListProtections scudo: ListResourcesInProtectionGroup scudo: UpdateApplicationLayerAutomaticResponse scudo: UpdateEmergencyContactSettings scudo: UpdateProtectionGroup scudo: UpdateSubscription

Prefisso del servizio	Azioni
signer	firmatario: AddProfilePermission
	firmatario: CancelSigningProfile
	firmatario: DescribeSigningJob
	firmatario: GetSigningPlatform
	firmatario: GetSigningProfile
	firmatario: ListProfilePermissions
	firmatario: ListSigningJobs
	firmatario: ListSigningPlatforms
	firmatario: ListSigningProfiles
	firmatario: PutSigningProfile
	firmatario: RemoveProfilePermission
	firmatario: RevokeSignature
	firmatario: RevokeSigningProfile
	firmatario: SignPayload
	firmatario: StartSigningJob

Prefisso del servizio	Azioni
simspaceweaver	simspaceweaver: CreateSnapshot simspaceweaver: DeleteApp simspaceweaver: DeleteSimulation simspaceweaver: DescribeApp simspaceweaver: DescribeSimulation simspaceweaver: ListApps simspaceweaver: ListSimulations simspaceweaver: StartApp simspaceweaver: StartClock simspaceweaver: StartSimulation simspaceweaver: StopApp simspaceweaver: StopClock simspaceweaver: StopSimulation

Prefisso del servizio	Azioni
sms	sms: CreateApp
	sms: CreateReplicationJob
	sms: DeleteApp
	sms: DeleteAppLaunchConfiguration
	sms: DeleteAppReplicationConfiguration
	sms: DeleteAppValidationConfiguration
	sms: DeleteReplicationJob
	sms: DeleteServerCatalog
	sms: DisassociateConnector
	sms: GenerateChangeSet
	sms: GenerateTemplate
	sms: GetApp
	sms: GetAppLaunchConfiguration
	sms: GetAppReplicationConfiguration
	sms: GetAppValidationConfiguration
	sms: GetAppValidationOutput
	sms: GetConnectors
	sms: GetReplicationJobs
	sms: GetReplicationRuns
	sms: GetServers
	sms: ImportAppCatalog

Prefisso del servizio	Azioni
	sms: ImportServerCatalog
	sms: LaunchApp
	sms: ListApps
	sms: NotifyAppValidationOutput
	sms: PutAppLaunchConfiguration
	sms: PutAppReplicationConfiguration
	sms: PutAppValidationConfiguration
	sms: StartAppReplication
	sms: StartOnDemandAppReplication
	sms: StartOnDemandReplicationRun
	sms: StopAppReplication
	sms: TerminateApp
	sms: UpdateApp
	sms: UpdateReplicationJob

Prefisso del servizio	Azioni
sms-voice	sms vocale: AssociateProtectConfiguration messaggio vocale via sms: CreateConfigurationSet messaggio vocale via sms: CreateConfigurationSetEventDestination messaggio vocale via sms: CreateEventDestination messaggio vocale via sms: CreateOptOutList messaggio vocale via sms: CreatePool messaggio vocale via sms: CreateProtectConfiguration messaggio vocale via sms: CreateRegistration messaggio vocale via sms: CreateRegistrationAssociation messaggio vocale via sms: CreateRegistrationAttachment messaggio vocale via sms: CreateRegistrationVersion messaggio vocale via sms: CreateVerifiedDestinationNumber messaggio vocale via sms: DeleteAccountDefaultProtectConfiguration messaggio vocale via sms: DeleteConfigurationSet messaggio vocale via sms: DeleteConfigurationSetEventDestination messaggio vocale via sms: DeleteDefaultMessageType messaggio vocale via sms: DeleteDefaultSenderId messaggio vocale via sms: DeleteEventDestination messaggio vocale via sms: DeleteKeyword

Prefisso del servizio	Azioni
	messaggio vocale via sms: DeleteMediaMessageSpendLimitOverride messaggio vocale via sms: DeleteOptOutList messaggio vocale via sms: DeleteOptedOutNumber messaggio vocale via sms: DeletePool messaggio vocale via sms: DeleteProtectConfiguration messaggio vocale via sms: DeleteProtectConfigurationRuleSetNumberOverride messaggio vocale via sms: DeleteRegistration messaggio vocale via sms: DeleteRegistrationAttachment messaggio vocale via sms: DeleteResourcePolicy messaggio vocale via sms: DeleteTextMessageSpendLimitOverride messaggio vocale via sms: DeleteVerifiedDestinationNumber messaggio vocale via sms: DeleteVoiceMessageSpendLimitOverride messaggio vocale via sms: DescribeAccountAttributes messaggio vocale via sms: DescribeAccountLimits messaggio vocale via sms: DescribeConfigurationSets messaggio vocale via sms: DescribeKeywords messaggio vocale via sms: DescribeOptOutLists messaggio vocale via sms: DescribeOptedOutNumbers messaggio vocale via sms: DescribePhoneNumbers

Prefisso del servizio	Azioni
	messaggio vocale via sms: DescribePools messaggio vocale via sms: DescribeProtectConfigurations messaggio vocale via sms: DescribeRegistrationAttachments messaggio vocale via sms: DescribeRegistrationFieldDefinitions messaggio vocale via sms: DescribeRegistrationFieldValues messaggio vocale via sms: DescribeRegistrationSectionDefinitions messaggio vocale via sms: DescribeRegistrationTypeDefinitions messaggio vocale via sms: DescribeRegistrationVersions messaggio vocale via sms: DescribeRegistrations messaggio vocale via sms: DescribeSenderIds messaggio vocale via sms: DescribeSpendLimits messaggio vocale via sms: DescribeVerifiedDestinationNumbers messaggio vocale via sms: DisassociateOriginationIdentity messaggio vocale via sms: DisassociateProtectConfiguration messaggio vocale via sms: DiscardRegistrationVersion messaggio vocale via sms: GetConfigurationSetEventDestinations messaggio vocale via sms: GetProtectConfigurationCountryRuleSet messaggio vocale via sms: GetResourcePolicy messaggio vocale via sms: ListConfigurationSets messaggio vocale via sms: ListPoolOriginationIdentities

Prefisso del servizio	Azioni
	messaggio vocale via sms: ListProtectConfigurationRuleSetNumberOverrides messaggio vocale via sms: ListRegistrationAssociations messaggio vocale via sms: PutKeyword messaggio vocale via sms: PutOptedOutNumber messaggio vocale via sms: PutProtectConfigurationRuleSetNumberOverride messaggio vocale via sms: PutResourcePolicy messaggio vocale via sms: ReleasePhoneNumber messaggio vocale via sms: ReleaseSenderId messaggio vocale via sms: RequestPhoneNumber messaggio vocale via sms: RequestSenderId messaggio vocale via sms: SendDestinationNumberVerificationCode messaggio vocale via sms: SetAccountDefaultProtectConfiguration messaggio vocale via sms: SetDefaultMessageFeedbackEnabled messaggio vocale via sms: SetDefaultMessageType messaggio vocale via sms: SetDefaultSenderId messaggio vocale via sms: SetMediaMessageSpendLimitOverride messaggio vocale via sms: SetTextMessageSpendLimitOverride messaggio vocale via sms: SetVoiceMessageSpendLimitOverride messaggio vocale via sms: SubmitRegistrationVersion

Prefisso del servizio	Azioni
	messaggio vocale via sms: UpdateConfigurationSetEventDestination messaggio vocale via sms: UpdateEventDestination messaggio vocale via sms: UpdatePhoneNumber messaggio vocale via sms: UpdatePool messaggio vocale via sms: UpdateProtectConfiguration messaggio vocale via sms: UpdateProtectConfigurationCountryRuleSet messaggio vocale via sms: UpdateSenderId

Prefisso del servizio	Azioni
snowball	palla di neve: CancelCluster
	palla di neve: CancelJob
	palla di neve: CreateAddress
	palla di neve: CreateCluster
	palla di neve: CreateJob
	palla di neve: CreateLongTermPricing
	palla di neve: CreateReturnShippingLabel
	palla di neve: DescribeAddress
	palla di neve: DescribeAddresses
	palla di neve: DescribeCluster
	palla di neve: DescribeJob
	palla di neve: DescribeReturnShippingLabel
	palla di neve: GetJobManifest
	palla di neve: GetJobUnlockCode
	palla di neve: GetSnowballUsage
	palla di neve: GetSoftwareUpdates
	palla di neve: ListClusterJobs
	palla di neve: ListClusters
	palla di neve: ListCompatibleImages
	palla di neve: ListJobs
	palla di neve: ListLongTermPricing

Prefisso del servizio	Azioni
	palla di neve: ListPickupLocations palla di neve: ListServiceVersions palla di neve: UpdateCluster palla di neve: UpdateJob palla di neve: UpdateJobShipmentState palla di neve: UpdateLongTermPricing
sqs	seghe: AddPermission seggioloni: CancelMessageMoveTask seggioloni: CreateQueue seggioloni: DeleteQueue seggioloni: PurgeQueue seggioloni: RemovePermission seggioloni: SetQueueAttributes

Prefisso del servizio	Azioni
ssm	ssm: AssociateOpsItemRelatedItem
	ssm: CancelCommand
	ssm: CancelMaintenanceWindowExecution
	ssm: CreateActivation
	ssm: CreateAssociation
	ssm: CreateAssociationBatch
	ssm: CreateDocument
	ssm: CreateMaintenanceWindow
	ssm: CreateOpsItem
	ssm: CreateOpsMetadata
	ssm: CreatePatchBaseline
	ssm: CreateResourceDataSync
	ssm: DeleteActivation
	ssm: DeleteAssociation
	ssm: DeleteCloudConnector
	ssm: DeleteDocument
	ssm: DeleteInventory
	ssm: DeleteMaintenanceWindow
	ssm: DeleteOpsItem
	ssm: DeleteOpsMetadata
	ssm: DeleteParameter

Prefisso del servizio	Azioni
	ssm: DeleteParameters
	ssm: DeletePatchBaseline
	ssm: DeleteResourceDataSync
	ssm: DeleteResourcePolicy
	ssm: DeregisterManagedInstance
	ssm: DeregisterPatchBaselineForPatchGroup
	ssm: DeregisterTargetFromMaintenanceWindow
	ssm: DeregisterTaskFromMaintenanceWindow
	ssm: DescribeActivations
	ssm: DescribeAssociation
	ssm: DescribeAssociationExecutionTargets
	ssm: DescribeAssociationExecutions
	ssm: DescribeAutomationExecutions
	ssm: DescribeAutomationStepExecutions
	ssm: DescribeAvailablePatches
	ssm: DescribeDocument
	ssm: DescribeDocumentParameters
	ssm: DescribeDocumentPermission
	ssm: DescribeEffectiveInstanceAssociations
	ssm: DescribeEffectivePatchesForPatchBaseline
	ssm: DescribeInstanceAssociationsStatus

Prefisso del servizio	Azioni
	ssm: DescribeInstanceInformation
	ssm: DescribeInstancePatchStates
	ssm: DescribeInstancePatchStatesForPatchGroup
	ssm: DescribeInstancePatches
	ssm: DescribeInstanceProperties
	ssm: DescribeInventoryDeletions
	ssm: DescribeMaintenanceWindowExecutionTaskInvocations
	ssm: DescribeMaintenanceWindowExecutionTasks
	ssm: DescribeMaintenanceWindowExecutions
	ssm: DescribeMaintenanceWindowSchedule
	ssm: DescribeMaintenanceWindowTargets
	ssm: DescribeMaintenanceWindowTasks
	ssm: DescribeMaintenanceWindows
	ssm: DescribeMaintenanceWindowsForTarget
	ssm: DescribeOpsItems
	ssm: DescribeParameters
	ssm: DescribePatchBaselines
	ssm: DescribePatchGroupState
	ssm: DescribePatchGroups
	ssm: DescribePatchProperties
	ssm: DescribeSessions

Prefisso del servizio	Azioni
	ssm: DisassociateOpsItemRelatedItem
	ssm: GetAccessToken
	ssm: GetAutomationExecution
	ssm: GetCalendarState
	ssm: GetCloudConnector
	ssm: GetCommandInvocation
	ssm: GetConnectionStatus
	ssm: GetDefaultPatchBaseline
	ssm: GetDeployablePatchSnapshotForInstance
	ssm: GetDocument
	ssm: GetExecutionPreview
	ssm: GetInventory
	ssm: GetInventorySchema
	ssm: GetMaintenanceWindow
	ssm: GetMaintenanceWindowExecution
	ssm: GetMaintenanceWindowExecutionTask
	ssm: GetMaintenanceWindowExecutionTaskInvocation
	ssm: GetMaintenanceWindowTask
	ssm: GetOpsItem
	ssm: GetOpsMetadata
	ssm: GetOpsSummary

Prefisso del servizio	Azioni
	ssm: GetParameter
	ssm: GetParameterHistory
	ssm: GetParameters
	ssm: GetParametersByPath
	ssm: GetPatchBaseline
	ssm: GetPatchBaselineForPatchGroup
	ssm: GetResourcePolicies
	ssm: GetServiceSetting
	ssm: LabelParameterVersion
	ssm: ListAssociationVersions
	ssm: ListAssociations
	ssm: ListCloudConnectors
	ssm: ListCommandInvocations
	ssm: ListCommands
	ssm: ListComplianceItems
	ssm: ListComplianceSummaries
	ssm: ListDocumentMetadataHistory
	ssm: ListDocumentVersions
	ssm: ListDocuments
	ssm: ListInstanceAssociations
	ssm: ListInventoryEntries

Prefisso del servizio	Azioni
	ssm: ListNodes
	ssm: ListNodesSummary
	ssm: ListOpsItemEvents
	ssm: ListOpsItemRelatedItems
	ssm: ListOpsMetadata
	ssm: ListResourceComplianceSummaries
	ssm: ListResourceDataSync
	ssm: ModifyDocumentPermission
	ssm: PutComplianceItems
	ssm: PutInventory
	ssm: PutParameter
	ssm: PutResourcePolicy
	ssm: RegisterDefaultPatchBaseline
	ssm: RegisterManagedInstance
	ssm: RegisterPatchBaselineForPatchGroup
	ssm: RegisterTargetWithMaintenanceWindow
	ssm: RegisterTaskWithMaintenanceWindow
	ssm: ResetServiceSetting
	ssm: ResumeSession
	ssm: SendAutomationSignal
	ssm: SendCommand

Prefisso del servizio	Azioni
	ssm: StartAssociationsOnce
	ssm: StartAutomationExecution
	ssm: StartChangeRequestExecution
	ssm: StartSession
	ssm: StopAutomationExecution
	ssm: TerminateSession
	ssm: UnlabelParameterVersion
	ssm: UpdateAssociation
	ssm: UpdateAssociationStatus
	ssm: UpdateCloudConnector
	ssm: UpdateDocument
	ssm: UpdateDocumentDefaultVersion
	ssm: UpdateDocumentMetadata
	ssm: UpdateInstanceInformation
	ssm: UpdateMaintenanceWindow
	ssm: UpdateMaintenanceWindowTarget
	ssm: UpdateMaintenanceWindowTask
	ssm: UpdateManagedInstanceRole
	ssm: UpdateOpsItem
	ssm: UpdateOpsMetadata
	ssm: UpdatePatchBaseline

Prefisso del servizio	Azioni
	ssm: UpdateResourceDataSync ssm: UpdateServiceSetting ssm: ValidateCloudConnector

Prefisso del servizio	Azioni
ssm-incidents	incidenti ssm: BatchGetIncidentFindings
	incidenti ssm: CreateReplicationSet
	incidenti ssm: CreateResponsePlan
	incidenti ssm: CreateTimelineEvent
	incidenti ssm: DeleteIncidentRecord
	incidenti ssm: DeleteReplicationSet
	incidenti ssm: DeleteResourcePolicy
	incidenti ssm: DeleteResponsePlan
	incidenti ssm: DeleteTimelineEvent
	incidenti ssm: GetIncidentRecord
	incidenti ssm: GetReplicationSet
	incidenti ssm: GetResourcePolicies
	incidenti ssm: GetResponsePlan
	incidenti ssm: GetTimelineEvent
	incidenti ssm: ListIncidentFindings
	incidenti ssm: ListIncidentRecords
	incidenti ssm: ListRelatedItems
	incidenti ssm: ListReplicationSets
	incidenti ssm: ListResponsePlans
	incidenti ssm: ListTimelineEvents
	incidenti ssm: PutResourcePolicy

Prefisso del servizio	Azioni
	incidenti ssm: StartIncident incidenti ssm: UpdateDeletionProtection incidenti ssm: UpdateIncidentRecord incidenti ssm: UpdateRelatedItems incidenti ssm: UpdateReplicationSet incidenti ssm: UpdateResponsePlan incidenti ssm: UpdateTimelineEvent

Prefisso del servizio	Azioni
ssm-sap	ssm-sap: BackupDatabase
	ssm-sap: DeleteResourcePermission
	ssm-sap: DeregisterApplication
	ssm-sap: GetApplication
	ssm-sap: GetComponent
	ssm-sap: GetConfigurationCheckOperation
	ssm-sap: GetDatabase
	ssm-sap: GetOperation
	ssm-sap: GetResourcePermission
	ssm-sap: ListApplications
	ssm-sap: ListComponents
	ssm-sap: ListConfigurationCheckDefinitions
	ssm-sap: ListConfigurationCheckOperations
	ssm-sap: ListDatabases
	ssm-sap: ListOperationEvents
	ssm-sap: ListOperations
	ssm-sap: ListSubCheckResults
	ssm-sap: ListSubCheckRuleResults
	ssm-sap: PutResourcePermission
	ssm-sap: RegisterApplication
	ssm-sap: RestoreDatabase

Prefisso del servizio	Azioni
	ssm-sap: StartApplication ssm-sap: StartApplicationRefresh ssm-sap: StartConfigurationChecks ssm-sap: StopApplication ssm-sap: UpdateApplicationSettings ssm-sap: UpdateHANABackupSettings

Prefisso del servizio	Azioni
states	afferma: CreateActivity stati: CreateStateMachine stati: CreateStateMachineAlias stati: DeleteActivity stati: DeleteStateMachine stati: DeleteStateMachineAlias stati: DeleteStateMachineVersion stati: DescribeActivity stati: DescribeExecution stati: DescribeMapRun stati: DescribeStateMachine stati: DescribeStateMachineAlias stati: DescribeStateMachineForExecution stati: GetExecutionHistory stati: ListActivities stati: ListExecutions stati: ListMapRuns stati: ListStateMachineAliases stati: ListStateMachineVersions stati: ListStateMachines stati: SendTaskFailure

Prefisso del servizio	Azioni
	stati: SendTaskHeartbeat
	stati: SendTaskSuccess
	stati: StartExecution
	stati: StopExecution
	stati: UpdateMapRun
	stati: UpdateStateMachine
	stati: UpdateStateMachineAlias
	stati: ValidateStateMachineDefinition
sts	set: AssumeRole
	set: AssumeRoleWithSAML
	set: AssumeRoleWithWebIdentity
	set: DecodeAuthorizationMessage
	set: GetAccessKeyInfo
	set: GetCallerIdentity
	set: GetFederationToken
	set: GetSessionToken
	set: GetWebIdentityToken

Prefisso del servizio	Azioni
swf	swf: DeleteActivityType file swf: DeleteWorkflowType file swf: DeprecateActivityType file swf: DeprecateDomain file swf: DeprecateWorkflowType file swf: DescribeActivityType file swf: DescribeDomain file swf: DescribeWorkflowType file swf: ListActivityTypes file swf: ListDomains file swf: ListWorkflowTypes file swf: RegisterActivityType file swf: RegisterDomain file swf: RegisterWorkflowType file swf: UndeprecateActivityType file swf: UndeprecateDomain file swf: UndeprecateWorkflowType

Prefisso del servizio	Azioni
synthetics	sintetici: AssociateResource
	sintetici: CreateCanary
	sintetici: CreateGroup
	sintetici: DeleteCanary
	sintetici: DeleteGroup
	sintetici: DescribeCanaries
	sintetici: DescribeCanariesLastRun
	sintetici: DescribeRuntimeVersions
	sintetici: DisassociateResource
	sintetici: GetCanary
	sintetici: GetCanaryRuns
	sintetici: GetGroup
	sintetici: ListAssociatedGroups
	sintetici: ListGroupResources
	sintetici: ListGroups
	sintetici: StartCanary
	sintetici: StartCanaryDryRun
	sintetici: StopCanary
	sintetici: UpdateCanary

Prefisso del servizio	Azioni
tag	etichetta: DescribeReportCreation etichetta: GetComplianceSummary etichetta: GetResources etichetta: StartReportCreation

Prefisso del servizio	Azioni
textract	estratto: AnalyzeDocument
	estratto: AnalyzeExpense
	textract:AnalyzeID
	estratto: CreateAdapter
	estratto: CreateAdapterVersion
	estratto: DeleteAdapter
	estratto: DeleteAdapterVersion
	estratto: DetectDocumentText
	estratto: GetAdapter
	estratto: GetAdapterVersion
	estratto: GetDocumentAnalysis
	estratto: GetDocumentTextDetection
	estratto: GetExpenseAnalysis
	estratto: GetLendingAnalysis
	estratto: GetLendingAnalysisSummary
	estratto: ListAdapterVersions
	estratto: ListAdapters
	estratto: StartDocumentAnalysis
	estratto: StartDocumentTextDetection
	estratto: StartExpenseAnalysis
	estratto: StartLendingAnalysis

Prefisso del servizio	Azioni
	estratto: UpdateAdapter

Prefisso del servizio	Azioni
timestream	flusso temporale: CancelQuery flusso temporale: CreateDatabase flusso temporale: CreateScheduledQuery flusso temporale: CreateTable flusso temporale: DeleteDatabase flusso temporale: DeleteScheduledQuery flusso temporale: DeleteTable flusso temporale: DescribeAccountSettings flusso temporale: DescribeDatabase flusso temporale: DescribeScheduledQuery flusso temporale: DescribeTable flusso temporale: ExecuteScheduledQuery flusso temporale: ListBatchLoadTasks flusso temporale: ListDatabases flusso temporale: ListScheduledQueries flusso temporale: ListTables flusso temporale: PrepareQuery flusso temporale: UpdateAccountSettings flusso temporale: UpdateDatabase flusso temporale: UpdateScheduledQuery flusso temporale: UpdateTable

Prefisso del servizio	Azioni
tnb	tb: CancelSolNetworkOperation
	tnb: CreateSolFunctionPackage
	tnb: CreateSolNetworkInstance
	tnb: CreateSolNetworkPackage
	tnb: DeleteSolFunctionPackage
	tnb: DeleteSolNetworkInstance
	tnb: DeleteSolNetworkPackage
	tnb: GetSolFunctionInstance
	tnb: GetSolFunctionPackage
	tnb: GetSolFunctionPackageContent
	tnb: GetSolFunctionPackageDescriptor
	tnb: GetSolNetworkInstance
	tnb: GetSolNetworkOperation
	tnb: GetSolNetworkPackage
	tnb: GetSolNetworkPackageContent
	tnb: GetSolNetworkPackageDescriptor
	tnb: InstantiateSolNetworkInstance
	tnb: ListSolFunctionInstances
	tnb: ListSolFunctionPackages
	tnb: ListSolNetworkInstances
	tnb: ListSolNetworkOperations

Prefisso del servizio	Azioni
	tnb: ListSolNetworkPackages tnb: PutSolFunctionPackageContent tnb: PutSolNetworkPackageContent tnb: TerminateSolNetworkInstance tnb: UpdateSolFunctionPackage tnb: UpdateSolNetworkInstance tnb: UpdateSolNetworkPackage tnb: ValidateSolFunctionPackageContent tnb: ValidateSolNetworkPackageContent

Prefisso del servizio	Azioni
transcribe	trascrivere: CreateCallAnalyticsCategory trascrivere: CreateLanguageModel trascrivere: CreateMedicalVocabulary trascrivere: CreateVocabulary trascrivere: CreateVocabularyFilter trascrivere: DeleteCallAnalyticsCategory trascrivere: DeleteCallAnalyticsJob trascrivere: DeleteLanguageModel trascrivere: DeleteMedicalScribeJob trascrivere: DeleteMedicalTranscriptionJob trascrivere: DeleteMedicalVocabulary trascrivere: DeleteTranscriptionJob trascrivere: DeleteVocabulary trascrivere: DeleteVocabularyFilter trascrivere: DescribeLanguageModel trascrivere: GetCallAnalyticsCategory trascrivere: GetCallAnalyticsJob trascrivere: GetMedicalScribeJob trascrivere: GetMedicalTranscriptionJob trascrivere: GetMedicalVocabulary trascrivere: GetTranscriptionJob

Prefisso del servizio	Azioni
	trascrivere: GetVocabulary trascrivere: GetVocabularyFilter trascrivere: ListCallAnalyticsCategories trascrivere: ListCallAnalyticsJobs trascrivere: ListLanguageModels trascrivere: ListMedicalScribeJobs trascrivere: ListMedicalTranscriptionJobs trascrivere: ListMedicalVocabularies trascrivere: ListTranscriptionJobs trascrivere: ListVocabularies trascrivere: ListVocabularyFilters trascrivere: StartCallAnalyticsJob trascrivere: StartCallAnalyticsStreamTranscription trascrivere: StartCallAnalyticsStreamTranscriptionWebSocket trascrivere: StartMedicalScribeJob trascrivere: StartMedicalStreamTranscription trascrivere: StartMedicalStreamTranscriptionWebSocket trascrivere: StartMedicalTranscriptionJob trascrivere: StartStreamTranscription trascrivere: StartStreamTranscriptionWebSocket trascrivere: StartTranscriptionJob

Prefisso del servizio	Azioni
	trascrivere: UpdateCallAnalyticsCategory trascrivere: UpdateMedicalVocabulary trascrivere: UpdateVocabulary trascrivere: UpdateVocabularyFilter

Prefisso del servizio	Azioni
transfer	trasferimento: CreateAccess
	trasferimento: CreateAgreement
	trasferimento: CreateConnector
	trasferimento: CreateProfile
	trasferimento: CreateServer
	trasferimento: CreateUser
	trasferimento: CreateWebApp
	trasferimento: CreateWorkflow
	trasferimento: DeleteAccess
	trasferimento: DeleteAgreement
	trasferimento: DeleteCertificate
	trasferimento: DeleteConnector
	trasferimento: DeleteHostKey
	trasferimento: DeleteProfile
	trasferimento: DeleteServer
	trasferimento: DeleteSshPublicKey
	trasferimento: DeleteUser
	trasferimento: DeleteWebApp
	trasferimento: DeleteWebAppCustomization
	trasferimento: DeleteWorkflow
	trasferimento: DescribeAccess

Prefisso del servizio	Azioni
	trasferimento: DescribeAgreement
	trasferimento: DescribeCertificate
	trasferimento: DescribeConnector
	trasferimento: DescribeExecution
	trasferimento: DescribeHostKey
	trasferimento: DescribeProfile
	trasferimento: DescribeSecurityPolicy
	trasferimento: DescribeServer
	trasferimento: DescribeUser
	trasferimento: DescribeWebApp
	trasferimento: DescribeWebAppCustomization
	trasferimento: DescribeWorkflow
	trasferimento: ImportCertificate
	trasferimento: ImportHostKey
	trasferimento: ImportSshPublicKey
	trasferimento: ListAccesses
	trasferimento: ListCertificates
	trasferimento: ListConnectors
	trasferimento: ListExecutions
	trasferimento: ListFileTransferResults
	trasferimento: ListHostKeys

Prefisso del servizio	Azioni
	trasferimento: ListProfiles
	trasferimento: ListSecurityPolicies
	trasferimento: ListServers
	trasferimento: ListUsers
	trasferimento: ListWebApps
	trasferimento: ListWorkflows
	trasferimento: SendWorkflowStepState
	trasferimento: StartDirectoryListing
	trasferimento: StartFileTransfer
	trasferimento: StartRemoteDelete
	trasferimento: StartRemoteMove
	trasferimento: StartServer
	trasferimento: StopServer
	trasferimento: TestConnection
	trasferimento: TestIdentityProvider
	trasferimento: UpdateAccess
	trasferimento: UpdateAgreement
	trasferimento: UpdateCertificate
	trasferimento: UpdateConnector
	trasferimento: UpdateHostKey
	trasferimento: UpdateProfile

Prefisso del servizio	Azioni
	trasferimento: UpdateServer trasferimento: UpdateUser trasferimento: UpdateWebApp trasferimento: UpdateWebAppCustomization
translate	tradurre: CreateParallelData tradurre: DeleteParallelData tradurre: DeleteTerminology tradurre: DescribeTextTranslationJob tradurre: GetParallelData tradurre: GetTerminology tradurre: ImportTerminology tradurre: ListLanguages tradurre: ListParallelData tradurre: ListTerminologies tradurre: ListTextTranslationJobs tradurre: StartTextTranslationJob tradurre: StopTextTranslationJob tradurre: TranslateDocument tradurre: TranslateText tradurre: UpdateParallelData

Prefisso del servizio	Azioni
voiceid	voiceid: AssociateFraudster
	identificatore vocale: CreateDomain
	identificatore vocale: CreateWatchlist
	identificatore vocale: DeleteDomain
	identificatore vocale: DeleteFraudster
	identificatore vocale: DeleteSpeaker
	identificatore vocale: DeleteWatchlist
	identificatore vocale: DescribeDomain
	identificatore vocale: DescribeFraudster
	identificatore vocale: DescribeFraudsterRegistrationJob
	identificatore vocale: DescribeSpeaker
	identificatore vocale: DescribeSpeakerEnrollmentJob
	identificatore vocale: DescribeWatchlist
	identificatore vocale: DisassociateFraudster
	identificatore vocale: EvaluateSession
	identificatore vocale: ListDomains
	identificatore vocale: ListFraudsterRegistrationJobs
	identificatore vocale: ListFraudsters
	identificatore vocale: ListSpeakerEnrollmentJobs
	identificatore vocale: ListSpeakers
	identificatore vocale: ListWatchlists

Prefisso del servizio	Azioni
	identificatore vocale: OptOutSpeaker identificatore vocale: StartFraudsterRegistrationJob identificatore vocale: StartSpeakerEnrollmentJob identificatore vocale: UpdateDomain identificatore vocale: UpdateWatchlist

Prefisso del servizio	Azioni
vpc-lattice	reticolo vpc: CreateAccessLogSubscription reticolo vpc: CreateListener reticolo vpc: CreateResourceConfiguration reticolo vpc: CreateResourceGateway reticolo vpc: CreateRule reticolo vpc: CreateService reticolo vpc: CreateServiceNetwork reticolo vpc: CreateServiceNetworkResourceAssociation reticolo vpc: CreateServiceNetworkServiceAssociation reticolo vpc: CreateServiceNetworkVpcAssociation reticolo vpc: CreateTargetGroup reticolo vpc: DeleteAccessLogSubscription reticolo vpc: DeleteAuthPolicy reticolo vpc: DeleteDomainVerification reticolo vpc: DeleteListener reticolo vpc: DeleteResourceConfiguration reticolo vpc: DeleteResourceEndpointAssociation reticolo vpc: DeleteResourceGateway reticolo vpc: DeleteResourcePolicy reticolo vpc: DeleteRule reticolo vpc: DeleteService

Prefisso del servizio	Azioni
	reticolo vpc: DeleteServiceNetwork reticolo vpc: DeleteServiceNetworkResourceAssociation reticolo vpc: DeleteServiceNetworkServiceAssociation reticolo vpc: DeleteServiceNetworkVpcAssociation reticolo vpc: DeleteTargetGroup reticolo vpc: DeregisterTargets reticolo vpc: GetAccessLogSubscription reticolo vpc: GetAuthPolicy reticolo vpc: GetDomainVerification reticolo vpc: GetListener reticolo vpc: GetResourceConfiguration reticolo vpc: GetResourceGateway reticolo vpc: GetResourcePolicy reticolo vpc: GetRule reticolo vpc: GetService reticolo vpc: GetServiceNetwork reticolo vpc: GetServiceNetworkResourceAssociation reticolo vpc: GetServiceNetworkServiceAssociation reticolo vpc: GetServiceNetworkVpcAssociation reticolo vpc: GetTargetGroup reticolo vpc: ListAccessLogSubscriptions

Prefisso del servizio	Azioni
	reticolo vpc: ListDomainVerifications reticolo vpc: ListListeners reticolo vpc: ListResourceConfigurations reticolo vpc: ListResourceEndpointAssociations reticolo vpc: ListResourceGateways reticolo vpc: ListRules reticolo vpc: ListServiceNetworkResourceAssociations reticolo vpc: ListServiceNetworkServiceAssociations reticolo vpc: ListServiceNetworkVpcAssociations reticolo vpc: ListServiceNetworkVpcEndpointAssociations reticolo vpc: ListServiceNetworks reticolo vpc: ListServices reticolo vpc: ListTargetGroups reticolo vpc: ListTargets reticolo vpc: PutAuthPolicy reticolo vpc: PutResourcePolicy reticolo vpc: RegisterTargets reticolo vpc: StartDomainVerification reticolo vpc: UpdateAccessLogSubscription reticolo vpc: UpdateListener reticolo vpc: UpdateResourceConfiguration

Prefisso del servizio	Azioni
	reticolo vpc: UpdateResourceGateway reticolo vpc: UpdateRule reticolo vpc: UpdateService reticolo vpc: UpdateServiceNetwork reticolo vpc: UpdateServiceNetworkVpcAssociation reticolo vpc: UpdateTargetGroup

Prefisso del servizio	Azioni
wafv2	wafv2: ACL AssociateWeb
	wafv2: CheckCapacity
	wafv2:CreateAPIKey
	wafv2:CreateIPSet
	wafv2: CreateRegexPatternSet
	wafv2: CreateRuleGroup
	wafv2: ACL CreateWeb
	wafv2:DeleteAPIKey
	wafv2: DeleteFirewallManagerRuleGroups
	wafv2:DeleteIPSet
	wafv2: DeleteLoggingConfiguration
	wafv2: DeletePermissionPolicy
	wafv2: DeleteRegexPatternSet
	wafv2: DeleteRuleGroup
	wafv2: ACL DeleteWeb
	wafv2: DescribeAllManagedProducts
	wafv2: DescribeManagedProductsByVendor
	wafv2: DescribeManagedRuleGroup
	wafv2: ACL DisassociateWeb
	wafv2: GenerateMobileSdkReleaseUrl
	wafv2: GetDecryptedAPIKey

Prefisso del servizio	Azioni
	wafv2:GetIPSet
	wafv2: GetLoggingConfiguration
	wafv2: GetManagedRuleSet
	wafv2: GetMobileSdkRelease
	wafv2: GetRateBasedStatementManagedKeys
	wafv2: GetRegexPatternSet
	wafv2: GetRevenueStatistics
	wafv2: GetRevenueStatisticsSummary
	wafv2: GetRevenueStatisticsTimeSeries
	wafv2: GetRuleGroup
	wafv2: GetSampledRequests
	wafv2: GetWebACLForResource
	wafv2:ListAPIKeys
	wafv2: ListAvailableManagedRuleGroupVersions
	wafv2: ListAvailableManagedRuleGroups
	wafv2:ListIPSets
	wafv2: ListLoggingConfigurations
	wafv2: ListManagedRuleSets
	wafv2: ListMobileSdkReleases
	wafv2: ListRegexPatternSets
	wafv2: ListResourcesForWebACL

Prefisso del servizio	Azioni
	wafv2: ListRuleGroups
	wafv2: ListSettlementRecords
	wafv2: ListWebACLs
	wafv2: PutLoggingConfiguration
	wafv2: PutManagedRuleSetVersions
	wafv2:UpdateIPSet
	wafv2: UpdateManagedRuleSetVersionExpiryDate
	wafv2: UpdateRegexPatternSet
	wafv2: UpdateRuleGroup
	wafv2: ACL UpdateWeb

Prefisso del servizio	Azioni
wellarchitected	ben architettato: AssociateLenses ben architettato: AssociateProfiles ben architettato: CreateLensShare ben architettato: CreateLensVersion ben architettato: CreateMilestone ben architettato: CreateProfile ben architettato: CreateProfileShare ben architettato: CreateReviewTemplate ben architettato: CreateWorkload ben architettato: CreateWorkloadShare ben architettato: DeleteLens ben architettato: DeleteLensShare ben architettato: DeleteProfile ben architettato: DeleteProfileShare ben architettato: DeleteReviewTemplate ben architettato: DeleteTemplateShare ben architettato: DeleteWorkload ben architettato: DeleteWorkloadShare ben architettato: DisassociateLenses ben architettato: DisassociateProfiles ben architettato: ExportLens

Prefisso del servizio	Azioni
	ben architettato: GetAnswer ben architettato: GetConsolidatedReport ben architettato: GetGlobalSettings ben architettato: GetLens ben architettato: GetLensReview ben architettato: GetLensReviewReport ben architettato: GetLensVersionDifference ben architettato: GetMilestone ben architettato: GetProfile ben architettato: GetProfileTemplate ben architettato: GetReviewTemplate ben architettato: GetReviewTemplateAnswer ben architettato: GetReviewTemplateLensReview ben architettato: GetWorkload ben architettato: ImportLens ben architettato: ListAnswers ben architettato: ListCheckDetails ben architettato: ListCheckSummaries ben architettato: ListLensReviewImprovements ben architettato: ListLensReviews ben architettato: ListLensShares

Prefisso del servizio	Azioni
	ben architettato: ListLenses ben architettato: ListMilestones ben architettato: ListNotifications ben architettato: ListProfileNotifications ben architettato: ListProfileShares ben architettato: ListProfiles ben architettato: ListReviewTemplateAnswers ben architettato: ListReviewTemplates ben architettato: ListShareInvitations ben architettato: ListTemplateShares ben architettato: ListWorkloadShares ben architettato: ListWorkloads ben architettato: UpdateAnswer ben architettato: UpdateGlobalSettings ben architettato: UpdateIntegration ben architettato: UpdateLensReview ben architettato: UpdateProfile ben architettato: UpdateReviewTemplate ben architettato: UpdateReviewTemplateLensReview ben architettato: UpdateShareInvitation ben architettato: UpdateWorkload

Prefisso del servizio	Azioni
	ben architettato: UpdateWorkloadShare ben architettato: UpgradeLensReview ben architettato: UpgradeProfileVersion ben architettato: UpgradeReviewTemplateLensReview

Prefisso del servizio	Azioni
wisdom	saggezza: CreateAssistant
	saggezza: CreateAssistantAssociation
	saggezza: CreateContent
	saggezza: CreateKnowledgeBase
	saggezza: CreateQuickResponse
	saggezza: CreateSession
	saggezza: DeleteAssistant
	saggezza: DeleteAssistantAssociation
	saggezza: DeleteContent
	saggezza: DeleteImportJob
	saggezza: DeleteKnowledgeBase
	saggezza: DeleteQuickResponse
	saggezza: GetAssistant
	saggezza: GetAssistantAssociation
	saggezza: GetContent
	saggezza: GetContentAssociation
	saggezza: GetContentSummary
	saggezza: GetImportJob
	saggezza: GetKnowledgeBase
	saggezza: GetRecommendations
	saggezza: GetSession

Prefisso del servizio	Azioni
	saggezza: ListAssistantAssociations
	saggezza: ListAssistants
	saggezza: ListContentAssociations
	saggezza: ListContents
	saggezza: ListImportJobs
	saggezza: ListKnowledgeBases
	saggezza: ListQuickResponses
	saggezza: NotifyRecommendationsReceived
	saggezza: QueryAssistant
	saggezza: RemoveKnowledgeBaseTemplateUri
	saggezza: SearchContent
	saggezza: SearchQuickResponses
	saggezza: SearchSessions
	saggezza: StartContentUpload
	saggezza: StartImportJob
	saggezza: UpdateContent
	saggezza: UpdateKnowledgeBaseTemplateUri
	saggezza: UpdateQuickResponse
	saggezza: UpdateSession

Prefisso del servizio	Azioni
worklink	collegamento di lavoro: AssociateDomain collegamento di lavoro: AssociateWebsiteAuthorizationProvider collegamento di lavoro: AssociateWebsiteCertificateAuthority collegamento di lavoro: CreateFleet collegamento di lavoro: DeleteFleet collegamento di lavoro: DescribeAuditStreamConfiguration collegamento di lavoro: DescribeCompanyNetworkConfiguration collegamento di lavoro: DescribeDevice collegamento di lavoro: DescribeDevicePolicyConfiguration collegamento di lavoro: DescribeDomain collegamento di lavoro: DescribeFleetMetadata collegamento di lavoro: DescribeIdentityProviderConfiguration collegamento di lavoro: DescribeWebsiteCertificateAuthority collegamento di lavoro: DisassociateDomain collegamento di lavoro: DisassociateWebsiteAuthorizationProvider collegamento di lavoro: DisassociateWebsiteCertificateAuthority collegamento di lavoro: ListDevices collegamento di lavoro: ListDomains collegamento di lavoro: ListFleets collegamento di lavoro: ListWebsiteAuthorizationProviders collegamento di lavoro: ListWebsiteCertificateAuthorities

Prefisso del servizio	Azioni
	collegamento di lavoro: RestoreDomainAccess collegamento di lavoro: RevokeDomainAccess collegamento di lavoro: SignOutUser collegamento di lavoro: UpdateAuditStreamConfiguration collegamento di lavoro: UpdateCompanyNetworkConfiguration collegamento di lavoro: UpdateDevicePolicyConfiguration collegamento di lavoro: UpdateDomainMetadata collegamento di lavoro: UpdateFleetMetadata collegamento di lavoro: UpdateIdentityProviderConfiguration

Prefisso del servizio	Azioni
workspace	spazi di lavoro: AcceptAccountLinkInvitation
	spazi di lavoro: AssociateConnectionAlias
	spazi di lavoro: AssociateIpGroups
	spazi di lavoro: AssociateWorkspaceApplication
	spazi di lavoro: CopyWorkspaceImage
	spazi di lavoro: CreateAccountLinkInvitation
	spazi di lavoro: CreateConnectClientAddIn
	spazi di lavoro: CreateConnectionAlias
	spazi di lavoro: CreateIpGroup
	spazi di lavoro: CreateStandbyWorkspaces
	spazi di lavoro: CreateUpdatedWorkspaceImage
	spazi di lavoro: CreateWorkspaceBundle
	spazi di lavoro: CreateWorkspaceImage
	spazi di lavoro: CreateWorkspaces
	spazi di lavoro: CreateWorkspacesPool
	spazi di lavoro: DeleteAccountLinkInvitation
	spazi di lavoro: DeleteClientBranding
	spazi di lavoro: DeleteConnectClientAddIn
	spazi di lavoro: DeleteConnectionAlias
	spazi di lavoro: DeleteIpGroup
	spazi di lavoro: DeleteWorkspaceBundle

Prefisso del servizio	Azioni
	spazi di lavoro: DeleteWorkspaceImage
	spazi di lavoro: DeployWorkspaceApplications
	spazi di lavoro: DeregisterWorkspaceDirectory
	spazi di lavoro: DescribeAccount
	spazi di lavoro: DescribeAccountModifications
	spazi di lavoro: DescribeApplicationAssociations
	spazi di lavoro: DescribeApplications
	spazi di lavoro: DescribeBundleAssociations
	spazi di lavoro: DescribeClientBranding
	spazi di lavoro: DescribeClientProperties
	spazi di lavoro: DescribeConnectClientAddIns
	spazi di lavoro: DescribeConnectionAliasPermissions
	spazi di lavoro: DescribeConnectionAliases
	spazi di lavoro: DescribeCustomWorkspaceImageImport
	spazi di lavoro: DescribeImageAssociations
	spazi di lavoro: DescribeIpGroups
	spazi di lavoro: DescribeWorkspaceAssociations
	spazi di lavoro: DescribeWorkspaceBundles
	spazi di lavoro: DescribeWorkspaceDirectories
	spazi di lavoro: DescribeWorkspaceImagePermissions
	spazi di lavoro: DescribeWorkspaceSnapshots

Prefisso del servizio	Azioni
	spazi di lavoro: DescribeWorkspaces
	spazi di lavoro: DescribeWorkspacesConnectionStatus
	spazi di lavoro: DescribeWorkspacesPoolSessions
	spazi di lavoro: DescribeWorkspacesPools
	spazi di lavoro: DisassociateConnectionAlias
	spazi di lavoro: DisassociateIpGroups
	spazi di lavoro: DisassociateWorkspaceApplication
	spazi di lavoro: GetAccountLink
	spazi di lavoro: ImportClientBranding
	spazi di lavoro: ImportWorkspaceImage
	spazi di lavoro: ListAccountLinks
	spazi di lavoro: ListAvailableManagementCidrRanges
	spazi di lavoro: MigrateWorkspace
	spazi di lavoro: ModifyAccount
	spazi di lavoro: ModifyCertificateBasedAuthProperties
	spazi di lavoro: ModifyClientProperties
	spazi di lavoro: ModifyEndpointEncryptionMode
	spazi di lavoro: ModifySamlProperties
	spazi di lavoro: ModifySelfservicePermissions
	spazi di lavoro: ModifyStreamingProperties
	spazi di lavoro: ModifyWorkspaceAccessProperties

Prefisso del servizio	Azioni
	spazi di lavoro: ModifyWorkspaceCreationProperties
	spazi di lavoro: ModifyWorkspaceProperties
	spazi di lavoro: ModifyWorkspaceState
	spazi di lavoro: RebootWorkspaces
	spazi di lavoro: RebuildWorkspaces
	spazi di lavoro: RegisterWorkspaceDirectory
	spazi di lavoro: RejectAccountLinkInvitation
	spazi di lavoro: RestoreWorkspace
	spazi di lavoro: StartWorkspaces
	spazi di lavoro: StartWorkspacesPool
	spazi di lavoro: StopWorkspaces
	spazi di lavoro: StopWorkspacesPool
	spazi di lavoro: TerminateWorkspaces
	spazi di lavoro: TerminateWorkspacesPool
	spazi di lavoro: TerminateWorkspacesPoolSession
	spazi di lavoro: UpdateConnectClientAddIn
	spazi di lavoro: UpdateConnectionAliasPermission
	spazi di lavoro: UpdateWorkspaceBundle
	spazi di lavoro: UpdateWorkspaceImagePermission
	spazi di lavoro: UpdateWorkspacesPool

Prefisso del servizio	Azioni
xray	radiografia: CreateGroup radiografia: CreateSamplingRule radiografia: DeleteGroup radiografia: DeleteResourcePolicy radiografia: DeleteSamplingRule radiografia: GetEncryptionConfig radiografia: GetGroup radiografia: GetGroups radiografia: GetInsight radiografia: GetInsightEvents radiografia: GetInsightImpactGraph radiografia: GetInsightSummaries radiografia: GetSamplingRules radiografia: ListResourcePolicies radiografia: PutEncryptionConfig radiografia: PutResourcePolicy radiografia: UpdateGroup radiografia: UpdateSamplingRule

Quote Sistema di analisi degli accessi AWS IAM

Sistema di analisi degli accessi AWS IAM ha le seguenti quote:

Risorsa	Quota predefinita	Quota massima
Numero massimo di analizzatori a livello di account per tipo di analizzatore dell' Account AWS per regione	1	1
Numero massimo di analizzatori di accesso esterni o non utilizzati a livello di organizzazione per tipo di analizzatore per regione Account AWS	5	20[1]
Numero massimo di analizzatori degli accessi interni a livello di organizzazione per organizzazione AWS e Regione	1	1
Numero massimo di regole di archiviazione per analizzatore	100 Ogni regola di archivio può avere fino a 20 valori per criterio.	1.000[1]
Numero massimo di anteprime di accesso per analizzatore all'ora	1.000	1.000
AWS CloudTrail file di registro elaborati per generazioni di policy	100.000	100.000
Generazioni simultanee di policy	1	1

Risorsa	Quota predefinita	Quota massima
Dimensione dei AWS CloudTrail dati di generazione delle politiche	25 GB	25 GB
AWS CloudTrail Intervallo di tempo di generazione delle politiche	90 giorni	90 giorni
Generazioni di policy al giorno	Africa (Città del Capo): 5 Asia Pacifico (Hong Kong): 5 Asia Pacifico (Giacarta): 5 Europa (Milano): 5 Medio Oriente (Bahrein): 5 Tutte le altre regioni supportate: 50 **Note** Le richieste di generazione delle policy annullate si applicano alla quota giornaliera.	Africa (Città del Capo): 5 Asia Pacifico (Hong Kong): 5 Asia Pacifico (Giacarta): 5 Europa (Milano): 5 Medio Oriente (Bahrein): 5 Tutte le altre regioni supportate: 50

[1]Alcune quote possono essere configurate dal cliente tramite Service Quotas.

Risoluzione dei problemi relativi a IAM

Utilizza le informazioni qui riportate per eseguire la diagnosi e risolvere problemi comuni durante l'utilizzo di AWS Identity and Access Management (IAM).

Problemi

- Non riesco ad accedere al mio AWS account
- Chiavi di accesso smarrite
- Variabili della policy non funzionanti
- Le modifiche che apporto non sono sempre immediatamente visibili
- Non sono autorizzato a eseguire: iam: DeleteVirtualMFADevice
- Come posso creare utenti IAM in modo sicuro?
- Risorse aggiuntive
- Risoluzione dei problemi relativi ai messaggi di errore di accesso rifiutato
- Risolvi i problemi relativi ai messaggi di errore relativi all'accesso negato con ID di autorizzazione (anteprima)
- Risoluzione dei problemi con l'utente root
- Risoluzione dei problemi relativi alle policy IAM
- Risoluzione dei problemi relativi alle passkey e alle chiavi di sicurezza FIDO
- Risoluzione dei problemi relativi ai ruoli IAM
- Risoluzione dei problemi relativi a IAM e Amazon EC2
- Risoluzione dei problemi relativi a IAM ed Amazon S3
- Risoluzione dei problemi di federazione SAML con IAM

Non riesco ad accedere al mio AWS account

Verifica di disporre delle credenziali corrette e di utilizzare il metodo corretto per accedere. Per ulteriori informazioni, consulta Risoluzione dei problemi di accesso nella Guida per l'utente di Accedi ad AWS .

Chiavi di accesso smarrite

Le chiavi di accesso sono costituite da due parti:

- Identificatore della chiave di accesso: Questo non è un segreto e può essere visualizzato nella console IAM ovunque le chiavi di accesso siano elencate, ad esempio nella pagina di riepilogo dell'utente.
- Chiave di accesso segreta: questa informazione viene fornita quando si crea inizialmente la coppia di chiavi di accesso. Proprio come una password, non può essere recuperata in seguito. Se la chiave di accesso segreta viene persa, è necessario creare una nuova coppia di chiavi di accesso. Se si disponi già del numero massimo di chiavi di accesso, è necessario eliminare una coppia esistente prima di crearne un'altra.

Se perdi la chiave di accesso segreta, è necessario eliminarla e crearne una nuova. Per ulteriori istruzioni, consulta Aggiornare le chiavi di accesso.

Variabili della policy non funzionanti

Se le variabili della policy non funzionano, si è verificato uno dei seguenti errori:

La data è errata nell'elemento della policy Version.

> Verificare che tutte le policy che includono variabili includano il seguente numero di versione nella policy: `"Version": "2012-10-17"`. Senza il numero di versione corretto, le variabili non vengono sostituite durante la valutazione. Al contrario, le variabili vengono valutate letteralmente. Le policy che non includono variabili continueranno a funzionare se si include il numero di versione più recente.
>
> Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Una versione della policy viene creata quando si modifica una policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version. Per ulteriori informazioni sulle versioni di policy, consultare the section called "Controllo delle versioni delle policy IAM".

I caratteri variabili sono scritti con lettere maiuscole e minuscole errate.

> Verificare che le variabili della policy applichino la distinzione maiuscole/minuscole corretta. Per informazioni dettagliate, vedi Elementi delle policy IAM: variabili e tag.

Le modifiche che apporto non sono sempre immediatamente visibili

Essendo un servizio a cui si accede da computer in data center presenti in tutto il mondo, IAM utilizza un modello di elaborazione distribuito denominato consistenza finale. Qualsiasi modifica apportata in IAM (o in altri AWS servizi), inclusi i tag ABAC (Attribute-based Access Control), richiede tempo per diventare visibile da tutti gli endpoint possibili. Alcuni dei ritardi sono dovuti al tempo necessario per inviare i dati da un server a un altro, da una zona di replica a un'altra e da una regione a un'altra. IAM utilizza inoltre la memorizzazione nella cache per migliorare le prestazioni, è possibile che ciò aumenti ulteriormente il tempo richiesto, in quanto la modifica potrebbe risultare visibile solo dopo il timeout dei dati memorizzati nella cache.

È necessario progettare le applicazioni globali in modo da considerare questi potenziali ritardi e assicurarsi che funzionino come previsto, anche quando una modifica apportata in una posizione non è immediatamente visibile in un'altra. Tali modifiche includono la creazione o l'aggiornamento di utenti, gruppi, ruoli, o policy. Si consiglia di non includere tali modifiche IAM nei percorsi critici e ad alta disponibilità del codice dell'applicazione. Al contrario, apporta modifiche IAM in un'inizializzazione separata o in una routine di configurazione che si esegue meno frequentemente. Inoltre, assicurarsi di verificare che le modifiche siano state propagate prima che i flussi di lavoro di produzione dipendano da esse.

Per ulteriori informazioni su come alcuni altri AWS servizi ne sono influenzati, consulta le seguenti risorse:

- Amazon DynamoDB: Consistenza di lettura nella Guida per gli sviluppatori di DynamoDB e Consistenza di lettura nella Guida per gli sviluppatori di Amazon DynamoDB.
- Amazon EC2: Consistenza finale di EC2 nella Documentazione di riferimento dell'API Amazon EC2.
- Amazon EMR: Garantire la consistenza quando si utilizzano Amazon S3 e Amazon EMR per flussi di lavoro ETL nel blog dei big data AWS
- Amazon Redshift: Gestione della consistenza dei dati nella Guida per gli sviluppatori di Amazon Redshift Database
- Amazon S3: Modello di consistenza dei dati di Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service

Non sono autorizzato a eseguire: iam: DeleteVirtualMFADevice

Quando si tenta di assegnare o rimuovere un dispositivo MFA virtuale per sé stessi o altri, è possibile che venga visualizzato il seguente errore:

```
User: arn:aws:iam::123456789012:user/Diego is not authorized to
 perform: iam:DeleteVirtualMFADevice on resource: arn:aws:iam::123456789012:mfa/Diego
 with an explicit deny
```

Ciò può accadere se qualcuno in precedenza ha iniziato ad assegnare un dispositivo MFA virtuale a un utente nella console IAM e poi ha annullato il processo. Questa operazione crea un dispositivo MFA virtuale per l'utente in IAM, ma non lo assegna mai all'utente. Eliminare il dispositivo MFA virtuale esistente prima di poter creare un nuovo dispositivo MFA virtuale con lo stesso nome del dispositivo.

Per risolvere questo problema, un amministratore non dovrebbe modificare le policy di autorizzazioni. L'amministratore deve invece utilizzare l' AWS API AWS CLI o per eliminare il dispositivo MFA virtuale esistente ma non assegnato.

Per eliminare un dispositivo MFA virtuale esistente, ma non assegnato

1. Visualizzare i dispositivi MFA virtuali nel proprio account.
   - AWS CLI: `aws iam list-virtual-mfa-devices`
   - AWS API: `ListVirtualMFADevices`
2. Nella risposta, individuare l'ARN del dispositivo MFA virtuale dell'utente per il quale si sta tentando di correggere l'anomalia.
3. Eliminare il dispositivo MFA virtuale.
   - AWS CLI: `aws iam delete-virtual-mfa-device`
   - AWS API: `DeleteVirtualMFADevice`

Come posso creare utenti IAM in modo sicuro?

Se hai dipendenti che richiedono l'accesso a AWS, puoi scegliere di creare utenti IAM o utilizzare IAM Identity Center per l'autenticazione. Se utilizzi IAM, ti AWS consiglia di creare un utente IAM e di comunicare in modo sicuro le credenziali al dipendente. Se non ci si trova fisicamente accanto

al dipendente, si consiglia di utilizzare un flusso di lavoro sicuro per comunicare le credenziali ai dipendenti.

Utilizza il seguente flusso di lavoro sicuro per creare un nuovo utente in IAM:

1. Crea un nuovo utente utilizzando la Console di gestione AWS. Scegli di concedere Console di gestione AWS l'accesso con una password generata. Se necessario, seleziona la casella di controllo accanto a L'utente deve creare una nuova password all'accesso successivo. Non aggiungere una policy di autorizzazione all'utente fino a quando non ha cambiato la password.
2. Dopo avere aggiunto l'utente, copia l'URL di accesso, il nome utente e la password per il nuovo utente. Per visualizzare la password, scegli Mostra.
3. Invia la password al tuo dipendente utilizzando un metodo di comunicazione sicuro della tua azienda, ad esempio e-mail, chat o un sistema di ticket. Separatamente, fornisci agli utenti il collegamento alla console utente IAM e il relativo nome utente. Chiedi al dipendente di confermare che riesce ad accedere correttamente prima di concedergli le autorizzazioni.
4. Dopo che il dipendente ha confermato, aggiungi le autorizzazioni necessarie. Come buona prassi di sicurezza, aggiungi una policy che richiede all'utente di autenticarsi utilizzando la MFA per gestire le proprie credenziali. Per un esempio di policy, consulta AWS: consente agli utenti IAM autenticati con MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Risorse aggiuntive

Le seguenti risorse possono aiutarti a risolvere i problemi mentre lavori con. AWS

- AWS CloudTrail Guida per l'utente : consente AWS CloudTrail di tenere traccia di una cronologia delle chiamate API effettuate AWS e di archiviare tali informazioni nei file di registro. Ciò consente di determinare quali utenti e account hanno effettuato l'accesso alle risorse nell'account, quando sono state effettuate le chiamate, quali operazioni sono state richieste e altro ancora. Per ulteriori informazioni, consulta Registrazione di IAM e AWS STS Chiamate API con AWS CloudTrail.
- AWS Knowledge Center : trova le domande frequenti e i link ad altre risorse per aiutarti a risolvere i problemi.
- AWS Centro assistenza : richiedi supporto tecnico.
- AWS Premium Support Center : ottieni supporto tecnico premium.

Risoluzione dei problemi relativi ai messaggi di errore di accesso rifiutato

Le seguenti informazioni possono aiutarti a identificare, diagnosticare e risolvere gli errori di accesso negato con AWS Identity and Access Management. Gli errori di accesso negato vengono visualizzati quando si nega AWS esplicitamente o implicitamente una richiesta di autorizzazione.

- Un rifiuto esplicito si verifica quando una policy contiene una `Deny` dichiarazione per l'azione specifica. AWS
- Un diniego implicito si verifica quando non è presente un'istruzione `Deny` applicabile e non è presente neppure un'istruzione `Allow` applicabile. Dato che una policy IAM nega un principale IAM per impostazione predefinita, la policy deve consentire esplicitamente al principale di eseguire un'operazione. In caso contrario, la policy nega implicitamente l'accesso. Per ulteriori informazioni, consulta Differenza tra rifiuto esplicito e implicito.

Quando si effettua una richiesta a un servizio o a una risorsa, alla richiesta possono essere applicate più policy. Rivedi tutte le policy applicabili oltre a quella specificata nel messaggio di errore.

- Se più policy dello stesso tipo negano una richiesta, il messaggio di errore di accesso negato non specifica il numero di policy valutate.
- Se più tipi di policy rifiutano una richiesta di autorizzazione, AWS include solo uno di questi tipi di policy nel messaggio di errore.

> **Important**
>
> Hai problemi ad accedere a AWS? Assicurati di essere nella pagina di accesso AWS corretta per il tuo tipo di utente. Se sei il Utente root dell'account AWS (proprietario dell'account), puoi accedere AWS utilizzando le credenziali che hai impostato quando hai creato il Account AWS. Se sei un utente IAM, l'amministratore dell'account può fornirti le credenziali di accesso per AWS . Se hai bisogno di supporto, non utilizzare il link Feedback in questa pagina. Il modulo viene ricevuto dal team di AWS documentazione, non Supporto. Invece, nella pagina Contattaci scegli Ancora impossibile accedere al tuo AWS account e quindi scegli una delle opzioni di supporto disponibili.

Ricevo la dicitura «accesso negato» quando faccio una richiesta a un AWS service

- Verifica se il messaggio di errore include il tipo e il nome Amazon Resource Name (ARN) della politica responsabile del rifiuto dell'accesso. In tal caso, verifica la presenza di dichiarazioni di rifiuto relative all'azione prevista dalla policy specificata. Se viene fornito il tipo di policy ma non è presente alcun ARN di policy, concentratevi sulla risoluzione dei problemi relativi a quel tipo di policy: controllate le dichiarazioni di rifiuto relative all'azione nelle policy del tipo specificato. Se il messaggio di errore non riporta il tipo di policy responsabile del rifiuto dell'accesso, utilizza le altre linee guida in questa sezione per risolvere i problemi.
- Verificare di disporre dell'autorizzazione della policy basata su identità necessaria per chiamare l'operazione e le risorse richieste. Se sono impostate delle condizioni, è necessario soddisfare anche tali condizioni quando si invia la richiesta. Per informazioni sulla visualizzazione o la modifica delle policy IAM per un utente, gruppo o ruolo, consulta Gestire le policy IAM.
- Se Console di gestione AWS restituisce un messaggio che indica che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore ti ha fornito le credenziali di accesso o il link di accesso.

  L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `widgets:`*`GetWidget`* fittizie.

  ```
  User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
   widgets:GetWidget on resource: my-example-widget
  ```

  In questo caso, Mateo deve richiedere al suo amministratore di aggiornare le sue policy per poter accedere alla risorsa *`my-example-widget`* utilizzando l'operazione `widgets:`*`GetWidget`*.
- Stai cercando di accedere a un servizio che supportapolicy basate sulle risorse, ad esempio Amazon S3, Amazon SNS o Amazon SQS? In tal caso, verificare che la policy specifichi l'utente come principale capitale e conceda l'accesso. Se si effettua una richiesta a un servizio all'interno dell'account, le policy basate su identità o le policy basate su risorse possono concedere l'autorizzazione. Se si effettua una richiesta a un servizio in un altro account, sia le policy basate su identità che le policy basate su risorse devono concedere l'autorizzazione. Per scoprire quali servizi supportano le policy basate su risorse, consultare la pagina AWS servizi supportati da IAM.
- Se la policy include una condizione con una coppia chiave-valore, esaminala con attenzione. Gli esempi includono la chiave di condizione `aws:RequestTag/`*`tag-key`* globale AWS KMS

`kms:EncryptionContext:encryption_context_key`, la e la chiave di `ResourceTag/tag-key` condizione supportate da più servizi. Verifica che il nome della chiave non corrisponda a più risultati. Poiché i nomi delle chiavi di distinzioni non fanno distinzione tra maiuscole e minuscole, una condizione che verifica la presenza di una chiave denominata `foo` corrisponde a `foo`, `Foo` o `F00`. Se la richiesta include più coppie chiave-valore con nomi di chiavi che cambiano solo la dimensione dei caratteri, l'accesso potrebbe essere inaspettatamente negato. Per ulteriori informazioni, consulta Elementi della policy JSON IAM: Condizione.

- Se è presente un limite delle autorizzazioni, è necessario verificare che la policy utilizzata per il limite delle autorizzazioni consenta la richiesta. Se le policy basate su identità consentono la richiesta, ma il limite delle autorizzazioni non la consente, la richiesta viene rifiutata. Un limite di autorizzazioni controlla le autorizzazioni massime che può avere un responsabile IAM (utente o ruolo). Resource-based le politiche non sono limitate dai limiti delle autorizzazioni. I limiti delle autorizzazioni non sono comuni. Per ulteriori informazioni su come AWS valuta le politiche, vedere. Logica di valutazione delle policy
- Se stai firmando manualmente richieste API (senza utilizzare gli SDK AWS), verifica di aver firmato correttamente la richiesta.
- Se utilizzi una policy per gli endpoint VPC Amazon e ricevi un errore di accesso negato quando non hai effettuato l'accesso a AWS CloudTrail, è possibile che l'account del proprietario dell'endpoint VPC sia diverso dall'account che ha effettuato la chiamata o dall'account del ruolo di destinazione.

Messaggio di accesso rifiutato quando si effettua una richiesta con credenziali di sicurezza temporanee

- Innanzitutto, occorre verificare che l'accesso non venga negato per un motivo non legato alle credenziali temporanee. Per ulteriori informazioni, consulta Ricevo la dicitura «accesso negato» quando faccio una richiesta a un AWS service.
- Verificare che il servizio accetti le credenziali di sicurezza temporanee, consultare AWS servizi supportati da IAM.
- Verifica che le tue richieste vengano firmate correttamente e che il formato della richiesta sia valido. Per ulteriori informazioni, consulta la documentazione del kit di strumenti o Usa credenziali temporanee con AWS risorse.
- Verifica che le credenziali di sicurezza provvisorie non siano scadute. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM.

- Verifica che l'utente o il ruolo IAM dispongano delle autorizzazioni corrette. Le autorizzazioni per le credenziali di sicurezza temporanee sono derivate da un utente o ruolo IAM. Di conseguenza, le autorizzazioni sono limitate a quelle che vengono concesse al ruolo di cui hai assunto le credenziali temporanee. Per Scopri come vengono determinate le autorizzazioni per le credenziali di sicurezza temporanee, consultare Autorizzazioni per le credenziali di sicurezza temporanee.
- Se hai assunto un ruolo, la sessione del ruolo potrebbe essere limitata da policy di sessione. Quando richiedi credenziali di sicurezza temporanee utilizzando il codice AWS STS, puoi facoltativamente passare policy di sessione in linea o gestite. ??? Le policy di sessione sono policy avanzate che vengono passate come parametro durante la creazione di una sessione temporanea per un ruolo a livello di programmazione. Puoi passare un singolo documento della policy di sessione inline JSON utilizzando il parametro `Policy`. Puoi utilizzare il parametro `PolicyArns` per specificare fino a 10 policy di sessione gestite. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. In alternativa, se l'amministratore o un programma personalizzato fornisce le credenziali temporanee, potrebbero includere policy di sessione per limitare l'accesso.
- Se sei un utente principale AWS STS federato, la tua sessione potrebbe essere limitata dai criteri di sessione. Puoi creare una sessione utente federata accedendo AWS come utente IAM e quindi richiedendo un token di federazione. Per ulteriori informazioni, consulta Richiesta di credenziali tramite un gestore di identità personalizzato. Se l'utente o un gestore identità ha passato policy di sessione durante la richiesta di un token di federazione, la sessione è limitata da quelle policy. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate sull'identità dell'utente IAM e delle policy di sessione. Per ulteriori informazioni sulle policy di sessione, consulta Policy di sessione.
- Se si accede a una risorsa che dispone di una policy basata sulle risorse tramite un ruolo, verificare che la policy conceda le autorizzazioni per il ruolo. Ad esempio, la policy seguente permette `MyRole` dell'account `111122223333` per l'accesso a `amzn-s3-demo-bucket`.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Sid": "S3BucketPolicy",
    "Effect": "Allow",
    "Principal": {"AWS": ["arn:aws:iam::111122223333:role/MyRole"]},
    "Action": ["s3:PutObject"],
    "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/*"]
  }]
```

```
}
```

Esempi di messaggi di errore di accesso negato

Per la maggior parte, i messaggi di errore di accesso negato sono visualizzati nel formato `User` *`user`* `is not authorized to perform` *`action`* `on` *`resource`* `because` *`context`*. In questo esempio, *`user`* è l'ARN del principale a cui viene negato l'accesso, *`action`* è l'azione di servizio negata dalla policy ed *`resource`* è l'ARN della risorsa su cui la policy agisce. Il *`context`* campo fornisce un contesto aggiuntivo sul tipo di policy che ha negato l'accesso. In alcuni casi, contiene anche l'ARN della politica che ha negato l'accesso.

Quando una policy nega esplicitamente l'accesso perché contiene una `Deny` dichiarazione, AWS include la frase `with an explicit deny in a` *`type`* `policy` nel messaggio di errore di accesso negato. Questa frase può anche specificare l'ARN della politica, come segue:. `with an explicit deny in a` *`type`* `policy:` *`policy ARN`*

Quando la politica nega implicitamente l'accesso, AWS include la frase `because no` *`type`* `policy allows the` *`action`* `action` nel messaggio di errore di accesso negato.

> **Note**
>
> Alcuni AWS servizi non supportano questo formato di messaggio di errore di accesso negato. Il contenuto dei messaggi di errore di accesso negato può variare a seconda del servizio che effettua la richiesta di autorizzazione.

Gli esempi seguenti mostrano il formato di vari tipi di messaggi di errore di accesso negato.

Accesso negato a causa di una policy di controllo dei servizi: diniego implicito

1. Verifica un'istruzione `Allow` mancante relativa all'operazione nelle tue policy di controllo dei servizi. Per l'esempio seguente, l'operazione è `codecommit:ListRepositories`.
2. Aggiorna la tua SCP aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS Organizations .

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 codecommit:ListRepositories
```

```
because no service control policy allows the codecommit:ListRepositories action
```

Accesso negato a causa di una policy di controllo dei servizi: diniego esplicito

1. Se nel messaggio di errore viene fornito un ARN della policy, verifica la presenza di una `Deny` dichiarazione relativa all'azione nella policy di controllo del servizio (SCP) specificata. Nell'esempio seguente, l'azione è. `codecommit:ListRepositories`
2. Se nel messaggio di errore non viene fornito alcun ARN della policy, verificate la presenza di una `Deny` dichiarazione relativa all'azione nei vostri SCP.
3. Aggiorna la tua SCP rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta Update a service control policy (SCP) nella Guida per l'AWS Organizations utente.

Messaggio di errore con un ARN della policy:

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 codecommit:ListRepositories
with an explicit deny in a service control policy:
 arn:aws:organizations::777788889999:policy/o-exampleorgid/service_control_policy/p-
examplepolicyid123
```

Messaggio di errore senza un ARN della policy:

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 codecommit:ListRepositories
with an explicit deny in a service control policy
```

Accesso negato a causa di una policy di controllo delle risorse - rifiuto esplicito

1. Se nel messaggio di errore viene fornito un ARN della policy, verifica la presenza di una `Deny` dichiarazione per l'azione nella policy di controllo delle risorse (RCP) specificata. Nell'esempio seguente, l'azione è. `secretsmanager:GetSecretValue`
2. Se nel messaggio di errore non viene fornito alcun ARN della policy, verificate la presenza di una `Deny` dichiarazione relativa all'azione negli RCP.
3. Aggiorna la tua RCP rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta Update a resource control policy (RCP) nella Guida per l'utente. AWS Organizations

Messaggio di errore con un ARN della policy:

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 secretsmanager:GetSecretValue
on resource: arn:aws:secretsmanager:us-east-1:123456789012:secret:*
with an explicit deny in a resource control policy:
 arn:aws:organizations::777788889999:policy/o-exampleorgid/resource_control_policy/p-
examplepolicyid456
```

Messaggio di errore senza un ARN della policy:

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 secretsmanager:GetSecretValue
on resource: arn:aws:secretsmanager:us-east-1:123456789012:secret:*
with an explicit deny in a resource control policy
```

Accesso negato a causa di una policy dell'endpoint VPC: diniego implicito

1. Verifica l'assenza di un'istruzione `Allow` per l'azione nelle tue policy dell'endpoint del cloud privato virtuale (VPC). Per l'esempio seguente, l'operazione è `codecommit:ListRepositories`.
2. Aggiorna la tua policy sugli endpoint VPC aggiungendo l'iistruzione `Allow`. Per ulteriori informazioni, consulta Aggiornamento di una policy dell'endpoint VPC nella AWS PrivateLink Guida.

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 codecommit:ListRepositories
because no VPC endpoint policy allows the codecommit:ListRepositories action
```

Accesso negato a causa di una policy dell'endpoint VPC: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` esplicita relativa all'azione nelle tue policy dell'endpoint del cloud privato virtuale (VPC). Per l'esempio seguente, l'operazione è `codedeploy:ListDeployments`.
2. Aggiorna la tua policy sugli endpoint VPC rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta Aggiornamento di una policy dell'endpoint VPC nella AWS PrivateLink Guida.

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 codedeploy:ListDeployments
on resource: arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:*
```

```
with an explicit deny in a VPC endpoint policy
```

Accesso negato a causa di limiti delle autorizzazioni: diniego implicito

1. Verifica la presenza di un'istruzione `Allow` mancante relativa all'azione nel limite delle autorizzazioni. Per l'esempio seguente, l'operazione è `codedeploy:ListDeployments`.
2. Aggiorna il limite delle autorizzazioni aggiungendo l'istruzione `Allow`relativa alla tua policy IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM e Modificare le policy IAM.

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 codedeploy:ListDeployments
on resource: arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:*
because no permissions boundary allows the codedeploy:ListDeployments action
```

Accesso negato a causa di un limite delle autorizzazioni: diniego esplicito

1. Se nel messaggio di errore viene fornito un ARN della policy, verifica la presenza di una `Deny` dichiarazione per l'azione nel limite delle autorizzazioni specificato. Nell'esempio seguente, l'azione è. `sagemaker:ListModels`
2. Se nel messaggio di errore non viene fornito alcun ARN della policy, verifica la presenza di un'`Deny`istruzione per l'azione nel limite delle autorizzazioni associato al principale.
3. Aggiorna il limite delle autorizzazioni rimuovendo l'istruzione `Deny`relativa alla tua policy IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM e Modificare le policy IAM.

Messaggio di errore con un ARN della policy:

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 sagemaker:ListModels
with an explicit deny in a permissions boundary: arn:aws:iam::123456789012:policy/
DeveloperPermissionBoundary
```

Messaggio di errore senza un ARN della policy:

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 sagemaker:ListModels
```

```
with an explicit deny in a permissions boundary
```

Accesso negato a causa di policy di sessione: diniego implicito

1. Verifica la presenza di un'istruzione `Allow` mancante relativa all'azione nelle tue policy di sessione. Per l'esempio seguente, l'operazione è `codecommit:ListRepositories`.
2. Aggiorna la tua policy di sessione aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta la sezione Policy di sessione e Modificare le policy IAM.

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 codecommit:ListRepositories
because no session policy allows the codecommit:ListRepositories action
```

Accesso negato a causa di policy di sessione: diniego esplicito

1. Se nel messaggio di errore viene fornito un ARN della policy, verifica la presenza di una `Deny` dichiarazione per l'azione nella policy di sessione specificata. Nell'esempio seguente, l'azione è`codedeploy:ListDeployments`.
2. Se nel messaggio di errore non viene fornito alcun ARN della policy, verificate la presenza di una `Deny` dichiarazione per l'azione nelle policy di sessione.
3. Aggiorna la tua policy di sessione rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta la sezione Policy di sessione e Modificare le policy IAM.

Messaggio di errore con un ARN della policy:

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 codedeploy:ListDeployments
on resource: arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:*
with an explicit deny in a session policy: arn:aws:iam::123456789012:policy/
DeveloperSessionPolicy
```

Messaggio di errore senza un ARN della policy:

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 codedeploy:ListDeployments
on resource: arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:*
with an explicit deny in a session policy
```

Accesso negato a causa di policy basate sulle risorse: diniego implicito

1. Verifica la presenza di un'istruzione `Allow` mancante relativa all'azione nella tua policy basata sulle risorse. Per l'esempio seguente, l'operazione è `secretsmanager:GetSecretValue`.
2. Aggiorna la tua policy aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta Resource-based le politiche eModificare le policy IAM.

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 secretsmanager:GetSecretValue
because no resource-based policy allows the secretsmanager:GetSecretValue action
```

Accesso negato a causa di policy basate sulle risorse: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` esplicita relativa all'azione nella tua policy basata sulle risorse. Per l'esempio seguente, l'operazione è `secretsmanager:GetSecretValue`.
2. Aggiorna la tua policy rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta Resource-based le politiche eModificare le policy IAM.

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform:
 secretsmanager:GetSecretValue
on resource: arn:aws:secretsmanager:us-east-1:123456789012:secret:*
with an explicit deny in a resource-based policy
```

Accesso negato a causa di policy di attendibilità dei ruoli: diniego implicito

1. Verifica la presenza di un'istruzione `Allow` mancante relativa all'azione nella tua policy di attendibilità dei ruoli. Per l'esempio seguente, l'operazione è `sts:AssumeRole`.
2. Aggiorna la tua policy aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta Resource-based le politiche eModificare le policy IAM.

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform: sts:AssumeRole
because no role trust policy allows the sts:AssumeRole action
```

Accesso negato a causa di policy di attendibilità dei ruoli: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` esplicita relativa all'azione nella tua policy di attendibilità dei ruoli. Per l'esempio seguente, l'operazione è `sts:AssumeRole`.
2. Aggiorna la tua policy rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta Resource-based le politiche e Modificare le policy IAM.

```
User: arn:aws:iam::123456789012:user/John is not authorized to perform: sts:AssumeRole
with an explicit deny in the role trust policy
```

Accesso negato a causa di policy basate sull'identità: diniego implicito

1. Verifica l'eventuale mancanza di un'istruzione `Allow` per l'azione nelle politicy basate sull'identità collegate all'identità. Nel seguente esempio, l'azione `codecommit:ListRepositories` è collegata al ruolo HR.
2. Aggiorna la tua policy aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta Identity-based le politiche e Modificare le policy IAM.

```
User: arn:aws:iam::123456789012:role/HR is not authorized to perform:
 codecommit:ListRepositories
because no identity-based policy allows the codecommit:ListRepositories action
```

Accesso negato a causa di policy basate sull'identità: diniego esplicito

1. Se nel messaggio di errore viene fornito un ARN della policy, verifica la presenza di una `Deny` dichiarazione per l'azione nella policy specificata. Nell'esempio seguente, l'azione è`codedeploy:ListDeployments`.
2. Se nel messaggio di errore non viene fornito alcun ARN della policy, cerca una `Deny` dichiarazione per l'azione nelle policy basate sull'identità allegate all'identità.
3. Aggiorna la tua policy rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, vedere policy e. Identity-based Modificare le policy IAM

Messaggio di errore con un ARN della policy:

```
User: arn:aws:iam::123456789012:role/HR is not authorized to perform:
 codedeploy:ListDeployments
on resource: arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:*
with an explicit deny in an identity-based policy: arn:aws:iam::123456789012:policy/
HRAccessPolicy
```

Messaggio di errore senza un ARN della policy:

```
User: arn:aws:iam::123456789012:role/HR is not authorized to perform:
 codedeploy:ListDeployments
on resource: arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:*
with an explicit deny in an identity-based policy
```

Risolvi i problemi relativi ai messaggi di errore relativi all'accesso negato con ID di autorizzazione (anteprima)

> **Important**
>
> Access Troubleshooter è disponibile in anteprima pubblica.

Panoramica di

Quando si AWS rifiuta una richiesta per un'API supportata, la risposta di errore di accesso negato di solito include un ID di autorizzazione. Questo ID fa riferimento alla valutazione dell'autorizzazione per quella richiesta negata. Per recuperare i dettagli della valutazione e identificare la causa del rifiuto, effettua una richiesta all'`GetRequestAuthorizationDetails`API e fornisci l'ID di autorizzazione oppure apri il link presente nel messaggio di errore. Ciò consente di identificare la causa di una richiesta rifiutata in una singola chiamata, anziché risolvere i rifiuti uno alla volta.

> **Note**
>
> AWS non garantisce un ID di autorizzazione per ogni richiesta rifiutata. Vedi Quando non ricevi un ID di autorizzazione in questo documento.

L'esempio seguente mostra un messaggio di errore di accesso negato che include un ARN della policy, un collegamento alla console di AWS gestione e un ID di autorizzazione:

```
User: arn:aws:iam::123456789012:user/janedoe is not authorized to perform:
 iam:CreateInstanceProfile
on resource: arn:aws:iam::123456789012:instance-profile/my-instance-profile
with an explicit deny in an identity-based policy: arn:aws:iam::123456789012:policy/
ExampleDenyPolicy.
Go to https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/
authorization-details/EXAMPLE1a2b3c4d5e6f7g8h9i0j for complete details,
or call the GetRequestAuthorizationDetails API with the following authorization id:
 EXAMPLE1a2b3c4d5e6f7g8h9i0j
```

Concetti e definizioni

- Dettagli dell'autorizzazione. I dettagli dell'autorizzazione includono il contesto della richiesta, le valutazioni eseguite e le politiche che sono state valutate.
- ID di autorizzazione. Un identificatore univoco incluso in una risposta di errore di accesso negato supportata. Fa riferimento alla valutazione dell'autorizzazione per quella richiesta negata ed è il valore passato al link della AWS Management Console o all'`GetRequestAuthorizationDetails`API.
- Contesto della richiesta. I valori delle chiavi di contesto della condizione valutate per la richiesta. Sono incluse le chiavi di contesto delle condizioni globali e le chiavi di contesto delle condizioni specifiche del servizio.
- Valutazione. Per ogni action/resource coppia nella richiesta, le chiavi di contesto della condizione applicate e il risultato valutato:`ALLOW`, `EXPLICIT_DENY` (una politica ha esplicitamente negato l'azione) o `IMPLICIT_DENY` (nessuna politica ha consentito l'azione) e un riferimento a ciascuna politica e dichiarazione che corrisponde alla action/resource coppia. Ogni riferimento è un ARN della politica (per le politiche gestite) o un URI (per le politiche in linea e altre politiche non gestite) che viene riutilizzato in tutta la risposta.
- Politiche valutate durante l'autorizzazione. Le politiche AWS prese in considerazione per la richiesta, comprese le politiche che non corrispondono alla action/resource coppia. Ogni voce indica il tipo di politica (ad esempio, basata sull'identità, basata sulle risorse, SCP, RCP, limite delle autorizzazioni o politica di sessione) e lo stesso ARN (per le politiche gestite) o URI utilizzato nelle valutazioni (per le politiche in linea e altre politiche non gestite). Utilizza l'ARN o l'URI per far corrispondere una policy alla valutazione a cui fa riferimento. Il campo «Allegato a» mostra l'ARN principale del principale a cui è collegata la politica valutata.

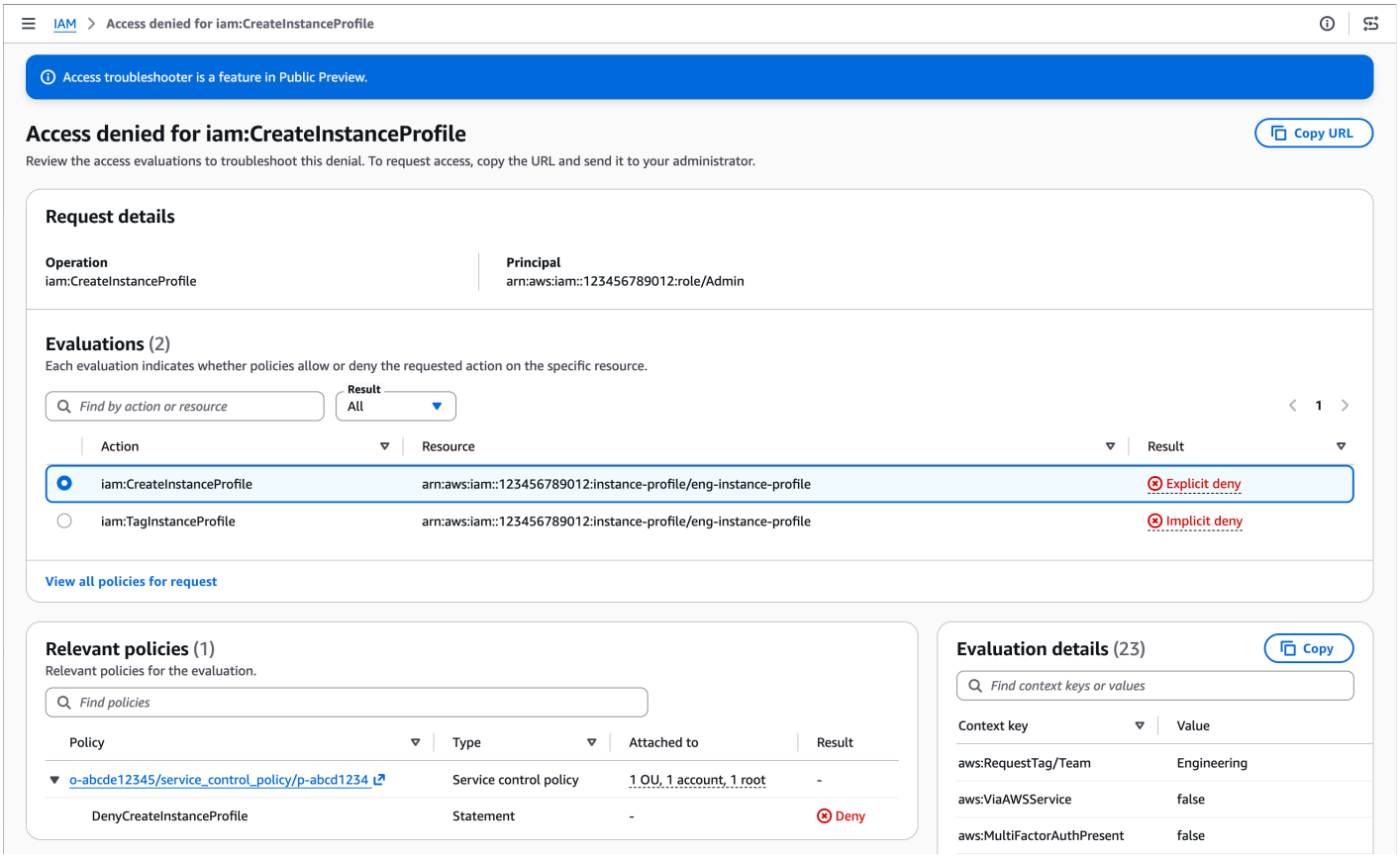
Usa questi risultati insieme per identificare a quale politica è stato negato l'accesso e perché. La valutazione riflette le politiche così come erano al momento del rifiuto. Le politiche potrebbero essere cambiate da allora.

AWS include automaticamente un ID di autorizzazione nelle risposte di errore di accesso negato per le API supportate, ma gli ID di autorizzazione non sono garantiti. In alcuni casi, AWS non registra i dettagli di autorizzazione per un rifiuto e la risposta all'errore non include un ID di autorizzazione. Quando i dettagli vengono registrati, potrebbero non essere disponibili immediatamente, perché li AWS registra in modo asincrono. Se una chiamata `GetRequestAuthorizationDetails` restituisce una risposta non trovata, attendi qualche istante e riprova. AWS conserva i dettagli dell'autorizzazione per 24 ore dopo il rifiuto.

Risolvi i problemi utilizzando AWS Console di gestione

Scegli il link nel messaggio di errore di accesso negato per aprire i dettagli dell'autorizzazione nella Console di Access Troubleshooter.

- Cosa cercare. L'Access Troubleshooter Console mostra la valutazione per ogni action/resource coppia nella richiesta, il risultato di ciascuna (`AllowExplicit deny`,, `oImplicit deny`) e le politiche prese in considerazione. Individua la coppia che è stata rifiutata e la politica che ha causato il rifiuto. Guarda la seguente schermata come esempio:



- Determina se il rifiuto era intenzionale. Rivedi la valutazione prima di modificare una politica. Un rifiuto può essere corretto e a volte la causa è il contesto della richiesta piuttosto che la politica. Ad esempio, in una richiesta potrebbe mancare un tag di sessione previsto o provenire da un percorso di rete imprevisto.
- Come risolvere il rifiuto. Individua la policy corrispondente e aggiornala per risolvere il problema di accesso. Per un rifiuto esplicito, rimuovi o restringi la `Deny` dichiarazione relativa all'azione. Per un rifiuto implicito, aggiungi una `Allow` dichiarazione per l'azione. Per ulteriori informazioni, consulta Modificare le policy IAM.

Chi può accedere ai dettagli dell'autorizzazione. Per una richiesta all'interno di un singolo account o di una singola organizzazione, chiunque sia autorizzato ad accedere `iam:GetRequestAuthorizationDetails` può accedere ai dettagli della richiesta. Per una richiesta che riguarda più di un'organizzazione, un utente di una determinata organizzazione può visualizzare solo i dettagli relativi alla propria organizzazione.

Risolvi i problemi relativi all'utilizzo AWS API o AWS CLI

Per risolvere un problema di rifiuto utilizzando l' AWS API o: AWS CLI

1. Individua l'ID di autorizzazione nella risposta all'errore di accesso negato.
2. Chiama `GetRequestAuthorizationDetails` con l'ID di autorizzazione, nello stesso in Regione AWS cui si è verificato il rifiuto. L'esempio seguente mostra il AWS CLI comando:

```
aws iam-toolbox get-request-authorization-details \
    --region us-east-1 \
    --authorization-id EXAMPLE1a2b3c4d5e6f7g8h9i0j
```

3. Esamina i risultati della valutazione per identificare quali politiche hanno negato l'accesso e il motivo del rifiuto.
4. Aggiorna le politiche identificate per risolvere il problema di accesso. Per ulteriori informazioni, consulta Modificare le policy IAM.

Quando non ricevi un ID di autorizzazione

AWS non restituisce un ID di autorizzazione nei seguenti casi:

- L'autorizzazione è troppo grande per essere registrata. Quando una richiesta comporta un numero elevato di politiche, AWS può decidere di non registrare la richiesta.
- Il servizio non è ancora supportato. Consulta Servizi supportati.
- L'API non è supportata. Alcune API all'interno di un servizio supportato non restituiscono un ID di autorizzazione.
- Il tipo di richiesta non è supportato. Ciò include le richieste effettuate da un servizio per conto dell'utente e le richieste anonime.

Se non ricevi un ID di autorizzazione, consulta altri modi Risoluzione dei problemi relativi ai messaggi di errore di accesso rifiutato per diagnosticare il rifiuto.

Ulteriori considerazioni

Autorizzazioni richieste. Per chiamare`GetRequestAuthorizationDetails`, devi disporre delle autorizzazioni per l'`iam:GetRequestAuthorizationDetails`azione.

API regionale. `GetRequestAuthorizationDetails`è un'API regionale. Chiamalo nello stesso Regione AWS punto in cui sono stati generati l'errore di accesso negato e l'ID di autorizzazione.

Cross-organization richieste. Quando una richiesta rifiutata coinvolge più di un'organizzazione, `GetRequestAuthorizationDetails` restituisce solo i dettagli che appartengono alla propria organizzazione. Ciò vale in entrambe le direzioni: se si fa parte dell'organizzazione del chiamante rifiutato, vengono visualizzati solo i dettagli relativi a tale organizzazione. Se fai parte dell'organizzazione della risorsa negata, vengono visualizzati solo i dettagli relativi a tale organizzazione. I dettagli di proprietà di un'altra organizzazione, ad esempio le politiche o i valori chiave di contesto di tale organizzazione, non vengono restituiti.

Se ricevi un errore di accesso negato da un'API, un servizio o una regione che non include ancora un ID di autorizzazione, consultaRisoluzione dei problemi relativi ai messaggi di errore di accesso rifiutato.

Servizi supportati

Il supporto per `GetRequestAuthorizationDetails` viene implementato gradualmente tra le AWS API e Regione AWS. Supporto `GetRequestAuthorizationDetails` per i seguenti servizi:

- La maggior parte delle API in IAM

Risoluzione dei problemi con l'utente root

Le informazioni seguenti ti aiutano a risolvere i problemi relativi all'utente root di un Account AWS.

> **Note**
>
> Account AWS managed using AWS Organizations potrebbe avere l'accesso root centralizzato abilitato per gli account dei membri. Questi account membri non dispongono di credenziali dell'utente root, non possono accedere come utente root e non possono recuperare la password dell'utente root. Contatta l'amministratore se devi eseguire un'operazione che richiede le credenziali dell'utente root.

Non riesco a eseguire le attività che mi aspetto di poter eseguire quando effettuo l'accesso come utente root dell'account

L'account deve essere un membro dell'organizzazione in AWS Organizations. L'amministratore dell'organizzazione potrebbe disporre di una policy di controllo del servizio (SCP) per limitare le

autorizzazioni dell'account. SCPs ha un impatto su tutti gli utenti, incluso l'utente root. Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .

Ho dimenticato la password dell'utente root per il mio Account AWS

Se sei un utente root e hai perso o dimenticato la password del tuo Account AWS, puoi reimpostarla. È necessario conoscere l'indirizzo e-mail utilizzato per creare Account AWS il e avere accesso all'account di posta elettronica. Per ulteriori informazioni, consulta Reimpostare una password dell'utente root persa o dimenticata.

Non ho accesso all'e-mail del mio Account AWS

Quando ne crei uno Account AWS, fornisci un indirizzo email e una password. Queste sono le credenziali per Utente root dell'account AWS. Se non sei sicuro dell'indirizzo e-mail associato al tuo Account AWS, cerca i messaggi inviati da `@signin.aws` o `@verify.signin.aws` verso qualsiasi indirizzo e-mail della tua organizzazione che potrebbe essere stato utilizzato per aprire il Account AWS.

Se conosci l'indirizzo e-mail ma non hai più accesso alla posta elettronica, prova innanzitutto a recuperare l'accesso all'e-mail. Utilizza una delle seguenti opzioni per ottenere l'accesso all'indirizzo e-mail:

- Se sei il proprietario del dominio dell'indirizzo e-mail, puoi ripristinare un indirizzo e-mail eliminato. In alternativa, è possibile impostare un metodo catch-all per l'account di posta elettronica. Un catch-all raccoglie tutti i messaggi inviati a indirizzi e-mail che non esistono più nel server di posta e li reindirizza a un altro indirizzo e-mail.
- Se l'indirizzo e-mail dell'account è parte del sistema di posta elettronica aziendale, si consiglia di contattare gli amministratori del sistema IT. Potrebbero essere in grado di aiutare a ottenere nuovamente l'accesso all'e-mail.

Se ancora non riesci ad accedere al tuo Account AWS, puoi trovare opzioni di supporto alternative alla pagina Contattaci.

Risoluzione dei problemi relativi alle policy IAM

Una policy è un'entità in AWS cui, quando è collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale, ad esempio un utente, effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta.

Le politiche vengono archiviate AWS come documenti JSON allegati ai principali come politiche basate sull'identità o alle risorse come politiche basate sulle risorse. Identity-based le politiche includono politiche gestite, politiche AWS gestite dal cliente e politiche in linea.

È possibile creare e modificare politiche gestite dal cliente Console di gestione AWS utilizzando le opzioni dell'editor Visual e JSON. Quando si visualizza una politica in Console di gestione AWS, è possibile visualizzare un riepilogo delle autorizzazioni concesse da tale politica. L'editor visivo e i riepiloghi di policy consentono di individuare e risolvere errori comuni durante la gestione delle policy IAM.

Tieni presente che tutte le policy IAM vengono archiviate utilizzando una sintassi che inizia con le regole di JavaScript Object Notation (JSON). Non è necessario conoscere questa sintassi per creare o gestire le policy. È possibile creare e modificare una policy utilizzando l'editor visivo nella Console di gestione AWS. Per ulteriori informazioni sulla sintassi JSON nelle policy IAM, consulta Sintassi del linguaggio della policy JSON IAM.

Risoluzione dei problemi degli argomenti della policy IAM

- Risoluzione dei problemi tramite l'editor visivo
  - Modifica della struttura delle policy
  - Scelta di un ARN della risorsa nell'editor visivo
  - Autorizzazioni nell'editor visivo
  - Specifica di più servizi nell'editor visivo
  - Riduzione delle dimensioni della policy nell'editor visivo
  - Correzione di servizi, operazioni o tipi di risorse non riconosciuti nell'editor visivo
- Risoluzione dei problemi tramite i riepiloghi delle policy
  - Riepilogo della policy mancante
  - Il riepilogo della policy include servizi, operazioni o tipi di risorse non riconosciuti
  - Il servizio non supporta i riepiloghi delle policy IAM
  - La policy non concede le autorizzazioni previste
- Risoluzione dei problemi di gestione delle policy
  - Collegamento o scollegamento di una policy in un account IAM
  - Modifica delle policy per le identità IAM in base alla loro attività
- Risoluzione dei problemi relativi ai documenti di policy JSON
  - Convalida delle policy

- Non ho autorizzazioni per la convalida di policy nell'editor JSON
- Più di un oggetto di policy JSON
- Più di un elemento di istruzione JSON
- Più di un elemento Effect, Action o Resource in un elemento di istruzione JSON
- Elemento versione JSON mancante

Risoluzione dei problemi tramite l'editor visivo

Quando si crea o si modifica una policy gestita dal cliente, è possibile utilizzare le informazioni nell'editor Visivo per semplificare la risoluzione degli errori della policy. Per visualizzare un esempio dell'editor visivo per creare una policy, consultare the section called "Controllo dell'accesso alle identità".

Modifica della struttura delle policy

Quando crei una policy, la AWS convalida, elabora e trasforma la policy prima di archiviarla. Quando la policy viene recuperata, la AWS trasforma nuovamente in un formato leggibile dall'uomo senza modificare le autorizzazioni. Ciò può causare differenze in ciò che viene visualizzato nell'editor visivo della policy o nella scheda JSON.

- I blocchi di autorizzazioni dell'editor visivo possono essere aggiunti, rimossi o riordinati e il contenuto all'interno di un blocco può essere ottimizzato.
- Nella scheda JSON lo spazio bianco non significativo può essere rimosso e gli elementi all'interno di mappe JSON possono essere riordinati. Inoltre, gli Account AWS ID all'interno degli elementi principali possono essere sostituiti dall'Amazon Resource Name (ARN) di. Utente root dell'account AWS

A causa di queste possibili modifiche, non è possibile confrontare i documenti di policy JSON come stringhe.

Quando crei una policy gestita dal cliente in Console di gestione AWS, puoi scegliere di lavorare interamente nell'editor JSON. Se non si apportano modifiche alla policy nell'editor Visivo e si seleziona Successivo dall'editor JSON, è meno probabile che la policy venga ristrutturata. Quando si utilizza l'editor visivo, IAM potrebbe ristrutturare la policy per ottimizzarne l'aspetto. Questa ristrutturazione esiste solo nella sessione di modifica e non viene salvata automaticamente.

Se la policy è stata ristrutturata nella sessione di modifica, IAM determina se salvare la ristrutturazione in base alle seguenti situazioni:

Utilizzo di questa opzione dell'editor	Se si modifica la policy	Quindi scegli Successivo da questa scheda	Quando si sceglie Save changes (Salva modifiche)
Visivo	Modificata	Visivo	La policy viene ristrutturata
Visivo	Modificata	JSON	La policy viene ristrutturata
Visivo	Non modificata	Visivo	La policy viene ristrutturata
JSON	Modificata	Visivo	La policy viene ristrutturata
JSON	Modificata	JSON	La struttura della policy non viene modificata
JSON	Non modificata	JSON	La struttura della policy non viene modificata

IAM potrebbe ristrutturare policy complesse o policy che hanno blocchi di autorizzazione o istruzioni per permettere più servizi, tipi di risorse o chiavi di condizioni.

Scelta di un ARN della risorsa nell'editor visivo

Quando si crea o si modifica una policy utilizzando l'editor visivo, è necessario prima selezionare un servizio, quindi selezionare operazioni da quel servizio. Se il servizio e le operazioni selezionate supportano la scelta di risorse specifiche, l'editor visivo elenca i tipi di risorse supportati. È quindi possibile selezionare Add ARN (Aggiungi ARN) per fornire i dettagli sulla risorsa. È possibile selezionare tra le seguenti opzioni per aggiungere un ARN per un tipo di risorsa.

- Utilizza il builder di ARN: in base al tipo di risorsa, è possibile che siano visualizzati campi diversi per la creazione dell'ARN. È anche possibile selezionare Any (Qualsiasi) per fornire le autorizzazioni per qualsiasi valore per l'impostazione specificata. Ad esempio, se si seleziona il gruppo di livello di accesso Lettura di Amazon EC2, allora le operazioni nella policy supportano il tipo di risorsa `instance`. Fornisci la regione, l'account e InstanceId i valori per la tua risorsa. Se si fornisce l'ID account ma si seleziona Qualsiasi per la regione e l'ID istanza, la policy concede le autorizzazioni a qualsiasi istanza dell'account.
- Digita o incolla l'ARN: puoi possibile specificare le risorse in base ai relativi Amazon Resource Name (ARN). È possibile includere caratteri jolly ***** in qualsiasi campo dell'ARN (tra ogni coppia di due punti). Per ulteriori informazioni, consulta Elementi della policy JSON IAM: Risorsa.

Autorizzazioni nell'editor visivo

Per impostazione predefinita, la policy creata tramite l'editor visuale permette le operazioni scelte. Per rifiutare invece le operazioni scelte, selezionare Switch to deny permissions (Passa a rifiuto autorizzazioni). Poiché le richieste vengono rifiutate per impostazione predefinita, si consiglia come best practice di sicurezza di permettere le autorizzazioni solo alle operazioni e alle risorse necessarie per un utente. È necessario creare un'istruzione per rifiutare le autorizzazioni solo se si desidera ignorare separatamente un'autorizzazione permessa da un'altra istruzione o policy. Si consiglia di limitare al minimo il numero di autorizzazioni di rifiuto perché possono aumentare la difficoltà di risoluzione dei problemi relative alle autorizzazioni. Per ulteriori informazioni sulla logica di valutazione della policy IAM, consulta Logica di valutazione delle policy.

> **Note**
>
> Per impostazione predefinita, solo l'utente Utente root dell'account AWS ha accesso a tutte le risorse di quell'account. Pertanto, se non è stato effettuato l'accesso come utente root, è necessario disporre delle autorizzazioni concesse da una policy.

Specifica di più servizi nell'editor visivo

Quando si utilizza l'editor visivo per creare una policy, è possibile selezionare solo un servizio alla volta. Si tratta di una best practice consigliata in quanto l'editor visivo consente in questo modo di selezionare tra le operazioni per tale singolo servizio. Quindi si sceglie tra le risorse supportate da tale servizio e le operazioni selezionate. Ciò semplifica la creazione e la risoluzione dei problemi della policy.

È anche possibile usare un carattere jolly (*) per specificare manualmente più servizi. Ad esempio, digitare **`Code*`** per fornire le autorizzazioni per tutti i servizi che iniziano con `Code`, ad esempio `CodeBuild` e `CodeCommit`. Tuttavia, è necessario digitare gli ARN della risorsa e le risorse per completare le policy. Inoltre, quando si salva la policy, potrebbe venire ristrutturata per includere ciascun servizio in un blocco di autorizzazioni separate.

In alternativa, per utilizzare una sintassi JSON (ad esempio, i caratteri jolly) per i servizi, crea, modifica e salva la policy utilizzando l'opzione dell'editor JSON.

Riduzione delle dimensioni della policy nell'editor visivo

Quando utilizzi l'editor visivo per creare una policy, IAM crea un documento JSON per archiviare la policy. È possibile visualizzare questo documento passando all'opzione dell'editor JSON. Se questo documento JSON supera il limite di dimensioni di una policy, l'editor visivo visualizza un messaggio di errore. Non sarai in grado di rivedere e salvare la policy. Per visualizzare i limiti di IAM per le dimensioni di una policy gestita, consulta Limiti di caratteri di IAM e STS.

Per ridurre le dimensioni delle policy nell'editor visivo, modificare la policy o spostare blocchi di autorizzazioni in un'altra policy. Il messaggio di errore include il numero di caratteri contenuto nel documento di policy. Puoi utilizzare queste informazioni per ridurre le dimensioni della policy.

Correzione di servizi, operazioni o tipi di risorse non riconosciuti nell'editor visivo

È possibile che venga visualizzato un avviso nell'editor visivo che indica che la policy include un servizio, un'operazione o un tipo di risorsa non riconosciuti.

> **Note**
>
> IAM rivede i nomi di servizi, le operazioni e i tipi di risorse per i servizi che supportano i riepiloghi della policy. Tuttavia, il riepilogo della policy può includere un valore di risorse o una condizione che non esiste. Esegui sempre un test delle policy tramite il simulatore di policy.

Se la policy include servizi, operazioni o tipi di risorse non riconosciuti, si è verificato uno dei seguenti errori:

- Servizio di anteprima: i servizi in anteprima non supportano l'editor visivo. Se partecipi all'anteprima, dovrai digitare manualmente le operazioni e gli ARN delle risorse per completare la policy. Puoi ignorare qualsiasi avviso e continuare. In alternativa, per digitare o incollare un documento della policy JSON è possibile scegliere l'opzione dell'editor JSON.

- Servizio personalizzato: i servizi personalizzati non supportano l'editor visivo. Se utilizzi un servizio personalizzato, dovrai digitare manualmente le operazioni e gli ARN delle risorse per completare la policy. Puoi ignorare qualsiasi avviso e continuare. In alternativa, per digitare o incollare un documento della policy JSON è possibile scegliere l'opzione dell'editor JSON.
- Il servizio non supporta l'editor visivo: se la policy include un servizio con disponibilità a livello generale (GA) non supportato dall'editor visivo, dovrai digitare manualmente gli ARN di operazioni e risorse per completare la policy. Puoi ignorare qualsiasi avviso e continuare. In alternativa, per digitare o incollare un documento della policy JSON è possibile scegliere l'opzione dell'editor JSON.

  I servizi disponibili a livello generale sono servizi che vengono rilasciati per il pubblico e non sono servizi di anteprima o personalizzati. Se un servizio non riconosciuto è disponibile a livello generale e il nome è scritto correttamente, significa che il servizio non supporta l'editor visivo. Per informazioni su come richiedere supporto per l'editor visivo o per il riepilogo della policy per un servizio con disponibilità generale, consultare Il servizio non supporta i riepiloghi delle policy IAM.
- L'operazione non supporta l'editor visivo: se la policy include un servizio supportato con un'operazione non supportata, dovrai digitare manualmente gli ARN di operazioni e risorse per completare la policy. Puoi ignorare qualsiasi avviso e continuare. In alternativa, per digitare o incollare un documento della policy JSON è possibile scegliere l'opzione dell'editor JSON.

  Se la policy include un servizio supportato con un'operazione non supportata, il servizio non supporta completamente l'editor visivo. Per informazioni su come richiedere supporto per l'editor visivo o per il riepilogo della policy per un servizio con disponibilità generale, consultare Il servizio non supporta i riepiloghi delle policy IAM.
- Il tipo di risorsa non supporta l'editor visivo: se la policy include un'operazione supportata con un tipo di risorsa non supportato, è possibile ignorare l'avviso e continuare. Tuttavia, IAM non è in grado di confermare di aver incluso risorse per tutte le operazioni selezionate e potrebbero venire visualizzati avvisi aggiuntivi.
- Refuso: quando si digita manualmente un servizio, un'operazione o una risorsa nell'editor visivo, è possibile che venga creata una policy che include un errore di battitura. Ti consigliamo di utilizzare l'editor visivo selezionando dall'elenco di servizi e operazioni. Quindi, completa la sezione delle risorse in base alle istruzioni. Se un servizio non supporta completamente l'editor visivo, potrebbe essere necessario digitare manualmente le parti della policy.

  Se si è certi che la policy non contenga nessuno degli errori sopra riportati, è possibile che includa un refuso. Verifica i problemi seguenti:

- Nomi di servizi, azioni e tipi di risorsa scritti in modo errato, ad esempio `s2` invece di `s3` o `ListMyBuckets` al posto di `ListAllMyBuckets`
- Testo non necessario negli ARN, ad esempio `arn:aws:s3: : :*`
- Due punti (:) mancanti nelle azioni, ad esempio `iam.CreateUser`

È possibile valutare una policy che potrebbe contenere refusi scegliendo Successivo per rivedere il riepilogo della policy. Quindi, verifica se la policy fornisce le autorizzazioni previste.

Risoluzione dei problemi tramite i riepiloghi delle policy

È possibile individuare e risolvere i problemi relativi ai riepiloghi delle policy.

Riepilogo della policy mancante

La console IAM include tabelle di riepilogo di policy che descrivono il livello di accesso, le risorse e le condizioni concesse o negate per ciascun servizio in una policy. Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione.

La tabella riepilogo della policy include un elenco di servizi e riepiloghi delle autorizzazioni definite dalla policy scelta. Il riepilogo della politica è disponibile nella pagina dei dettagli della politica per qualsiasi politica allegata o nella pagina delle politiche per le politiche gestite. Se non AWS è possibile visualizzare un riepilogo di una policy, la console visualizza il documento della policy JSON e il seguente errore:

Impossibile generare un riepilogo per questa policy. Puoi comunque visualizzare o modificare il documento di policy JSON.

Se la policy non include un riepilogo, si è verificato uno dei seguenti errori:

- Elemento della policy non supportato: IAM non supporta la generazione di riepiloghi di policy per le policy che includono uno dei seguenti elementi di policy:
  - `Principal`
  - `NotPrincipal`
  - `NotResource`
- Nessuna autorizzazione di policy: se una policy non fornisce le autorizzazioni valide, non è possibile generare il riepilogo della policy. Ad esempio, se una policy include una singola istruzione con l'elemento `"NotAction": "*"`, si permette l'accesso a tutte le operazioni ad eccezione di "tutte le operazioni" (*). Questo significa che concede accesso `Deny` o `Allow` a nulla.

> **Note**
>
> Prestare attenzione all'utilizzo di elementi di policy quali `NotPrincipal`, `NotAction` e `NotResource`. Per ulteriori informazioni sull'utilizzo degli elementi delle policy, consultare Documentazione di riferimento degli elementi delle policy JSON IAM.

Se si forniscono servizi e risorse non corrispondenti, è possibile creare una policy che non fornisce autorizzazioni valide. Ciò può verificarsi se si specificano le operazioni in un servizio e le risorse di un altro servizio. In questo caso, viene visualizzato il riepilogo della policy. L'unica indicazione che si è verificato un problema è che la colonna delle risorse nel riepilogo può includere una risorsa di un servizio diverso. Se questa colonna include una risorsa non corrispondente, è necessario verificare se ci sono errori nella policy. Per comprendere meglio la policy, esegui sempre un test tramite il simulatore di policy.

Il riepilogo della policy include servizi, operazioni o tipi di risorse non riconosciuti

Nella console IAM, se un riepilogo di policy include un simbolo di avvertenza
( ),
la policy potrebbe includere un tipo di servizio, operazione o risorsa non riconosciuto. Per ulteriori informazioni sulle avvertenze in un riepilogo della policy, consultare Riepilogo della policy (elenco di servizi).

> **Note**
>
> IAM rivede i nomi di servizio, le operazioni e i tipi di risorse per i servizi che supportano i riepiloghi della policy. Tuttavia, il riepilogo della policy può includere un valore di risorse o una condizione che non esiste. Esegui sempre un test delle policy tramite il simulatore di policy.

Se la policy include servizi, operazioni o tipi di risorse non riconosciuti, si è verificato uno dei seguenti errori:

- Servizio di anteprima: i servizi in anteprima non supportano i riepiloghi di policy.
- Servizio personalizzato: i servizi personalizzati non supportano i riepiloghi di policy.

- Il servizio non supporta riepiloghi: se la policy include un servizio disponibile a livello generale (GA) che non supporta riepiloghi di policy, allora il servizio viene incluso nella sezione Servizi non riconosciuti della tabella di riepilogo della policy. I servizi disponibili a livello generale sono servizi che vengono rilasciati per il pubblico e non sono servizi di anteprima o personalizzati. Se un servizio non riconosciuto è disponibile a livello generale e il nome è scritto correttamente, allora significa che il servizio non supporta i riepiloghi di policy IAM. Per informazioni su come richiedere supporto per il riepilogo della policy per un servizio con disponibilità generale, consultare Il servizio non supporta i riepiloghi delle policy IAM.
- L'operazione non supporta i riepiloghi: se la policy include un servizio supportato con un'operazione non supportata, allora l'operazione viene inclusa nella sezione Operazioni non riconosciute della tabella di riepilogo del servizio. Per ulteriori informazioni sulle avvertenze in un riepilogo di servizio, consultare Riepilogo del servizio (elenco di operazioni).
- Il tipo di risorsa non supporta i riepiloghi: se la policy include un'operazione supportata con un tipo di risorsa non supportato, allora la risorsa viene inclusa nella sezione Tipi di risorse non riconosciuti della tabella di riepilogo del servizio. Per ulteriori informazioni sulle avvertenze in un riepilogo di servizio, consultare Riepilogo del servizio (elenco di operazioni).
- Errore di battitura: AWS verifica che il JSON sia sintatticamente corretto e che la policy non includa errori di battitura o altri errori come parte della convalida della politica. ???

> **Note**
>
> Come best practice, ti consigliamo di utilizzare IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali. Consigliamo di aprire le policy esistenti e rivedere e risolvere eventuali suggerimenti di convalida della policy.

Il servizio non supporta i riepiloghi delle policy IAM

È possibile che i riepiloghi delle policy IAM o l'editor visivo non supportino un'operazione o un servizio disponibile a livello generale. I servizi disponibili a livello generale sono servizi che vengono rilasciati per il pubblico e che non sono servizi di anteprima o personalizzati. Se un servizio non riconosciuto è disponibile a livello generale e il nome è scritto correttamente, significa che il servizio non supporta queste caratteristiche. Se la policy include un servizio supportato con un'operazione non supportata, il servizio non supporta completamente il riepilogo della policy IAM.

Come richiedere che un servizio aggiunga il supporto per l'editor visivo o per il riepilogo della policy IAM

1. Accedi e apri la console IAM all' Console di gestione AWS indirizzo. https://console.aws.amazon.com/iam/
2. Individuare la policy che include il servizio non supportati:
   - Se la policy è una policy gestita, selezionare Policies (Policy) nel riquadro di navigazione. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
   - Se la policy è una policy inline collegata all'utente, selezionare Users (Utenti) nel riquadro di navigazione. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare. Nella tabella di policy per l'utente, espandere l'intestazione per il riepilogo della policy che si desidera visualizzare.
3. Nella parte sinistra del Console di gestione AWS piè di pagina, scegli Feedback. Nella casella Feedback per IAM, digita **`I request that the <ServiceName> service add support for IAM policy summaries and the visual editor`**. Se si desidera che più di un servizio supporti i riepiloghi, digitare **`I request that the <ServiceName1>, <ServiceName2>, and <ServiceName3> services add support for IAM policy summaries and the visual editor`**.

Come richiedere che un servizio aggiunga il supporto per l'editor visivo o per il riepilogo della policy IAM per un'operazione mancante

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Individuare la policy che include il servizio non supportati:
   - Se la policy è una policy gestita, selezionare Policies (Policy) nel riquadro di navigazione. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
   - Se la policy è una policy inline collegata all'utente, selezionare Users (Utenti) nel riquadro di navigazione. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare. Nella tabella delle policy per l'utente, selezionare il nome della policy che cui si desidera visualizzare per espandere il riepilogo della policy.
3. Nel riepilogo della policy, selezionare il nome del servizio che include un'operazione non supportata.

4. Nella parte sinistra del Console di gestione AWS piè di pagina, scegli Feedback. Nella casella Feedback per IAM, digita **`I request that the <ServiceName> service add IAM policy summary and the visual editor support for the <ActionName> action`**. Se si desidera segnalare più di un'operazione non supportata, digitare **`I request that the <ServiceName> service add IAM policy summary and the visual editor support for the <ActionName1>, <ActionName2>, and <ActionName3> actions`**.

Per richiedere che un altro servizio includa operazioni mancanti, ripetere gli ultimi tre passaggi.

La policy non concede le autorizzazioni previste

Per assegnare le autorizzazioni a un utente, gruppo, ruolo o risorsa, è necessario creare una policy, ovvero un documento che definisca le autorizzazioni. Il documento della policy include i seguenti elementi:

- Effetto: indica se la policy consente o nega l'accesso
- Azione: l'elenco delle operazioni consentite o rifiutate dalla policy
- Risorsa: l'elenco delle risorse in cui possono essere eseguite le operazioni
- Condizione (facoltativo): le circostanze in base alle quali la policy concede l'autorizzazione

Per informazioni su questi elementi di policy, consultare Documentazione di riferimento degli elementi delle policy JSON IAM.

Per garantire l'accesso, la policy deve definire un'operazione con una risorsa supportata. Se la policy include anche una condizione, tale condizione deve includere una chiave di condizione globale o deve essere applicata all'operazione. Per scoprire quali risorse sono supportate da un'operazione, consulta la documentazione di AWS relativa al servizio. Per sapere quali condizioni sono supportate da un'azione, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.

Verifica se la policy definisce un'operazione, una risorsa o una condizione che non concede autorizzazioni. Visualizza il riepilogo delle policy relative alla tua policy utilizzando la console IAM all'indirizzo https://console.aws.amazon.com/iam/. È possibile utilizzare riepiloghi di policy per identificare e risolvere i problemi della policy.

Esistono diversi motivi per cui un elemento potrebbe non concedere autorizzazioni nonostante sia definito nella policy IAM:

- Un'operazione è definita senza una risorsa applicabile
- Una risorsa è definita senza un'operazione applicabile
- Una condizione è definita senza un'operazione applicabile

Per visualizzare esempi di riepiloghi di policy che includono avvisi, consultare the section called "Riepilogo della policy (elenco di servizi)".

Un'operazione è definita senza una risorsa applicabile

La policy di seguito definisce tutte le operazioni `ec2:Describe*` e definisce una risorsa specifica. Nessuna delle `ec2:Describe` azioni viene concessa perché nessuna di queste azioni supporta autorizzazioni a livello di risorsa. Resource-level le autorizzazioni indicano che l'azione supporta le risorse che utilizzano gli ??? ARN nell'elemento della policy. `Resource` Se un'operazione non supporta le autorizzazioni a livello di risorsa, tale istruzione della policy deve utilizzare un carattere jolly (`*`) nell'elemento `Resource`. Per scoprire i servizi che supportano le autorizzazioni a livello di risorsa, consultare AWS servizi supportati da IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "arn:aws:ec2:us-east-2:111122223333:instance/*"
        }
    ]
}
```

Questa policy non fornisce alcuna autorizzazione e il riepilogo della policy include il seguente errore:

`This policy does not grant any permissions. To grant access, policies must have an action that has an applicable resource or condition.`

Per correggere la policy, è necessario utilizzare `*` nell'elemento `Resource`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "ec2:Describe*",
        "Resource": "*"
    }]
}
```

Una risorsa è definita senza un'operazione applicabile

La policy di seguito definisce una risorsa del bucket Amazon S3 ma non include un'operazione S3 che può essere eseguita su tale risorsa. Questa politica garantisce inoltre l'accesso completo a tutte le azioni di Amazon. CloudFront

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Action": "cloudfront:*",
        "Resource": [
            "arn:aws:cloudfront:*",
            "arn:aws:s3:::amzn-s3-demo-bucket"
        ]
        }
    ]
}
```

Questa politica fornisce le autorizzazioni per tutte le CloudFront azioni. Tuttavia, poiché la policy definisce le risorse `amzn-s3-demo-bucket` di S3 senza definire alcuna operazione S3, il riepilogo della policy include il seguente avviso:

`This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition.`

Per correggere questa policy per fornire autorizzazioni del bucket S3, è necessario definire operazioni S3 che possono essere eseguite su una risorsa bucket.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudfront:*",
                "s3:CreateBucket",
                "s3:ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
            "Resource": [
                "arn:aws:cloudfront:*",
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

In alternativa, per correggere questa politica in modo da fornire solo le CloudFront autorizzazioni, rimuovi la risorsa S3.

Una condizione è definita senza un'operazione applicabile

La policy di seguito definisce due operazioni Amazon S3 per tutte le risorse S3, se il prefisso S3 è uguale a `custom` e l'ID della versione è uguale a `1234`. Tuttavia, la chiave della condizione `s3:VersionId` viene utilizzata per il tagging della versione dell'oggetto e non è supportata dalle operazioni bucket definite. Per sapere quali condizioni sono supportate da un'azione, consulta Azioni, risorse e chiavi di condizione per AWS i servizi e scegli il servizio per visualizzare la documentazione del servizio per le chiavi di condizione.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketVersions",
                "s3:ListBucket"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": [
                        "custom"
                    ],
                    "s3:VersionId": [
                        "1234"
                    ]
                }
            }
        }
    ]
}
```

Questa policy fornisce le autorizzazioni per l'operazione `s3:ListBucketVersions` e l'operazione `s3:ListBucket` se il nome del bucket include il prefisso `custom`. Tuttavia, poiché la condizione `s3:VersionId` non è supportata da nessuna delle operazioni definite, il riepilogo della policy include il seguente errore:

`This policy does not grant any permissions. To grant access, policies must have an action that has an applicable resource or condition.`

Per correggere questa policy e utilizzare il tagging della versione dell'oggetto S3, è necessario definire un'operazione S3 che supporti la chiave della condizione `s3:VersionId`.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketVersions",
                "s3:ListBucket",
                "s3:GetObjectVersion"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": [
                        "custom"
                    ],
                    "s3:VersionId": [
                        "1234"
                    ]
                }
            }
        }
    ]
}
```

Questa policy fornisce le autorizzazioni per ogni operazione e condizione nella policy. Tuttavia, la policy non fornisce nessuna autorizzazione perché non esiste un caso in cui una singola operazione corrisponda a entrambe le condizioni. Al contrario, è necessario creare due dichiarazioni separate che includano ciascuna solo le operazioni con le condizioni a cui si applicano.

Per correggere questa policy, è necessario creare due istruzioni. La prima istruzione include le operazioni che supportano la condizione `s3:prefix` e la seconda istruzione include le operazioni che supportano la condizione `s3:VersionId`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:ListBucketVersions",
                "s3:ListBucket"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": "custom"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObjectVersion",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:VersionId": "1234"
                }
            }
        }
    ]
}
```

Risoluzione dei problemi di gestione delle policy

È possibile individuare e risolvere i problemi correlati alla gestione delle policy.

Collegamento o scollegamento di una policy in un account IAM

Alcune policy AWS gestite sono collegate a un servizio. Queste policy vengono utilizzate solo con un ruolo collegato a un servizio per tale servizio. Nella console IAM, quando si visualizza la pagina Dettagli della policy, sarà presente un banner per indicare che la policy è collegata a un servizio. Non è possibile collegare questa policy a un utente, un gruppo o un ruolo all'interno di IAM. Quando si crea un ruolo collegato al servizio per il servizio, questa policy viene automaticamente collegata al nuovo ruolo. Poiché la policy è obbligatoria, non è possibile distaccare la policy dal ruolo collegato al servizio.

Modifica delle policy per le identità IAM in base alla loro attività

È possibile aggiornare le policy per le identità IAM (utenti, gruppi e ruoli) in base alla loro attività. Per fare ciò, visualizza gli eventi del tuo account nella cronologia degli CloudTrail eventi. CloudTrail i

registri degli eventi includono informazioni dettagliate sugli eventi che puoi utilizzare per modificare le autorizzazioni della politica.

Un utente o un ruolo sta tentando di eseguire un'azione in AWS e tale richiesta viene negata.

Valuta se l'utente o il ruolo deve disporre dell'autorizzazione per eseguire l'operazione. In questo caso, è possibile aggiungere alla policy l'operazione e anche l'ARN della risorsa a cui cerca di accedere.

Un utente o un ruolo dispone di autorizzazioni che non utilizza.

Valuta la possibilità di rimuovere tali autorizzazioni dalla loro policy. Verifica che le policy concedano i privilegi minimi necessari per eseguire solo le operazioni necessarie.

Per ulteriori informazioni sull'utilizzo CloudTrail, vedere Visualizzazione CloudTrail degli eventi nella CloudTrail console nella Guida per l'AWS CloudTrail utente.

Risoluzione dei problemi relativi ai documenti di policy JSON

È possibile individuare e risolvere i problemi relativi ai documenti di policy JSON.

Convalida delle policy

Quando crei o si modifichi una policy JSON, IAM può eseguire la convalida delle policy per facilitare la creazione di una policy efficace. IAM identificherà gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente le policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni sui controlli di policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

Non ho autorizzazioni per la convalida di policy nell'editor JSON

In Console di gestione AWS, potresti ricevere il seguente errore se non disponi delle autorizzazioni per visualizzare i risultati della convalida delle policy di IAM Access Analyzer:

```
You need permissions. You do not have the permissions required to perform this operation. Ask your administrator to add permissions.
```

Per risolvere questo errore, chiedere all'amministratore di aggiungere l'autorizzazione `access-analyzer:ValidatePolicy` per proprio conto.

Più di un oggetto di policy JSON

Una policy IAM deve essere costituita da un solo oggetto JSON. È possibile denotare un oggetto racchiudendolo tra parentesi graffe { }. Puoi nidificare altri oggetti all'interno di un oggetto JSON incorporando ulteriori parentesi graffe { } all'interno della coppia esterna. Una policy deve contenere solo una coppia più esterna di { } parentesi graffe. L'esempio seguente non è corretto perché contiene due oggetti al livello superiore (richiamati in): *red*

JSON

```
{
      "Version":"2012-10-17",
      "Statement":
      {
         "Effect":"Allow",
         "Action":"ec2:Describe*",
         "Resource":"*"
      }
    }
    {
      "Statement": {
         "Effect": "Allow",
         "Action": "s3:*",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }
    }
```

Tuttavia, è possibile soddisfare l'intenzione dell'esempio precedente con l'uso di una corretta grammatica di policy. Anziché includere due oggetti di policy completi ciascuno con il proprio elemento `Statement`, è possibile combinare i due blocchi in un singolo elemento `Statement`. L'elemento `Statement` dispone di una serie di due oggetti come valore, come mostrato nell'esempio seguente (evidenziato in grassetto):

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [
        {
```

```
          "Effect": "Allow",
          "Action": "ec2:Describe*",
          "Resource":"*"
        },
        {
          "Effect": "Allow",
          "Action": "s3:*",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
      ]
    }
```

Più di un elemento di istruzione JSON

Questo errore potrebbe apparentemente sembrare una variazione della sezione precedente. Tuttavia, sintatticamente si tratta di un altro tipo di errore. L'esempio seguente include un solo oggetto della policy come indicato da una singola coppia di parentesi graffe { } al livello più alto. Tuttavia, quell'oggetto contiene due elementi `Statement` al suo interno.

Una policy IAM deve contenere solo un elemento `Statement`, che include il nome (`Statement`) che appare alla sinistra di due punti, seguito dal rispettivo valore sulla destra. Il valore di un elemento `Statement` deve essere un oggetto, contrassegnato da parentesi graffe {}, che contiene un elemento `Effect`, un elemento `Action` e un elemento `Resource`. L'esempio seguente non è corretto perché contiene due `Statement` elementi nell'oggetto policy (richiamato in*red*):

JSON

```
{
      "Version":"2012-10-17",
      "Statement": {
        "Effect": "Allow",
        "Action": "ec2:Describe*",
        "Resource": "*"
      },
      "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }
    }
```

Un oggetto valore può essere una matrice di oggetti a valore multiplo. Per risolvere questo problema, combinare i due elementi `Statement` in un unico elemento con una matrice di oggetti, come mostrato nell'esempio seguente (evidenziato in grassetto):

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": "ec2:Describe*",
          "Resource":"*"
        },
        {
          "Effect": "Allow",
          "Action": "s3:*",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
     ]
    }
```

Il valore dell'elemento `Statement` è una matrice di oggetti. La matrice in questo esempio è costituita da due oggetti, ognuno dei quali è di per sé un valore corretto per un elemento `Statement`. Ogni oggetto nella matrice è separato da virgole.

Più di un elemento Effect, Action o Resource in un elemento di istruzione JSON

Per quanto riguarda il valore della `Statement` name/value coppia, l'oggetto deve essere composto da un solo `Effect` elemento, un `Action` elemento e un `Resource` elemento. La policy seguente non è corretta poiché include due elementi `Effect` in `Statement`:

JSON

```
{
      "Version":"2012-10-17",
      "Statement": {
        "Effect": "Deny",
        "Effect": "Allow",
```

```
      "Action": "ec2:* ",
      "Resource": "*"
    }
  }
```

> **Note**
>
> Il motore di policy non permette tali errori nelle policy nuove o modificate. Tuttavia, il motore di policy permettere le policy che sono state salvate prima che il motore fosse aggiornato. Il comportamento delle policy esistenti con l'errore è il seguente:
>
> - Più elementi `Effect`: viene osservato solo l'ultimo elemento `Effect`. Gli altri valori vengono ignorati.
> - Più elementi `Action`: tutti gli elementi `Action` vengono combinati internamente e trattati come se fossero un unico elenco.
> - Più elementi `Resource`: tutti gli elementi `Resource` vengono combinati internamente e trattati come se fossero un unico elenco.
>
> Il motore di policy non permette di salvare alcuna policy con errori di sintassi. Correggi gli errori nella policy prima di salvarla. Rivedi e correggi tutti i suggerimenti di convalida delle policy per le tue policy.

In ogni caso, la soluzione consiste nel rimuovere l'elemento aggiuntivo. Per elementi `Effect`, è semplice: se si desidera che l'esempio precedente rifiuti le autorizzazioni per le istanze Amazon EC2, è necessario rimuovere la riga `"Effect": "Allow",` dalla policy come segue:

JSON

```
{
      "Version":"2012-10-17",
      "Statement": {
        "Effect": "Deny",
        "Action": "ec2:*",
        "Resource": "*"
      }
    }
```

Tuttavia, se l'elemento duplicato è `Action` oppure `Resource`, la risoluzione può essere più complicata. Potrebbero essere presenti più operazioni per cui si desidera permettere (o rifiutare) l'autorizzazione oppure si potrebbe voler controllare l'accesso a più risorse. Ad esempio, l'esempio seguente non è corretto perché contiene più `Resource` elementi (richiamati in*red*):

JSON

```
{
      "Version":"2012-10-17",
      "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }
    }
```

Tutti gli elementi necessari in un oggetto valore di un elemento `Statement` può essere presente una sola volta. La soluzione consiste nel posizionare ogni valore in una matrice. L'esempio che segue illustra questa operazione separando i due elementi risorsa in un elemento `Resource` con una matrice come valore oggetto (evidenziata in grassetto):

JSON

```
{
      "Version":"2012-10-17",
      "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": [
          "arn:aws:s3:::amzn-s3-demo-bucket",
          "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ]
      }
    }
```

Elemento versione JSON mancante

Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Per contro, una versione della policy viene creata quando si modifica una policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version. Per ulteriori informazioni sulle versioni di policy, consultare the section called "Controllo delle versioni delle policy IAM".

Man mano che AWS le funzionalità si evolvono, vengono aggiunte nuove funzionalità alle policy IAM per supportarle. A volte, un aggiornamento della sintassi della policy include un nuovo numero di versione. Se si utilizzano le caratteristiche più recenti della grammatica di policy nella policy, è necessario indicare al motore di analisi delle policy quale versione si sta utilizzando.

La versione di policy predefinita è "2008-10-17". Se si desidera utilizzare qualsiasi funzione di policy introdotta successivamente, è necessario specificare il numero di versione che supporta la funzionalità desiderata. Si consiglia di includere sempre il numero di versione della sintassi di policy più recente, che è al momento `"Version": "2012-10-17"`.

Ad esempio, la policy seguente non è corretta perché utilizza una variabile di policy `${...}` nell'ARN per una risorsa. Ma non riesce a specificare una versione della sintassi delle policy che supporti le variabili di policy (richiamate in*red*):

```
{
  "Statement":
  {
    "Action": "iam:*AccessKey*",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::123456789012:user/${aws:username}"
  }
}
```

Aggiungendo un elemento `Version` nella parte superiore della policy con il valore `2012-10-17`, la prima versione di API IAM che supporta variabili di policy, è possibile risolvere il problema (evidenziato in grassetto):

JSON

```
{
  "Version":"2012-10-17",
  "Statement":
  {
    "Action": "iam:*AccessKey*",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::123456789012:user/${aws:username}"
  }
}
```

Risoluzione dei problemi relativi alle passkey e alle chiavi di sicurezza FIDO

Utilizza le informazioni contenute in questa pagina per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo delle chiavi di sicurezza FIDO2.

Argomenti

- Non riesco ad abilitare la chiave di sicurezza FIDO
- Non riesco a effettuare l'accesso utilizzando la chiave di sicurezza FIDO
- Ho perso o danneggiato la mia chiave di sicurezza FIDO
- Altri problemi.

Non riesco ad abilitare la chiave di sicurezza FIDO

Consulta le seguenti soluzioni a seconda del tuo stato come amministratore di sistema o utente IAM.

Utenti IAM

Se non riesci ad abilitare la chiave di sicurezza FIDO, verifica quanto segue:

- Stai utilizzando una configurazione supportata?

  IAM supporta dispositivi di sicurezza FIDO2 che si connettono ai dispositivi tramite USB, Bluetooth o NFC. IAM supporta anche gli autenticatori di piattaforma come TouchID o FaceID. IAM non supporta la registrazione locale delle passkey per Windows Hello. Per creare e utilizzare le

passkey, gli utenti Windows devono utilizzare l'autenticazione tra dispositivi, che prevede l'utilizzo di una passkey di un dispositivo, ad esempio un dispositivo mobile, o di una chiave di sicurezza hardware per accedere su un altro dispositivo, ad esempio un laptop.

Per informazioni sui dispositivi e sui browser che puoi utilizzare con WebAuthn e AWS, vediConfigurazioni supportate per l'uso delle passkey e delle chiavi di sicurezza.

- Stai utilizzando plug-in di browser?

  AWS non supporta l'uso di plugin per aggiungere il supporto WebAuthn del browser. Utilizza invece un browser che offra il supporto nativo dello WebAuthn standard.

  Anche se utilizzi un browser supportato, potresti avere un plug-in incompatibile con WebAuthn. Un plug-in incompatibile potrebbe impedirti di abilitare e utilizzare la tua chiave FIDO-compliant di sicurezza. Disabilita qualsiasi plug-in che può risultare incompatibile e riavvia il browser. Quindi riprova ad abilitare la chiave di sicurezza FIDO.

- Disponi delle autorizzazioni appropriate?

  Se non hai nessuno dei problemi descritti precedentemente, è possibile che non disponi delle autorizzazioni appropriate. Contatta l'amministratore di sistema.

Amministratori di sistema

Se i tuoi utenti IAM non sono in grado di abilitare le relative chiavi di sicurezza FIDO nonostante utilizzino una configurazione supportata, controlla le loro autorizzazioni. Per un esempio dettagliato, consulta Tutorial IAM: consentire agli utenti di gestire le proprie credenziali e impostazioni MFA.

Non riesco a effettuare l'accesso utilizzando la chiave di sicurezza FIDO

Se non riesci ad accedere Console di gestione AWS utilizzando la tua chiave di sicurezza FIDO, consulta prima. Configurazioni supportate per l'uso delle passkey e delle chiavi di sicurezza Se utilizzi una configurazione supportata, ma non puoi effettuare l'accesso, contatta l'amministratore di sistema per ricevere assistenza.

Ho perso o danneggiato la mia chiave di sicurezza FIDO

Puoi assegnare a un utente fino a otto dispositivi MFA in qualsiasi combinazione dei tipi di MFA attualmente supportati. Con più dispositivi MFA, per accedere alla Console di gestione AWSè necessario un solo dispositivo MFA. La sostituzione di una chiave di sicurezza FIDO è analoga alla

sostituzione di un token TOTP hardware. Se si perde o si rompe qualsiasi tipo di dispositivo MFA, consulta Recuperare un'identità protetta da MFA in IAM.

Altri problemi.

Se hai un problema con le chiavi di sicurezza FIDO non descritto qui, procedi in uno dei seguenti modi:

- Utenti IAM: contattare l'amministratore di sistema.
- Account AWS utenti root: contatta l'AWS assistenza.

Risoluzione dei problemi relativi ai ruoli IAM

Utilizza le informazioni contenute in questa pagina per eseguire la diagnosi e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di ruoli IAM.

Argomenti

- Non è possibile assumere un ruolo
- È apparso un nuovo ruolo nel mio AWS account
- Non riesco a modificare o eliminare un ruolo nel mio Account AWS
- Non sono autorizzato a svolgere: iam: PassRole
- Perché non posso assumere un ruolo con una sessione di 12 ore? (AWS CLI, AWS API)
- Viene visualizzato un errore quando provo a passare da un ruolo a un altro nella console IAM
- Il mio ruolo ha un policy che mi consente di eseguire un'operazione, ma ricevo "Accesso negato"
- Il servizio non ha creato la versione delle policy predefinite del ruolo
- Non esiste un caso d'uso per un ruolo di servizio nella console

Non è possibile assumere un ruolo

Verifica quanto segue:

- Per consentire agli utenti di assumere nuovamente il ruolo corrente all'interno di una sessione di ruolo, specifica il ruolo ARN o Account AWS ARN come principale nella politica di fiducia dei ruoli. Servizi AWS che forniscono risorse di calcolo come Amazon EC2, Amazon ECS, Amazon EKS e Lambda forniscono credenziali temporanee e le aggiornano automaticamente. Ciò garantisce di

disporre sempre di un set di credenziali valido. Per questi servizi, non è necessario riassumere il ruolo attuale per ottenere credenziali temporanee. Tuttavia, se intendi passare tag di sessione o una Policy di sessione, devi riassumere il ruolo attuale. Per informazioni su come modificare una politica di fiducia dei ruoli per aggiungere il ruolo principale ARN o ARN, consulta. Account AWS Aggiornamento di una policy di attendibilità del ruolo

- Quando assumi un ruolo utilizzando il Console di gestione AWS, assicurati di utilizzare il nome esatto del tuo ruolo. I nomi dei ruoli fanno infatti distinzione tra maiuscole e minuscole.
- Quando assumi un ruolo utilizzando l' AWS STS API o AWS CLI, assicurati di utilizzare il nome esatto del tuo ruolo nell'ARN. I nomi dei ruoli fanno infatti distinzione tra maiuscole e minuscole.
- Quando assumi un ruolo utilizzando un provider di identità SAML-based federato e la crittografia SAML è abilitata, assicurati di aver caricato una chiave di decrittografia privata valida per il provider di identità SAML. Per ulteriori informazioni, consulta Gestire le chiavi di crittografia SAML.
- Verifica che la policy IAM conceda l'autorizzazione per chiamare `sts:AssumeRole` per il ruolo che desideri assumere. L'elemento `Action` della policy IAM deve consentire di chiamare l'operazione `AssumeRole`. Inoltre, l'elemento `Resource` della policy IAM deve specificare il ruolo che desideri assumere. Ad esempio, l'elemento `Resource` può specificare un ruolo in base all'Amazon Resource Name (ARN) o utilizzando un carattere jolly (*). Ad esempio, almeno una policy applicabile è necessaria per concedere le autorizzazioni simili a quanto segue:

```
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::account_id_number:role/role-name-you-want-to-assume"
```

- Verifica che l'identità IAM sia taggata con eventuali tag richiesti dalla policy IAM. Ad esempio, nella seguente policy di autorizzazione, l'elemento `Condition` richiede che il principale richiedente l'assunzione del ruolo debba avere un determinato tag. Al principale deve essere applicato il tag `department = HR` o `department = CS`. In caso contrario, non può assumere quel ruolo. Per ulteriori informazioni sul tagging di utenti e ruoli IAM, consulta the section called "Tag per risorse IAM".

```
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "*",
    "Condition": {"StringEquals": {"aws:PrincipalTag/department": [
            "HR",
            "CS"
        ]}}
```

- Verificare di soddisfare tutte le condizioni specificate nella policy di affidabilità del ruolo. Una `Condition` può specificare una data di scadenza, un ID esterno o che una richiesta deve provenire solo da indirizzi IP specifici. Considera l'esempio seguente: se la data corrente è qualsiasi momento dopo la data specifica, la policy non corrisponde mai e non è in grado di concedere l'autorizzazione per assumere il ruolo.

```
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::account_id_number:role/role-name-you-want-to-assume"
    "Condition": {
        "DateLessThan" : {
            "aws:CurrentTime" : "2016-05-01T12:00:00Z"
        }
    }
```

- Verifica che l'entità Account AWS da cui stai chiamando `AssumeRole` sia un'entità attendibile per il ruolo che stai assumendo. Le entità affidabili vengono definite come `Principal` in una policy di affidabilità del ruolo. L'esempio seguente è una policy di affidabilità collegata al ruolo che desideri assumere. In questo esempio, l'ID account con l'utente IAM con il quale hai effettuato l'accesso deve essere 123456789012. Se il numero di account non è elencato nell'elemento `Principal` della policy di affidabilità del ruolo, non puoi assumere il ruolo. Ciò è valido indipendentemente dalle autorizzazioni concesse nelle policy di accesso. Notare che la policy di esempio limita le autorizzazioni a operazioni che si verificano tra il 1° luglio 2017 e il 31 dicembre 2017 (UTC), inclusi. Se si effettua l'accesso prima o dopo tali date, la policy non corrisponde e non è possibile assumere il ruolo.

```
    "Effect": "Allow",
    "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
    "Action": "sts:AssumeRole",
    "Condition": {
      "DateGreaterThan": {"aws:CurrentTime": "2017-07-01T00:00:00Z"},
      "DateLessThan": {"aws:CurrentTime": "2017-12-31T23:59:59Z"}
    }
```

- Identità di origine: gli amministratori possono configurare i ruoli in modo da richiedere alle identità di passare una stringa personalizzata che identifica la persona o l'applicazione in cui vengono eseguite le azioni AWS, denominata identità di origine. Verifica se il ruolo assunto richiede l'impostazione di un'identità di origine. Per ulteriori informazioni sull'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

È apparso un nuovo ruolo nel mio AWS account

Alcuni AWS servizi richiedono l'utilizzo di un tipo di ruolo di servizio univoco collegato direttamente al servizio. Questo ruolo collegato ai servizi è predefinito dal servizio e include tutte le autorizzazioni che il servizio richiede. Ciò rende più semplice la configurazione di un servizio perché non si devono aggiungere manualmente le autorizzazioni necessarie. Per informazioni generali sui ruoli collegati al servizio, consultare Creare un ruolo collegato ai servizi.

Un servizio potrebbe essere già in utilizzo quando inizia a supportare i ruoli collegati al servizio. In questo caso, è possibile ricevere un'e-mail con informazioni su un nuovo ruolo nell'account. Questo ruolo include tutte le autorizzazioni delle quali il servizio ha bisogno per eseguire operazioni a proprio nome. Non bisogna eseguire alcuna operazione per supportare questo ruolo. Tuttavia, non bisogna eliminare il ruolo dall'account. Altrimenti si potrebbero rimuovere le autorizzazioni delle quali il servizio ha bisogno per accedere alle risorse AWS . Puoi visualizzare i ruoli collegati al servizio nel tuo account accedendo alla pagina Ruoli IAM della console IAM. Service-linked i ruoli vengono visualizzati con (Service-linked role) nella colonna Entità attendibili della tabella.

Per informazioni sui servizi che supportano i ruoli collegati ai servizi, vedi AWS servizi supportati da IAM e cerca i servizi con Sì nella colonna Service-Linked Ruolo. Per ulteriori informazioni sull'utilizzo di un ruolo collegato ai servizi per un servizio, selezionare il collegamento Yes (Sì).

Non riesco a modificare o eliminare un ruolo nel mio Account AWS

Non è possibile eliminare o modificare le autorizzazioni per un ruolo collegato ai servizi in IAM. Questi ruoli includono trust e autorizzazioni predefiniti richiesti dal servizio per eseguire operazioni a proprio nome. Puoi utilizzare la console o l'API IAM per modificare solo la descrizione di un ruolo collegato al servizio. AWS CLI Puoi visualizzare i ruoli collegati al servizio nel tuo account accedendo alla pagina IAM Roles nella console. Service-linked i ruoli vengono visualizzati con (Service-linked role) nella colonna Entità attendibili della tabella. Un banner nella pagina Summary (Riepilogo) del ruolo indica anche che un ruolo è un ruolo collegato ai servizi. È possibile gestire ed eliminare questi ruoli solo attraverso il servizio collegato, se quel servizio supporta l'operazione. Fare attenzione quando si modifica o elimina un ruolo collegato ai servizi poiché tale operazione può rimuovere le autorizzazioni delle quali il servizio ha bisogno per accedere alle risorse AWS .

Per informazioni sui servizi che supportano i ruoli collegati ai servizi, vedi AWS servizi supportati da IAM e cerca i servizi con Sì nella colonna Service-Linked Ruolo.

Non sono autorizzato a svolgere: iam: PassRole

Quando si crea un ruolo collegato ai servizi, è necessario disporre delle autorizzazioni per inoltrare quel ruolo al servizio. Alcuni servizi creano automaticamente un ruolo collegato ai servizi nell'account quando si esegue un'azione in quel servizio. Ad esempio, Amazon EC2 Auto Scaling crea il ruolo collegato ai servizi `AWSServiceRoleForAutoScaling` la prima volta che si crea un gruppo Auto Scaling. Se si cerca di creare un gruppo Auto Scaling senza l'autorizzazione `PassRole`, si riceve il seguente messaggio di errore:

```
ClientError: An error occurred (AccessDenied) when calling the
PutLifecycleHook operation: User: arn:aws:sts::111122223333:assumed-role/
Testrole/Diego is not authorized to perform: iam:PassRole on resource:
arn:aws:iam::111122223333:role/aws-service-role/autoscaling.amazonaws.com/
AWSServiceRoleForAutoScaling
```

Per risolvere questo errore, chiedere all'amministratore di aggiungere l'autorizzazione `iam:PassRole` per proprio conto.

Per scoprire i servizi che supportano i ruoli collegati ai servizi, consulta AWS servizi supportati da IAM. Per scoprire se un servizio crea automaticamente un ruolo collegato ai servizi, selezionare il link Yes (Sì) per visualizzare la documentazione del ruolo collegato ai servizi per quel servizio.

Perché non posso assumere un ruolo con una sessione di 12 ore? (AWS CLI, AWS API)

Quando si utilizzano le operazioni AWS STS `AssumeRole*` API o `assume-role*` CLI per assumere un ruolo, è possibile specificare un valore per il `DurationSeconds` parametro. Puoi specificare un valore da 900 secondi (15 minuti) fino alla durata massima impostata per la sessione per il ruolo. Se si specifica un valore superiore a questa impostazione, l'operazione ha esito negativo. Questa impostazione può avere un valore massimo di 12 ore. Ad esempio, se si specifica una durata di sessione di 12 ore, ma l'amministratore ha impostato la durata massima di sessione a 6 ore, l'operazione ha esito negativo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Aggiornamento della durata massima della sessione per un ruolo.

Se si utilizza l'concatenamento dei ruoli (utilizzando un ruolo per assumere un secondo ruolo), la sessione è limitata a un massimo di un'ora. Se successivamente utilizzi il parametro `DurationSeconds` per fornire un valore superiore a un'ora, l'operazione ha esito negativo.

Viene visualizzato un errore quando provo a passare da un ruolo a un altro nella console IAM

Le informazioni immesse nella pagina Cambia ruolo devono corrispondere a quelle relative al ruolo. In caso contrario, l'operazione non riesce e viene visualizzato il seguente errore:

```
Invalid information in one or more fields. Check your information or
contact your administrator.
```

Se viene visualizzato questo errore, verificare che le seguenti informazioni siano corrette:

- ID o alias dell'account: l' Account AWS ID è un numero di 12 cifre. Il tuo account potrebbe avere un alias, che è un identificativo intuitivo come il nome della tua azienda che può essere utilizzato al posto del tuo ID. Account AWS In questo campo è possibile utilizzare l'ID account o l'alias.
- Nome ruolo: i nomi dei ruoli fanno distinzione tra maiuscole e minuscole. L'ID account e il nome del ruolo devono corrispondere a quelli configurati per il ruolo.

Se si continua a ricevere un messaggio di errore, contattare l'amministratore per verificare le informazioni precedenti. La policy di attendibilità del ruolo o la policy dell'utente IAM potrebbe limitare l'accesso. L'amministratore può verificare le autorizzazioni per questi criteri.

Il mio ruolo ha un policy che mi consente di eseguire un'operazione, ma ricevo "Accesso negato"

La sessione del ruolo potrebbe essere limitata dalle policy di sessione. Quando richiedi credenziali di sicurezza temporanee utilizzando il codice AWS STS, puoi facoltativamente passare policy di sessione in linea o gestite. ??? Le policy di sessione sono policy avanzate che vengono passate come parametro durante la creazione di una sessione temporanea per un ruolo a livello di programmazione. Puoi passare un singolo documento della policy di sessione inline JSON utilizzando il parametro `Policy`. Puoi utilizzare il parametro `PolicyArns` per specificare fino a 10 policy di sessione gestite. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. In alternativa, se l'amministratore o un programma personalizzato fornisce le credenziali temporanee, potrebbero includere policy di sessione per limitare l'accesso.

Il servizio non ha creato la versione delle policy predefinite del ruolo

Un ruolo di servizio è un ruolo che un servizio assume per eseguire operazioni nel tuo account a tuo nome. Quando si configurano alcuni ambienti di AWS servizio, è necessario definire un ruolo che il servizio deve assumere. In alcuni casi, il servizio crea il ruolo di servizio e le relative policy in IAM per tuo conto. Sebbene sia possibile modificare o eliminare il ruolo di servizio e la relativa policy dall'interno di IAM, AWS consiglia di non seguire questa opzione. Il ruolo e il criterio sono destinati solo a tale servizio. Se si modifica il criterio e si imposta un altro ambiente, quando il servizio tenta di utilizzare lo stesso ruolo e criterio, l'operazione potrebbe non riuscire.

Ad esempio, quando si utilizza AWS CodeBuild per la prima volta, il servizio crea un ruolo denominato`codebuild-RWBCore-service-role`. Tale ruolo di servizio utilizza il criterio denominato `codebuild-RWBCore-managed-policy`. Se si modifica il criterio, viene creata una nuova versione che viene salvata come versione predefinita. Se si esegue un'operazione successiva in AWS CodeBuild, il servizio potrebbe tentare di aggiornare la policy. In tal caso, viene visualizzato il seguente errore:

```
codebuild.amazon.com did not create the default version (V2) of the
codebuild-RWBCore-managed-policy policy that is attached to the codebuild-
RWBCore-service-role role. To continue, detach the policy from any other
identities and then delete the policy and the role.
```

Se viene visualizzato questo errore, dovrai apportare le modifiche in IAM prima di poter continuare con l'operazione di servizio. Innanzitutto, impostare la versione predefinita del criterio su V1 e riprovare l'operazione. Se V1 è stato eliminato in precedenza o se la scelta di V1 non funziona, pulire ed eliminare il criterio e il ruolo esistenti.

Per ulteriori informazioni sulla modifica dei criteri gestiti, vedere Modifica di policy gestite dal cliente (console). Per ulteriori informazioni sulle versioni dei criteri, consulta Controllo delle versioni delle policy IAM.

Per eliminare un ruolo di servizio e il relativo criterio

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.

4. Scegli la scheda Entità collegate per visualizzare gli utenti, i gruppi o i ruoli IAM che utilizzano questa policy. Se una di queste identità utilizza il criterio, completare le seguenti attività:

   a. Creare un nuovo criterio gestito con le autorizzazioni necessarie. Per assicurarsi che le identità dispongano delle stesse autorizzazioni prima e dopo le azioni, copiare il documento dei criteri JSON dal criterio esistente. Crea quindi la nuova policy gestita e incolla il documento JSON come descritto in Creazione di policy utilizzando l'editor JSON.

   b. Per ogni identità interessata, allegare il nuovo criterio e quindi staccare quello precedente. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

5. Nel riquadro di navigazione, seleziona Ruoli.
6. Nell'elenco dei ruoli scegliere il nome del ruolo che si desidera eliminare.
7. Scegliere la scheda Relazioni di trust per visualizzare le entità che possono assumere il ruolo. Se è elencata un'entità diversa dal servizio, completare le seguenti attività:

   a. Creare un nuovo ruolo che si attenda a tali entità.

   b. Pertanto, devi collegare la policy creata nella fase precedente. Se questo passaggio è stato ignorato, creare subito il nuovo criterio gestito.

   c. Informare chiunque abbia assunto il ruolo che non può più farlo. Fornire loro informazioni su come assumere il nuovo ruolo e disporre delle stesse autorizzazioni.

8. Eliminare il criterio.
9. Eliminare il ruolo.

Non esiste un caso d'uso per un ruolo di servizio nella console

Alcuni servizi richiedono la creazione manuale di un ruolo di servizio per concedere al servizio autorizzazioni per eseguire operazioni per conto dell'utente. Se il servizio non è elencato nella console IAM, dovrai elencarlo manualmente come principale attendibile. Se la documentazione relativa al servizio o alla funzionalità in uso non include istruzioni per elencare il servizio come principale attendibile, fornisci un feedback sulla pagina.

Per creare manualmente un ruolo di servizio, è necessario conoscere il principale del servizio per il servizio che assumerà il ruolo. Un'entità servizio è un identificatore che viene utilizzato per concedere autorizzazioni a un servizio. Il principale del servizio è definito dal servizio.

È possibile trovare il principale del servizio per alcuni servizi con la seguente procedura:

1. Aprire AWS servizi supportati da IAM.

2. Controlla se il servizio ha Sì nella colonna Service-linked dei ruoli.
3. Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.
4. Trova la sezione delle autorizzazioni dei Service-linked ruoli per quel servizio per visualizzare l'entità principale del servizio.

È possibile creare manualmente un ruolo del servizio utilizzando i comandi della AWS CLI o le operazioni delle API AWS. Per creare manualmente un ruolo di servizio utilizzando la console IAM, completa le seguenti attività:

1. Crea un ruolo IAM utilizzando il tuo ID account. Non allegare una policy o concedere autorizzazioni. Per informazioni dettagliate, vedi Creazione di un ruolo per fornire le autorizzazioni a un utente IAM.
2. Apri il ruolo e modificare la relazione di attendibilità. Invece di fidarsi dell'account, il ruolo deve considerare attendibile il servizio. Ad esempio, aggiorna il seguente elemento `Principal`:

   ```
   "Principal": { "AWS": "arn:aws:iam::123456789012:root" }
   ```

   Cambia il valore del principale per il servizio, ad esempio IAM.

   ```
   "Principal": { "Service": "iam.amazonaws.com" }
   ```
3. Aggiungi le autorizzazioni richieste dal servizio allegando le policy di autorizzazione al ruolo.
4. Torna al servizio che richiede le autorizzazioni e utilizza il metodo documentato per notificare al servizio il nuovo ruolo di servizio.

Risoluzione dei problemi relativi a IAM e Amazon EC2

Le informazioni seguenti possono essere utili per risolvere i problemi relativi a IAM con Amazon EC2.

Argomenti

- Durante l'avvio di un'istanza, il ruolo non viene visualizzato nell'elenco Ruolo IAM nella console Amazon EC2.
- Le credenziali per l'istanza si riferiscono al ruolo errato
- Quando tento di chiamare AddRoleToInstanceProfile, viene visualizzato un errore AccessDenied

- Amazon EC2: quando provo ad avviare un'istanza con un ruolo, ricevo un errore AccessDenied
- Non è possibile accedere alle credenziali di sicurezza temporanee nell'istanza EC2
- Cosa significano gli errori riportati nel documento info nella sottostruttura IAM?

Durante l'avvio di un'istanza, il ruolo non viene visualizzato nell'elenco Ruolo IAM nella console Amazon EC2.

Verifica quanto segue:

- Se si è effettuato l'accesso come utente IAM, verificare di disporre dell'autorizzazione a chiamare `ListInstanceProfiles`. Per ulteriori informazioni sulle autorizzazioni necessarie per gestire i ruoli, consulta Autorizzazioni richieste per l'utilizzo dei ruoli con Amazon EC2. Per informazioni sull'aggiunta di autorizzazioni a un utente, consultare Gestire le policy IAM.

  Se non puoi modificare le tue autorizzazioni, contatta un amministratore che possa utilizzare IAM per aggiornare le autorizzazioni.
- Se è stato creato un ruolo utilizzando la CLI o l'API di IAM, verifica che:
  - Sia stato creato un profilo dell'istanza e che sia stato aggiunto a tale profilo.
  - Sia stato usato lo stesso nome per il ruolo e il profilo dell'istanza. Inoltre, se il ruolo e il profilo dell'istanza sono stati chiamati in modo diverso, nella console Amazon EC2 non verrà visualizzato il nome del ruolo corretto.

  L'elenco Ruolo IAM nella console Amazon EC2 riporta i nomi dei profili dell'istanza, non i nomi dei ruoli. Sarà necessario selezionare il nome del profilo dell'istanza che contiene il ruolo desiderato. Per ulteriori informazioni sui profili dell'istanza, consultare Usare profili dell'istanza.

> **Note**
>
> Se utilizzi la console IAM per creare ruoli, non è necessario lavorare con i profili dell'istanza. Per ogni ruolo creato nella console IAM viene creato un profilo dell'istanza con lo stesso nome del ruolo e il ruolo viene automaticamente aggiunto a tale profilo. Un profilo di istanza può contenere un solo ruolo IAM e tale limite non può essere aumentato.

Le credenziali per l'istanza si riferiscono al ruolo errato

Il ruolo nel profilo dell'istanza potrebbe essere stato sostituito di recente. In questo caso, l'applicazione deve attendere la prossima rotazione delle credenziali pianificata automaticamente prima che le credenziali per il ruolo diventino disponibili.

Per forzare la modifica, devi dissociare il profilo dell'istanza e quindi associare il profilo dell'istanza oppure arrestare l'istanza e riavviarla.

Quando tento di chiamare **`AddRoleToInstanceProfile`**, viene visualizzato un errore **`AccessDenied`**

Se stai effettuando richieste come utente IAM, verifica di disporre delle autorizzazioni seguenti:

- `iam:AddRoleToInstanceProfile` con la risorsa corrispondente all'ARN del profilo dell'istanza (ad esempio, `arn:aws:iam::999999999999:instance-profile/ExampleInstanceProfile`).

Per ulteriori informazioni sulle autorizzazioni necessarie per gestire i ruoli, consulta Come si inizia?. Per informazioni sull'aggiunta di autorizzazioni a un utente, consultare Gestire le policy IAM.

Amazon EC2: quando provo ad avviare un'istanza con un ruolo, ricevo un errore **`AccessDenied`**

Verifica quanto segue:

- Avviare un'istanza senza un profilo dell'istanza. In questo modo il problema sarà limitato ai ruoli IAM per le istanze Amazon EC2.
- Se stai effettuando richieste come utente IAM, verifica di disporre delle autorizzazioni seguenti:
  - `ec2:RunInstances` con una risorsa jolly ("*")
  - `iam:PassRole` con la risorsa corrispondente all'ARN del ruolo (ad esempio, `arn:aws:iam::999999999999:role/ExampleRoleName`)
- Chiama l'operazione `GetInstanceProfile` IAM per assicurarti di stare utilizzando un profilo dell'istanza valido o un ARN del profilo di istanza valido. Per ulteriori informazioni, consulta Utilizzo dei ruoli IAM con le istanze Amazon EC2.

- Chiama l'operazione `GetInstanceProfile` IAM per assicurarti che il profilo di istanza disponga di un ruolo. I profili dell'istanza vuoti genereranno un errore `AccessDenied`. Per ulteriori informazioni sulla creazione di un ruolo, consultare Creazione di ruoli IAM.

Per ulteriori informazioni sulle autorizzazioni necessarie per gestire i ruoli, consulta Come si inizia?. Per informazioni sull'aggiunta di autorizzazioni a un utente, consultare Gestire le policy IAM.

Non è possibile accedere alle credenziali di sicurezza temporanee nell'istanza EC2

Per accedere alle credenziali di sicurezza temporanee nell'istanza EC2, è necessario innanzitutto utilizzare la console IAM per creare un ruolo. Quindi, avviare un'istanza EC2 che utilizza tale ruolo ed esaminare l'istanza in esecuzione. Per ulteriori informazioni, consulta How Do I Get Started? in Utilizzare un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2.

Se non è ancora possibile accedere alle credenziali di sicurezza temporanee sull'istanza EC2, verificare quanto segue:

- È possibile accedere a un'altra parte di Instance Metadata Service (IMDS)? In caso contrario, verificare che non vi siano regole del firewall che bloccano l'accesso alle richieste a IMDS.

```
[ec2-user@domU-12-31-39-0A-8D-DE ~]$ GET http://169.254.169.254/latest/meta-data/
hostname; echo
```

- La sottostruttura `iam` di IMDS esiste? In caso contrario, verifica che all'istanza sia associato un profilo dell'istanza IAM chiamando l'operazione API `DescribeInstances` di EC2 o utilizzando il comando della CLI `aws ec2 describe-instances`.

```
[ec2-user@domU-12-31-39-0A-8D-DE ~]$ GET http://169.254.169.254/latest/meta-data/iam;
 echo
```

- Verifica la presenza di un errore nel documento `info` nella sottostruttura IAM. Se è presente un errore, consultare Cosa significano gli errori riportati nel documento `info` nella sottostruttura IAM? per ulteriori informazioni.

```
[ec2-user@domU-12-31-39-0A-8D-DE ~]$ GET http://169.254.169.254/latest/meta-data/iam/
info; echo
```

Cosa significano gli errori riportati nel documento **`info`** nella sottostruttura IAM?

Il documento **`iam/info`** indica **`"Code":"InstanceProfileNotFound"`**

Il profilo dell'istanza IAM è stato eliminato e Amazon EC2 non è più in grado di fornire le credenziali all'istanza. È necessario collegare un profilo dell'istanza valido all'istanza Amazon EC2.

Se esiste un profilo dell'istanza con il nome specificato, controllare che tale profilo non sia stato eliminato e che ne sia stato creato un altro con lo stesso nome:

1. Chiama l'operazione `GetInstanceProfile` IAM per ottenere `InstanceProfileId`.
2. Chiama l'operazione `DescribeInstances` di Amazon EC2 per ottenere il valore `IamInstanceProfileId` per l'istanza.
3. Verifica che il `InstanceProfileId` ottenuto dall'operazione IAM corrisponda al `IamInstanceProfileId` ottenuto dall'operazione Amazon EC2.

Se IDs sono diversi, il profilo di istanza associato alle istanze non è più valido. È necessario collegare un profilo dell'istanza valido all'istanza.

Il documento **`iam/info`** indica un esito positivo, ma indica anche **`"Message":"Instance Profile does not contain a role..."`**

Il ruolo è stato rimosso dal profilo dell'istanza dall'operazione `RemoveRoleFromInstanceProfile` IAM. È possibile utilizzare l'operazione `AddRoleToInstanceProfile` IAM per collegare un ruolo al profilo dell'istanza. L'applicazione dovrà attendere il successivo aggiornamento pianificato per accedere alle credenziali del ruolo.

Per forzare la modifica, devi dissociare il profilo dell'istanza e quindi associare il profilo dell'istanza oppure arrestare l'istanza e riavviarla.

Il documento **`iam/security-credentials/[role-name]`** indica **`"Code":"AssumeRoleUnauthorizedAccess"`**

Amazon EC2 non dispone dell'autorizzazione per assumere il ruolo. L'autorizzazione ad assumere il ruolo è determinata dalla policy di affidabilità collegata al ruolo, come nell'esempio che segue. Utilizza l'API `UpdateAssumeRolePolicy` IAM per aggiornare la policy di attendibilità.

JSON

```
{"Version":"2012-10-17",      "Statement": [{"Effect": "Allow","Principal":
 {"Service": ["ec2.amazonaws.com"]},"Action": ["sts:AssumeRole"]}]}
```

L'applicazione dovrà attendere il successivo aggiornamento pianificato automaticamente per accedere alle credenziali del ruolo.

Per forzare la modifica, devi dissociare il profilo dell'istanza e quindi associare il profilo dell'istanza oppure arrestare l'istanza e riavviarla.

Risoluzione dei problemi relativi a IAM ed Amazon S3

Utilizza le informazioni contenute in questa pagina per eseguire la diagnosi e risolvere i problemi che possono verificarsi durante l'utilizzo di Amazon S3 e IAM.

Come posso concedere l'accesso anonimo a un bucket Amazon S3?

È possibile utilizzare una policy del bucket Amazon S3 che specifica un carattere jolly (*) nell'elemento `principal`, il che significa che chiunque è in grado di accedere al bucket. Con l'accesso anonimo, chiunque (compresi gli utenti senza un Account AWS) sarà in grado di accedere al bucket. Per una policy di esempio, consulta Esempi di policy di bucket Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

Ho effettuato l'accesso come Account AWS utente root. Perché non riesco ad accedere a un bucket Amazon S3 con il mio account?

In alcuni casi, è possibile avere un utente IAM con accesso completo a IAM e Amazon S3. Se l'utente IAM assegna una policy del bucket a un bucket Amazon S3 e non specifica l'utente root come principale, all'utente root verrà negato l'accesso al bucket. Tuttavia, come utente root, puoi comunque accedere al bucket. A tale scopo, modifica la policy del bucket per consentire l'accesso dell'utente root dalla console Amazon S3 o la AWS CLI. Usa il seguente principal, sostituendolo *123456789012* con l'ID di Account AWS.

```
"Principal": { "AWS": "arn:aws:iam::123456789012:root" }
```

Risoluzione dei problemi di federazione SAML con IAM

Utilizza le informazioni contenute qui per diagnosticare e risolvere i problemi che puoi incontrare durante l'utilizzo di SAML 2.0 e la federazione con AWS Identity and Access Management.

Argomenti

- Errore: La tua richiesta include una risposta SAML non valida. Per disconnetterti, fai clic qui.
- Errore: RoleSessionName è obbligatorio in AuthnResponse (servizio: AWSSecurityTokenService; codice di stato: 400; codice di errore: InvalidIdentityToken)
- Errore: non autorizzato a eseguire sts: AssumeRoleWithSAML (service: AWSSecurityTokenService; codice di stato: 403; codice di errore: AccessDenied)
- Errore: AuthnResponse deve corrispondere RoleSessionName a [a-z A-Z _0-9+=, .@-] {2,64} (servizio: AWSSecurityTokenService; codice di stato: 400; codice di errore:) InvalidIdentityToken
- Errore: l'identità di origine deve corrispondere a [a-z A-Z _0-9+=, .@-] {2,256} e non iniziare con «aws:» (service:; codice di stato: 400 AWSSecurityTokenService; codice di errore:) InvalidIdentityToken
- Errore: firma di risposta non valida (servizio: AWSSecurityTokenService; codice di stato: 400; codice di errore: InvalidIdentityToken)
- Errore: chiave privata non valida.
- Errore: impossibile rimuovere la chiave privata.
- Errore: impossibile rimuovere la chiave privata perché l'ID della chiave non corrisponde a una chiave privata.
- Errore: impossibile assumere il ruolo: emittente non presente nel provider specificato (servizio:; codice di stato: 400 AWSOpenIdDiscoveryService; codice di errore:) AuthSamlInvalidSamlResponseException
- Errore: Impossibile analizzare i metadati.
- Errore: impossibile aggiornare i provider di identità. Non è stato definito alcun aggiornamento per i metadati o l'asserzione di crittografia.
- Errore: impossibile impostare la modalità di crittografia delle asserzioni su Richiesta perché non è stata fornita alcuna chiave privata.
- Errore: impossibile aggiungere e rimuovere chiavi private nella stessa richiesta. Imposta un valore solo per uno dei due parametri.
- Errore: Il provider specificato non esiste.

- Errore: DurationSeconds i valori richiesti sono superiori a quelli MaxSessionDuration impostati per questo ruolo.
- Errore: è stato raggiunto il limite di 2 per la chiave privata.
- Errore: la risposta non contiene il pubblico richiesto.

Errore: La tua richiesta include una risposta SAML non valida. Per disconnetterti, fai clic qui.

Questo errore può accadere quando la risposta SAML dall'identità del fornitore non include un attributo con `Name` impostato su `https://aws.amazon.com/SAML/Attributes/Role` L'attributo deve contenere uno o più elementi `AttributeValue`, ognuno contenente un paio di stringhe separate dalla virgola:

- L'ARN di un ruolo su cui l'utente può essere mappato
- L'ARN del fornitore SAML

L'errore può verificarsi anche quando i valori degli attributi SAML inviati dal provider di identità hanno uno spazio bianco iniziale o finale, o altri caratteri non validi nei valori degli attributi SAML. Per ulteriori informazioni sui valori previsti per gli attributi SAML, consulta Configurare le asserzioni SAML per la risposta di autenticazione

Per ulteriori informazioni, consulta Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Visualizzare una risposta SAML nel browser.

Errore: RoleSessionName è obbligatorio in AuthnResponse (servizio: AWSSecurityTokenService; codice di stato: 400; codice di errore: InvalidIdentityToken)

Questo errore può accadere quando la risposta SAML dall'identità del fornitore non include un attributo con `Name` impostato su `https://aws.amazon.com/SAML/Attributes/RoleSessionName` Il valore dell'attributo è un identificatore per l'utente e in genere è un ID utente o un indirizzo e-mail.

Per ulteriori informazioni, consulta Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Visualizzare una risposta SAML nel browser.

Errore: non autorizzato a eseguire sts: AssumeRoleWithSAML (service: AWSSecurityTokenService; codice di stato: 403; codice di errore: AccessDenied)

Questo errore può verificarsi se il ruolo IAM specificato nella risposta SAML è errato o non esiste. Assicurati di utilizzare il nome esatto del ruolo in quanto i nomi prevedono una distinzione tra lettere maiuscole e minuscole. Correggere il nome del ruolo nella configurazione del provider di servizi SAML.

L'accesso è consentito solo se il criterio di attendibilità del ruolo include l'azione `sts:AssumeRoleWithSAML`. Se l'asserzione SAML è configurata per utilizzare l'attributo `PrincipalTag`, i criteri di attendibilità devono includere anche l'azione `sts:TagSession`. Per ulteriori informazioni sui tag di sessione, consultare Passa i tag della sessione in AWS STS.

Questo errore può verificarsi se non disponi delle autorizzazioni `sts:SetSourceIdentity` nella policy di attendibilità del ruolo. Se l'asserzione SAML è configurata per utilizzare l'attributo `SourceIdentity`, le policy di attendibilità devono includere anche l'azione `sts:SetSourceIdentity`. Per ulteriori informazioni sull'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

Questo errore può verificarsi anche se un principale federato non ha le autorizzazioni per assumere il ruolo. Il ruolo deve avere una policy di affidabilità che specifica l'ARN del provider d'identità SAML IAM come il `Principal`. Il ruolo contiene anche le condizioni che controllano quali utenti possono assumere il ruolo. Verifica che gli utenti soddisfino i requisiti delle condizioni.

Questo errore può verificarsi anche se la risposta SAML non include un `Subject` contenente un `NameID`.

Per ulteriori informazioni, consulta Consentire ai principali federati SAML 2.0 di accedere a Console di gestione AWS e Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Visualizzare una risposta SAML nel browser.

Errore: AuthnResponse deve corrispondere RoleSessionName a [a-z A-Z _0-9+=, .@-] {2,64} (servizio: AWSSecurityTokenService; codice di stato: 400; codice di errore:) InvalidIdentityToken

Questo errore può verificarsi se il valore dell'attributo `RoleSessionName` è troppo lungo o contiene caratteri non validi. La lunghezza massima valida è 64 caratteri;

Per ulteriori informazioni, consulta Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Visualizzare una risposta SAML nel browser.

Errore: l'identità di origine deve corrispondere a [a-z A-Z _0-9+=, .@-] {2,256} e non iniziare con **«aws:»** (service:; codice di stato: 400 AWSSecurityTokenService; codice di errore:) InvalidIdentityToken

Questo errore può verificarsi se il valore dell'attributo `sourceIdentity` è troppo lungo o contiene caratteri non validi. La lunghezza massima valida è di 256 caratteri. Per ulteriori informazioni sull'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

Per ulteriori informazioni sulla creazione di asserzioni SAML, consulta Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Visualizzare una risposta SAML nel browser.

Errore: firma di risposta non valida (servizio: AWSSecurityTokenService; codice di stato: 400; codice di errore: InvalidIdentityToken)

Questo errore può verificarsi quando i metadati di federazione del provider di identità non soddisfano i metadati del provider di identità IAM. Ad esempio, il file dei metadati per il provider del servizio di identità potrebbe essere cambiato per aggiornare un certificato scaduto. Scaricare il file di metadati SAML aggiornato dal fornitore del servizio di identità. Quindi aggiornalo nell'entità del provider di AWS identità definita in IAM con il comando CLI `aws iam update-saml-provider` multipiattaforma o il cmdlet. `Update-IAMSAMLProvider` PowerShell

Errore: chiave privata non valida.

Questo errore può verificarsi se il file della chiave privata non è formattato correttamente. Questo errore può fornire ulteriori dettagli sul motivo per cui la chiave privata non è valida:

- La chiave è crittografata.
- Il formato della chiave non è stato riconosciuto. Il file della chiave privata deve essere un file .pem.

Quando Creare un provider di identità SAML in IAM accedi Console di gestione AWS, devi scaricare la chiave privata dal tuo provider di identità da fornire a IAM per abilitare la crittografia. La chiave privata deve essere un file .pem che utilizza AES-GCM il nostro algoritmo di AES-CBC crittografia per decrittografare le asserzioni SAML.

Errore: impossibile rimuovere la chiave privata.

Questo errore può verificarsi quando la crittografia SAML è impostata su Richiesta e la tua richiesta rimuoverebbe l'unica chiave di decrittografia privata per il provider IAM SAML. Per ulteriori informazioni sulla rotazione delle chiavi private, consulta Gestire le chiavi di crittografia SAML.

Errore: impossibile rimuovere la chiave privata perché l'ID della chiave non corrisponde a una chiave privata.

Questo errore può verificarsi se il valore `keyId` della chiave privata non corrisponde a nessuno dei due ID chiave dei file di chiave privata del provider di identità.

Quando utilizzi le operazioni API update-saml-provider o UpdateSAMLProvider per rimuovere le chiavi private di crittografia SAML, il valore in `RemovePrivateKey` deve essere un ID chiave valido per una chiave privata collegata al tuo provider di identità.

Errore: impossibile assumere il ruolo: emittente non presente nel provider specificato (servizio:; codice di stato: 400 AWSOpenIdDiscoveryService; codice di errore:) AuthSamlInvalidSamlResponseException

Questo errore può verificarsi se l'approvatore nella risposta SAML non corrisponde all'approvatore dichiarato nel file dei metadati di federazione Il file di metadati è stato caricato AWS quando hai creato il provider di identità in IAM.

Errore: Impossibile analizzare i metadati.

Questo errore può verificarsi se il file dei metadati non è formattato correttamente.

Quando crei o gestisci un provider di identità SAML in Console di gestione AWS, devi recuperare il documento di metadati SAML dal tuo provider di identità.

Questo file di metadati include il nome dell'approvatore, le informazioni sulla scadenza e le chiavi che possono essere utilizzate per convalidare la risposta di autenticazione SAML (asserzioni) ricevute dal provider di identità. Il file di metadati deve essere codificato in un UTF-8 formato senza un byte order mark (BOM). Per rimuovere la distinta base, puoi codificare il file UTF-8 utilizzando uno strumento di modifica del testo, come Notepad++.

Il X.509 certificato incluso nel documento di metadati SAML deve utilizzare una dimensione della chiave di almeno 1024 bit. Inoltre, il X.509 certificato deve essere privo di estensioni ripetute. È possibile utilizzare le estensioni, ma possono essere visualizzate una sola volta nel certificato. Se il X.509 certificato non soddisfa nessuna delle due condizioni, la creazione dell'IdP non riesce e restituisce un errore «Impossibile analizzare i metadati».

Come definito dal SAML V2.0 Metadata Interoperability Profile Version 1.0, IAM non valuta né interviene sulla scadenza dei certificati nei documenti di metadati SAML. X.509 Se sei preoccupato per i X.509 certificati scaduti, ti consigliamo di monitorare le date di scadenza dei certificati e di ruotare i certificati in base alle politiche di governance e sicurezza della tua organizzazione.

Errore: impossibile aggiornare i provider di identità. Non è stato definito alcun aggiornamento per i metadati o l'asserzione di crittografia.

Questo errore può verificarsi se utilizzi le operazioni di `update-saml-provider` CLI o l'API `UpdateSAMLProvider`, ma non fornisci valori di aggiornamento nei parametri della richiesta. Per ulteriori informazioni sull'aggiornamento del provider IAM SAML, consulta Creare un provider di identità SAML in IAM.

Errore: impossibile impostare la modalità di crittografia delle asserzioni su Richiesta perché non è stata fornita alcuna chiave privata.

Questo errore può verificarsi quando non hai precedentemente caricato una chiave di decrittografia privata e provi a impostare la crittografia SAML su Richiesta senza includere una chiave privata nella richiesta.

Assicurati che sia definita una chiave privata per il tuo provider IAM SAML quando utilizzi le operazioni di `create-saml-provider` CLI, l'API `CreateSAMLProvider`, `update-saml-provider` CLI o l'API `UpdateSAMLProvider` per richiedere asserzioni SAML crittografate.

Errore: impossibile aggiungere e rimuovere chiavi private nella stessa richiesta. Imposta un valore solo per uno dei due parametri.

Questo errore può verificarsi se entrambi i valori di aggiunta e rimozione della chiave privata sono inclusi nella stessa richiesta.

Quando utilizzi le operazioni API update-saml-provider o UpdateSAMLProvider per ruotare i file delle chiavi private di crittografia SAML, puoi solo aggiungere o rimuovere una chiave privata nella richiesta. Se aggiungi una chiave privata mentre rimuovi una chiave privata, l'operazione non riesce. Per ulteriori informazioni sulla rotazione delle chiavi private, consulta Gestire le chiavi di crittografia SAML.

Errore: Il provider specificato non esiste.

Questo errore può verificarsi se il nome del provider specificato nell'asserzione SAML non corrisponde al nome del provider in IAM. Per ulteriori informazioni sulla visualizzazione del nome del provider, consulta Creare un provider di identità SAML in IAM.

Errore: DurationSeconds i valori richiesti sono superiori a quelli MaxSessionDuration impostati per questo ruolo.

Questo errore può verificarsi se assumi un ruolo dall'API AWS CLI or.

Quando utilizzi la CLI o le operazioni AssumeRoleWithSAML API assume-role-with-saml per assumere un ruolo, puoi specificare un valore per il parametro. `DurationSeconds` Puoi specificare un valore da 900 secondi (15 minuti) fino alla durata massima impostata per la sessione per il ruolo. Se si specifica un valore superiore a questa impostazione, l'operazione ha esito negativo. Ad esempio, se si specifica una durata di sessione di 12 ore, ma l'amministratore ha impostato la durata massima di sessione a 6 ore, l'operazione ha esito negativo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Aggiornamento della durata massima della sessione per un ruolo.

Errore: è stato raggiunto il limite di 2 per la chiave privata.

Questo errore può verificarsi se si prova ad aggiungere una chiave privata al proprio provider di identità.

Puoi salvare un massimo di due chiavi private per ogni provider di identità. Quando utilizzi le operazioni API update-saml-provider o UpdateSAMLProvider per aggiungere una terza chiave privata, l'operazione non riesce.

Rimuovi le chiavi private scadute prima di aggiungere una nuova chiave privata. Per ulteriori informazioni sulla rotazione delle chiavi private, consulta Gestire le chiavi di crittografia SAML.

Errore: la risposta non contiene il pubblico richiesto.

Questo errore può verificarsi in caso di mancata corrispondenza tra l'URL del pubblico e il provider di identità nella configurazione SAML. Assicurati che l'identificativo del soggetto che si basa sul gestore dell'identità digitale corrisponda esattamente all'URL del pubblico (ID entità) fornito nella configurazione SAML.

Come IAM interagisce con altri AWS servizi

Sebbene IAM sia il AWS servizio principale che utilizzerai per gestire le risorse IAM, tutti gli altri AWS servizi funzionano con IAM per controllare l'accesso alle risorse del tuo account.

- CloudFormation

  CloudFormation si integra con IAM consentendoti di definire e gestire le risorse IAM come parte dei tuoi CloudFormation modelli. Puoi utilizzarlo CloudFormation per specificare le autorizzazioni IAM necessarie per le altre AWS risorse che fornisci. CloudFormation supporta anche l'uso dei ruoli IAM per gestire le credenziali necessarie per il provisioning e la gestione AWS dell'infrastruttura, e la funzionalità di rilevamento delle deviazioni consente di mantenere l'integrità delle configurazioni IAM.
- AWS CloudShell

  Quando accedi AWS CloudShell, l'autenticazione e l'autorizzazione vengono gestite tramite IAM. AWS CloudShell viene eseguito nel contesto di un ruolo IAM assegnato al tuo utente o account. All'avvio AWS CloudShell, genera automaticamente credenziali di sicurezza temporanee in base al ruolo IAM che ti è stato assegnato.
- AWS SDKs

  AWS SDKs Lavoriamo con IAM gestendo il processo di autenticazione e autorizzazione, gestendo AWS le credenziali e rispettando le autorizzazioni e le politiche definite in IAM per garantire che l'applicazione possa accedere solo alle risorse che è autorizzata a utilizzare. SDKs Forniscono meccanismi per ottenere e utilizzare credenziali di sicurezza temporanee, oltre a convalidare le autorizzazioni necessarie per le operazioni dell'applicazione.

Per un elenco dei AWS servizi che funzionano con IAM e le funzionalità IAM che supportano i servizi, consulta. AWS servizi supportati da IAM

Crea risorse IAM con CloudFormation

AWS Identity and Access Management è integrato con AWS CloudFormation, un servizio che consente di modellare e configurare le AWS risorse in modo da dedicare meno tempo alla creazione e alla gestione delle risorse e dell'infrastruttura. Crei un modello che descrive tutte le AWS risorse che desideri (come chiavi di accesso, gruppi, politiche di gruppo, profili di istanza, politiche gestite, provider OIDC, politiche in linea, ruoli, politiche di ruolo, provider SAML, certificati server, ruoli

collegati ai servizi, utenti (e aggiunta di utenti ai gruppi), politiche utente e dispositivi MFA virtuali) e fornisce e CloudFormation configura tali risorse per te.

Quando lo utilizzi CloudFormation, puoi riutilizzare il modello per configurare le risorse IAM in modo coerente e ripetuto. Descrivi le tue risorse una sola volta, quindi fornisci le stesse risorse più e più volte in più Account AWS regioni.

IAM e CloudFormation modelli

Per assegnare e configurare le risorse per IAM e i servizi correlati, devi conoscere i modelli CloudFormation. I modelli sono file di testo formattati in JSON o YAML. Questi modelli descrivono le risorse che desideri fornire nei tuoi CloudFormation stack. Se non conosci JSON o YAML, puoi usare CloudFormation Designer per iniziare a usare i modelli. CloudFormation Per ulteriori informazioni, consulta Che cos'è CloudFormation Designer? nella Guida per l'utente di AWS CloudFormation .

IAM supporta la creazione di chiavi di accesso, gruppi, policy di gruppo, profili di istanza, policy gestite, provider OIDC, policy in linea, ruoli, policy di ruolo, provider SAML, certificati server, ruoli collegati ai servizi, utenti (e aggiunta di utenti ai gruppi), policy utente e dispositivi MFA virtuali in. CloudFormation Per ulteriori informazioni, inclusi esempi di modelli JSON e YAML per le risorse IAM, consulta Riferimento al tipo di risorsa AWS Identity and Access Management nella Guida per l'utente di AWS CloudFormation .

Puoi anche creare modelli che creano risorse correlate, come ruoli e policy gestite.

Scopri di più su CloudFormation

Per ulteriori informazioni CloudFormation, consulta le seguenti risorse:

- AWS CloudFormation
- AWS CloudFormation Guida per l'utente
- CloudFormation Documentazione di riferimento delle API
- AWS CloudFormation Guida per l'utente dell'interfaccia a riga di comando

Utilizzo AWS CloudShell con cui lavorare AWS Identity and Access Management

AWS CloudShell è una shell preautenticata basata su browser che puoi avviare direttamente da. Console di gestione AWS Puoi eseguire AWS CLI comandi AWS sui servizi (inclusa la gestione delle

AWS identità e degli accessi) utilizzando la tua shell preferita (Bash o Z shell). PowerShell E puoi farlo senza dover scaricare o installare strumenti da riga di comando.

Si avvia AWS CloudShell da e Console di gestione AWS le AWS credenziali utilizzate per accedere alla console sono automaticamente disponibili in una nuova sessione di shell. Questa preautenticazione degli AWS CloudShell utenti consente di saltare la configurazione delle credenziali quando si interagisce con AWS servizi come IAM utilizzando la AWS CLI versione 2 (preinstallata nell'ambiente di elaborazione della shell).

Ottieni le autorizzazioni IAM per AWS CloudShell

Utilizzando le risorse di gestione degli accessi fornite da AWS Identity and Access Management, gli amministratori possono concedere autorizzazioni agli utenti IAM in modo che possano accedere AWS CloudShell e utilizzare le funzionalità dell'ambiente.

Il modo più rapido per un amministratore di concedere l'accesso agli utenti è tramite una AWS policy gestita. Una policy gestita da AWS è una policy autonoma che viene creata e amministrata da AWS. La seguente policy AWS gestita per CloudShell può essere associata alle identità IAM:

- `AWSCloudShellFullAccess`: concede il permesso di utilizzo AWS CloudShell con accesso completo a tutte le funzionalità.

Se desideri limitare l'ambito delle azioni che un utente IAM può eseguire AWS CloudShell, puoi creare una policy personalizzata che utilizzi la policy `AWSCloudShellFullAccess` gestita come modello. Per ulteriori informazioni sulla limitazione delle azioni disponibili per gli utenti in CloudShell, consulta Gestire AWS CloudShell l'accesso e l'utilizzo con le politiche IAM nella Guida per l'AWS CloudShell utente.

Interagire con IAM

Dopo l'avvio AWS CloudShell da Console di gestione AWS, puoi iniziare immediatamente a interagire con IAM utilizzando l'interfaccia a riga di comando.

> **Note**
>
> Quando si utilizza AWS CLI in AWS CloudShell, non è necessario scaricare o installare risorse aggiuntive. Inoltre, poiché hai già eseguito l'autenticazione alla shell, non è necessario configurare le credenziali prima di effettuare chiamate.

Crea un gruppo IAM e aggiungi un utente IAM al gruppo utilizzando AWS CloudShell

L'esempio seguente viene utilizzato CloudShell per creare un gruppo IAM, aggiungere un utente IAM al gruppo e quindi verificare che il comando sia riuscito.

1. Da Console di gestione AWS, puoi avviarlo CloudShell scegliendo le seguenti opzioni disponibili sulla barra di navigazione:

   - Scegli l' CloudShell icona.
   - Inizia a digitare «cloudshell» nella casella di ricerca, quindi scegli l'opzione. CloudShell
2. Per creare un gruppo IAM, inserisci il seguente comando nella riga di comando. CloudShell In questo esempio abbiamo denominato il gruppo east_coast:

   ```
   aws iam create-group --group-name east_coast
   ```

   Se la chiamata ha esito positivo, la riga di comando visualizza una risposta del servizio simile al seguente output:

   ```
   {
      "Group": {
          "Path": "/",
          "GroupName": "east_coast",
          "GroupId": "AGPAYBDBW4JBY3EXAMPLE",
          "Arn": "arn:aws:iam::111122223333:group/east_coast",
          "CreateDate": "2023-09-11T21:02:21+00:00"
       }
   }
   ```
3. Per aggiungere un utente al gruppo creato, utilizza il seguente comando, specificando il nome e il nome utente del gruppo. In questo esempio abbiamo denominato il gruppo east_coast e l'utente john:

   ```
   aws iam add-user-to-group --group-name east_coast --user-name john
   ```
4. Per verificare che l'utente sia nel gruppo, utilizza il seguente comando, specificando il nome del gruppo. In questo esempio continuiamo a utilizzare il gruppo east_coast:

   ```
   aws iam get-group --group-name east_coast
   ```

Se la chiamata ha esito positivo, la riga di comando visualizza una risposta del servizio simile al seguente output:


```
{
  "Users": [
    {
        "Path": "/",
        "UserName": "john",
        "UserId": "AIDAYBDBW4JBXGEXAMPLE",
        "Arn": "arn:aws:iam::552108220995:user/john",
        "CreateDate": "2023-09-11T20:43:14+00:00",
        "PasswordLastUsed": "2023-09-11T20:59:14+00:00"
    }
  ],
  "Group": {
        "Path": "/",
        "GroupName": "east_coast",
        "GroupId": "AGPAYBDBW4JBY3EXAMPLE",
        "Arn": "arn:aws:iam::111122223333:group/east_coast",
        "CreateDate": "2023-09-11T21:02:21+00:00"
     }
 }
```


Utilizzo di questo servizio con un AWS SDK

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK per C++	AWS SDK per C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice

Documentazione sugli SDK	Esempi di codice
AWS SDK per Java	AWS SDK per Java esempi di codice
AWS SDK per JavaScript	AWS SDK per JavaScript esempi di codice
AWS SDK per Kotlin	AWS SDK per Kotlin esempi di codice
AWS SDK per .NET	AWS SDK per .NET esempi di codice
AWS SDK per PHP	AWS SDK per PHP esempi di codice
AWS Strumenti per PowerShell	AWS Strumenti per PowerShell esempi di codice
AWS SDK per Python v4	AWS Esempi di codice SDK per Python v4
AWS SDK per Python (Boto3)	AWS SDK per Python (Boto3) esempi di codice
AWS SDK per Ruby	AWS SDK per Ruby esempi di codice
AWS SDK per Rust	AWS SDK per Rust esempi di codice
AWS SDK per SAP ABAP	AWS SDK per SAP ABAP esempi di codice
AWS SDK per Swift	AWS SDK per Swift esempi di codice

Per esempi specifici del servizio, consulta Esempi di codice per l'utilizzo di IAM AWS SDK.

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Fornisci un feedback nella parte inferiore di questa pagina.

Informazioni di riferimento per AWS Identity and Access Management

Utilizza gli argomenti di questa sezione per trovare materiali di riferimento dettagliati su diversi aspetti di IAM e AWS STS.

Argomenti

- Identificazione AWS risorse con Amazon Resource Names (ARN)
- Identificatori IAM
- IAM e AWS STS quote
- Supporto per endpoint dual-stack
- Endpoint VPC di interfaccia
- AWS servizi supportati da IAM
- AWS Signature versione 4 per le richieste API
- Riferimento alla policy JSON IAM

- Informazioni semplificate su Servizio AWS per l'accesso programmatico

Identificazione AWS risorse con Amazon Resource Names (ARN)

Gli Amazon Resource Names (ARN) identificano in modo univoco le risorse. AWS Abbiamo bisogno di un ARN quando è necessario specificare una risorsa in modo inequivocabile in tutte le aree AWS, ad esempio nelle policy IAM, nei tag Amazon Relational Database Service (Amazon RDS) e nelle chiamate API. Sebbene gli ARN, come qualsiasi informazione identificativa, debbano essere utilizzati e condivisi con attenzione, non sono considerati informazioni segrete, sensibili o riservate.

Formato ARN

Di seguito sono riportati i formati generali per gli ARN. I formati specifici dipendono dalla risorsa. Per utilizzare un ARN, sostituisci il testo con le informazioni specifiche della risorsa*italicized*. Tenere presente che gli ARN per alcune risorse omettono la regione, l'ID account o la regione e l'ID account.

```
arn:partition:service:region:account-id:resource-id
arn:partition:service:region:account-id:resource-type/resource-id
```

```
arn:partition:service:region:account-id:resource-type:resource-id
```

partition

La partizione in cui si trova la risorsa. Una partizione è un gruppo di regioni. AWS Ogni AWS account è associato a una partizione.

Di seguito sono riportate le partizioni supportate:

- `aws`- Regioni AWS
- `aws-cn`: regioni Cina
- `aws-us-gov` – Regioni AWS GovCloud (US)

service

Lo spazio dei nomi del servizio che identifica il prodotto. AWS

region

Il codice della regione. Ad esempio, `us-east-2` per Stati Uniti orientali (Ohio). Per un elenco dei codici delle regioni, consulta Endpoint regionali nella Riferimenti generali di AWS.

account-id

L'ID dell' AWS account proprietario della risorsa, senza trattini. Ad esempio, `123456789012`.

resource-type

Il tipo di risorsa. Ad esempio, `vpc` per un cloud privato virtuale (VPC).

resource-id

L'identificatore di risorsa. Si tratta del nome della risorsa, dell'ID della risorsa o del percorso della risorsa. Alcuni identificatori di risorsa includono una risorsa principale (sotto-risorsa- type/parent - resource/sub -risorsa) o un qualificatore come una versione (resource-type:resource-name:qualifier).

Esempi

Utente IAM

*123456789012*arn:aws:iam: ::user/ *john*

Argomento SNS

arn:aws:sns:*us-east-1*:: *123456789012* *example-sns-topic-name*

VPC

arn:aws:ec2: *us-east-1* ::vpc/ *123456789012* *vpc-0e9801d129EXAMPLE*

Ricerca del formato dell'ARN per una risorsa

Il formato esatto di un ARN dipende dal servizio e dal tipo di risorsa. Alcuni ARN di risorse possono includere un percorso, una variabile o un carattere jolly. Per cercare il formato ARN per una AWS risorsa specifica, apri il Service Authorization Reference, apri la pagina relativa al servizio e vai alla tabella dei tipi di risorse.

Percorsi negli ARN

Alcuni ARN delle risorse possono includere un percorso. Ad esempio, in Amazon S3, l'identificatore di risorsa è un nome oggetto che può includere barre in avanti (`/`) per creare un percorso. Allo stesso modo, anche i nomi utente e i nomi di gruppo IAM possono includere percorsi. Nei percorsi IAM sono consentiti solo caratteri alfanumerici e i seguenti caratteri: barra obliqua (`/`), segno del più (`+`), segno dell'uguale (`=`), virgola (`,`), punto (`.`) chiocciola (`@`) trattino basso (`_`) e trattino (`-`).

Utilizzo di caratteri jolly nei percorsi

I percorsi possono includere un carattere jolly, vale a dire un asterisco (`*`). Alcuni elementi della policy consentono l'uso di caratteri jolly, mentre altri no. È possibile utilizzare caratteri jolly per la risorsa o NotResource gli elementi ma non per il Principal o NotPrincipal gli elementi. Per ulteriori informazioni, consulta Riferimento alla policy JSON IAM.

È possibile specificare `role/*` per indicare tutti i ruoli nell'account 123456789012, come nel seguente esempio:

```
arn:aws:iam::123456789012:role/*
```

Inoltre, è possibile terminare il nome di una risorsa con un carattere jolly. Ad esempio, puoi specificare **`service-*`** per indicare tutti i ruoli che iniziano con `service` e terminano con caratteri diversi come `service-role1` o `service-test`:

```
arn:aws:iam::123456789012:role/service-*
```

Il seguente esempio mostra gli ARN per gli oggetti in un bucket Amazon S3 in cui il nome della risorsa include un percorso. L'ARN `arn:aws:s3:::amzn-s3-demo-bucket/*` è per tutti gli

oggetti all'interno di quel bucket, indipendentemente dal prefisso. L'ARN `arn:aws:s3:::amzn-s3-demo-bucket/Development/*` è per tutti gli oggetti creati all'interno del prefisso **`/Development/`**.

Puoi anche usare il carattere jolly ? per specificare un carattere in un ARN. Ad esempio, è possibile utilizzare il seguente ARN per tutte le cartelle che iniziano con quattro caratteri e terminano in **`-test`** nel bucket S3 denominato amzn-s3-demo-bucket. Alcune cartelle che potrebbero corrispondere a questo includono `1234-test`, `2024-test` o `a100-test`.

```
arn:aws:s3:::amzn-s3-demo-bucket/????-test
```

Puoi anche utilizzare i caratteri jolly nelle diverse sezioni di un ARN, delimitati da due punti «». **`:`** Nell'esempio seguente, vengono utilizzati due caratteri jolly per abbinare tutte le applicazioni e le risorse Amazon Q all'interno delle applicazioni in tutte le Regioni per l'account 123456789012:

```
arn:aws:qbusiness:*:123456789012:*
```

Analogamente, l'esempio seguente corrisponde a tutti i VPC Amazon in tutte le Regioni per l'account 123456789012:

```
arn:aws:ec2:*:123456789012:vpc/*
```

L'esempio seguente esegue la corrispondenza di tutti i volumi Amazon EBS in tutte le Regioni per l'account 123456789012:

```
arn:aws:ec2:*:123456789012:volume/*
```

Limitazioni all'utilizzo dei caratteri jolly all'interno degli ARN

Non è possibile utilizzare un carattere jolly nella porzione di ARN che specifica il tipo di risorsa. L'ARN di esempio seguente con un carattere jolly all'interno del tipo di risorsa non è valido:

```
arn:aws:lambda:us-east-2:123456789012:functi*:my-function <== not allowed
```

> **Note**
>
> Quando si specifica un ARN incompleto (uno con meno dei sei campi standard) in una policy basata sull'identità, completa AWS automaticamente l'ARN aggiungendo caratteri

jolly (*) a tutti i campi mancanti. Ad esempio, specificare equivale `aarn:aws:sqs:*:*:*`, che consente `arn:aws:sqs` l'accesso a tutte le risorse Amazon Amazon SQS in tutte le regioni e gli account. Tuttavia, le policy di sessione trasmesse e AWS STS AssumeRole AssumeRoleWithWebIdentity le AssumeRoleWithSAML richieste non supportano gli ARN incompleti. L'utilizzo di un ARN incompleto in una policy di sessione genererà un `MalformedPolicyDocumentException` errore.

Inoltre, non è possibile utilizzare un carattere jolly nel prefisso ARN o avere un carattere jolly nella sezione delle partizioni di un ARN.

```
arn:aws:redshift:us-east-1:123456789012:? <== not allowed
```

Identificatori IAM

IAM utilizza vari identificatori per utenti, gruppi IAM, ruoli, policy e certificati del server. Questa sezione descrive gli identificatori e spiega quando vanno utilizzati.

Argomenti

- Nomi descrittivi e percorsi
- ARN IAM
- Identificatori univoci
- Approfondimento dei prefissi ID univoci
- Ottenere l'identificatore univoco

Nomi descrittivi e percorsi

Quando crei un utente, un ruolo, un gruppo di utenti o una policy o quando carichi un certificato server, gli attribuisci un nome descrittivo. Gli esempi includono Bob, TestApp 1, Developers ManageCredentialsPermissions,, o ProdServerCert.

Se utilizzi l'API IAM o AWS Command Line Interface (AWS CLI) per creare risorse IAM, puoi aggiungere un percorso opzionale. Puoi decidere di usare un solo percorso o nidificare percorsi multipli come una struttura a cartelle. Ad esempio, puoi utilizzare il percorso nidificato `/division_abc/subdivision_xyz/product_1234/engineering/` per farlo corrispondere alla struttura organizzativa della tua azienda. A quel punto, potresti ceare una policy per consentire

a tutti gli utenti del percorso di accedere all'API di simulazione policy. Per visualizzare questa policy, consulta IAM: accesso all'API simulatore di policy basata sul percorso degli utenti. Per informazioni su come specificare un nome descrittivo, vedere la documentazione dell'API utente. Per ulteriori esempi di utilizzo dei percorsi, consulta ARN IAM.

Quando si utilizza CloudFormation per creare risorse, è possibile specificare un percorso per utenti, gruppi e ruoli IAM e policy gestite dal cliente.

Se disponi di un utente e di un gruppo di utenti nello stesso percorso, IAM non inserisce automaticamente l'utente in tale gruppo di utenti. Ad esempio, potresti creare il gruppo Sviluppatori e specificare come percorso come `/division_abc/subdivision_xyz/product_1234/engineering/`. Se crei un utente denominato Bob e gli aggiungi lo stesso percorso, Bob non viene inserito automaticamente all'interno del gruppo di utenti Sviluppatori. IAM non impone alcun limite tra utenti o gruppi IAM in base ai loro percorsi. Utenti con percorsi differenti possono utilizzare le stesse risorse (supponendo che abbiano ricevuto le autorizzazioni per farlo). Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

ARN IAM

La maggior parte delle risorse dispone di nomi descrittivi: ad esempio, un utente denominato `Bob` o un gruppo di utenti denominato `Developers`. Tuttavia, il linguaggio delle policy di autorizzazione richiede di specificare la risorsa o le risorse che utilizzano il seguente formato Amazon Resource Name (ARN).

```
arn:partition:service:region:account:resource
```

Dove:

- `partition` identifica la partizione in cui si trova la risorsa. Per le Regioni AWS standard, la partizione è `aws`. Se sono presenti risorse in altre partizioni, la partizione è `aws-`*`partitionname`*. Ad esempio, la partizione per le risorse nella regione Cina (Pechino) è `aws-cn`. Non è possibile delegare l'accesso tra account in partizioni diverse.
- `service`identifica il AWS prodotto. Le risorse IAM utilizzano sempre `iam`.
- `region` identifica la regione della risorsa. Per le risorse IAM, è sempre lasciato vuoto.
- `account`specifica l' Account AWS ID senza trattini.
- `resource` identifica la risorsa specifica in base al nome.

È possibile specificare IAM e AWS STS ARNs utilizzando la seguente sintassi. La porzione di regione dell'ARN è vuota perché le risorse IAM sono globali.

Sintassi:

```
arn:aws:iam::account:root
arn:aws:iam::account:user/user-name-with-path
arn:aws:iam::account:group/group-name-with-path
arn:aws:iam::account:role/role-name-with-path
arn:aws:iam::account:policy/policy-name-with-path
arn:aws:iam::account:instance-profile/instance-profile-name-with-path
arn:aws:sts::account:federated-user/user-name
arn:aws:sts::account:assumed-role/role-name/role-session-name
arn:aws:sts::account:self
arn:aws:iam::account:mfa/virtual-device-name-with-path
arn:aws:iam::account:u2f/u2f-token-id
arn:aws:iam::account:server-certificate/certificate-name-with-path
arn:aws:iam::account:saml-provider/provider-name
arn:aws:iam::account:oidc-provider/provider-name
arn:aws:iam::aws:contextProvider/context-provider-name
```

Molti esempi riportati di seguito includono percorsi nella parte della risorsa dell'ARN. I percorsi non possono essere creati o modificati nella Console di gestione AWS. Per utilizzare i percorsi, è necessario utilizzare la risorsa utilizzando l' AWS API AWS CLI, o gli Strumenti per Windows. PowerShell

Esempi:

```
arn:aws:iam::123456789012:root
arn:aws:iam::123456789012:user/John
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/Jane
arn:aws:iam::123456789012:group/Developers
arn:aws:iam::123456789012:group/division_abc/subdivision_xyz/product_A/Developers
arn:aws:iam::123456789012:role/S3Access
arn:aws:iam::123456789012:role/application_abc/component_xyz/RDSAccess
arn:aws:iam::123456789012:role/aws-service-role/access-analyzer.amazonaws.com/
AWSServiceRoleForAccessAnalyzer
arn:aws:iam::123456789012:role/service-role/QuickSightAction
arn:aws:iam::123456789012:policy/UsersManageOwnCredentials
arn:aws:iam::123456789012:policy/division_abc/subdivision_xyz/UsersManageOwnCredentials
arn:aws:iam::123456789012:instance-profile/Webserver
arn:aws:sts::123456789012:federated-user/John
```

```
arn:aws:sts::123456789012:assumed-role/Accounting-Role/Jane
arn:aws:sts::123456789012:self
arn:aws:iam::123456789012:mfa/JaneMFA
arn:aws:iam::123456789012:u2f/user/John/default (U2F security key)
arn:aws:iam::123456789012:server-certificate/ProdServerCert
arn:aws:iam::123456789012:server-certificate/division_abc/subdivision_xyz/
ProdServerCert
arn:aws:iam::123456789012:saml-provider/ADFSProvider
arn:aws:iam::123456789012:oidc-provider/GoogleProvider
arn:aws:iam::123456789012:oidc-provider/oidc.eks.us-west-2.amazonaws.com/id/
a1b2c3d4567890abcdefEXAMPLE11111
arn:aws:iam::123456789012:oidc-provider/server.example.org
arn:aws:iam::aws:contextProvider/IdentityCenter
```

Gli esempi seguenti forniscono maggiori dettagli per aiutarti a comprendere il formato ARN per diversi tipi di IAM e AWS STS risorse.

- Un utente IAM nell'account:

> **Note**
>
> Ogni nome utente IAM è univoco. Il nome utente non fa distinzione tra maiuscole e minuscole per l'utente, ad esempio durante il processo di accesso, ma la fa quando lo si utilizza in una politica o come parte di un ARN.

```
arn:aws:iam::123456789012:user/John
```

- Altro utente con un percorso che riflette un organigramma:

```
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/Jane
```

- Un gruppo di utenti IAM:

```
arn:aws:iam::123456789012:group/Developers
```

- Un gruppo di utenti IAM con un percorso:

```
arn:aws:iam::123456789012:group/division_abc/subdivision_xyz/product_A/Developers
```

- Ruolo IAM:

```
arn:aws:iam::123456789012:role/S3Access
```

- Un ruolo collegato al servizio:

```
arn:aws:iam::123456789012:role/aws-service-role/access-analyzer.amazonaws.com/
AWSServiceRoleForAccessAnalyzer
```

- Un ruolo di servizio:

```
arn:aws:iam::123456789012:role/service-role/QuickSightAction
```

- Policy gestita:

```
arn:aws:iam::123456789012:policy/ManageCredentialsPermissions
```

- Profilo dell'istanza che può essere associato a un'istanza Amazon EC2:

```
arn:aws:iam::123456789012:instance-profile/Webserver
```

- Un utente principale AWS STS federato identificato in IAM come «Paulo»:

```
arn:aws:sts::123456789012:federated-user/Paulo
```

- La sessione attiva di qualcuno che assume il ruolo di "Accounting-Role«, con il nome di sessione di ruolo «Mary»:

```
arn:aws:sts::123456789012:assumed-role/Accounting-Role/Mary
```

- Rappresenta la sessione del chiamante quando viene utilizzata come risorsa in una chiamata API, come l' AWS STS SetContextAPI, che opera nella sessione chiamante:

```
arn:aws:sts::123456789012:self
```

- Dispositivo di autenticazione a più fattori assegnato all'utente denominato Jorge:

```
arn:aws:iam::123456789012:mfa/Jorge
```

- Certificato del server:

```
arn:aws:iam::123456789012:server-certificate/ProdServerCert
```

- Certificato server con un percorso che riflette un organigramma:

```
arn:aws:iam::123456789012:server-certificate/division_abc/subdivision_xyz/
ProdServerCert
```

- Provider di identità (SAML e OIDC):

```
arn:aws:iam::123456789012:saml-provider/ADFSProvider
arn:aws:iam::123456789012:oidc-provider/GoogleProvider
arn:aws:iam::123456789012:oidc-provider/server.example.org
```

- Provider di identità OIDC con un percorso che riflette l'URL di un provider di identità OIDC di Amazon EKS:

```
arn:aws:iam::123456789012:oidc-provider/oidc.eks.us-west-2.amazonaws.com/id/
a1b2c3d4567890abcdefEXAMPLE11111
```

- Il provider di contesto ARN utilizzato per la propagazione dell'identità Centro identità AWS IAM affidabile, da cui è stata generata l'asserzione di contesto attendibile:

```
arn:aws:iam::aws:contextProvider/IdentityCenter
```

Un altro importante ARN è l'ARN dell'utente root. Sebbene questa non sia una risorsa IAM, bisognerebbe essere a conoscenza del formato di questo ARN. Viene spesso utilizzato nell'elemento `Principal` di una policy basata su risorse.

- Account AWS Visualizza quanto segue:

```
arn:aws:iam::123456789012:root
```

L'esempio seguente mostra una policy che può essere assegnata a Richard per la gestione autonoma delle sue chiavi di accesso. La risorsa è l'utente IAM Richard.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "ManageRichardAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:*AccessKey*",
                "iam:GetUser"
            ],
            "Resource": "arn:aws:iam::*:user/division_abc/subdivision_xyz/
Richard"
        },
        {
            "Sid": "ListForConsole",
            "Effect": "Allow",
            "Action": "iam:ListUsers",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Quando utilizzi gli ARN per identificare le risorse in una policy IAM, puoi includere le variabili delle policy. Le variabili dei criteri possono includere segnaposti per le informazioni di runtime (ad esempio il nome dell'utente) come parte dell'ARN. Per ulteriori informazioni, consulta la sezione Elementi delle policy IAM: variabili e tag

Utilizzo di caratteri jolly e percorsi negli ARN

Puoi utilizzare i caratteri jolly nella *resource* parte dell'ARN per specificare più utenti o gruppi o policy IAM. Ad esempio, per specificare tutti gli utenti che lavorano su un prodotto denominato product_1234, puoi utilizzare:

```
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/product_1234/*
```

Se hai utenti i cui nomi iniziano con la stringa app_, puoi fare riferimento a tutti quelli con il seguente ARN.

```
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/product_1234/app_*
```

Per specificare tutti gli utenti, i gruppi IAM o le policy presenti nel tuo Account AWS, utilizza una jolly rispettivamente dopo l'`user/`ARN o `policy/` parte di esso. `group/`

```
arn:aws:iam::123456789012:user/*
arn:aws:iam::123456789012:group/*
arn:aws:iam::123456789012:policy/*
```

Se si specifica il seguente ARN per un utente `arn:aws:iam::111122223333:user/*`, questo corrisponderà a entrambi gli esempi seguenti.

```
arn:aws:iam::111122223333:user/John
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/JaneDoe
```

Ma, se specifichi il seguente ARN per un utente `arn:aws:iam::111122223333:user/division_abc*`, corrisponderà al secondo esempio, ma non al primo.

```
arn:aws:iam::111122223333:user/John
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/Jane
```

Non utilizzare caratteri jolly nella parte `user/`, `group/`o `policy/` dell'ARN. Ad esempio, IAM non consente quanto segue:

```
arn:aws:iam::123456789012:u*
```

> **Note**
>
> Quando specifichi un ARN incompleto (uno con meno dei sei campi standard) in una policy basata sull'identità, completa AWS automaticamente l'ARN aggiungendo caratteri jolly (*) a tutti i campi mancanti. Ad esempio, specificare equivale a`arn:aws:sqs:*:*:*`, che consente `arn:aws:sqs` l'accesso a tutte le risorse Amazon Amazon SQS in tutte le regioni e gli account. Tuttavia, le policy di sessione trasmesse e AWS STS AssumeRole AssumeRoleWithWebIdentity le AssumeRoleWithSAML richieste non supportano gli ARN incompleti. L'utilizzo di un ARN incompleto in una policy di sessione genererà un `MalformedPolicyDocumentException` errore.

Example Esempio di utilizzo di percorsi e ARN per un gruppo di utenti basato su progetto

I percorsi non possono essere creati o modificati nella Console di gestione AWS. Per utilizzare i percorsi è necessario utilizzare la risorsa utilizzando l' AWS API AWS CLI, o gli Strumenti per Windows PowerShell.

In questo esempio, Jules nel gruppo di utenti crea un gruppo di Marketing_Admin utenti basato sul progetto all'interno del percorso /marketing/. Jules assegna gli utenti provenienti da diverse parti dell'azienda al gruppo di utenti. Questo esempio illustra come il percorso di un utente non sia correlato ai gruppi di utenti in cui si trova l'utente.

Il gruppo di marketing ha un nuovo prodotto da lanciare, quindi Jules crea un nuovo gruppo di utenti nel percorso /marketing/ chiamato. Widget_Launch Jules assegna al gruppo la seguente policy, che fornisce l'accesso al gruppo di utenti a oggetti nella parte del `example_bucket` progettato appositamente per il lancio del prodotto.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::example_bucket/marketing/newproductlaunch/widget/
*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket*",
      "Resource": "arn:aws:s3:::example_bucket",
      "Condition": {"StringLike": {"s3:prefix": "marketing/newproductlaunch/
widget/*"}}
    }
  ]
}
```

Jules assegna quindi al gruppo di utenti gli utenti coinvolti nel lancio del prodotto. Fra questi ci sono Patricia ed Eli dal percorso /marketing/. Sono inclusi anche Chris e Chloe dal percorso /vendite/ e Aline e Jim dal percorso /legale/.

Identificatori univoci

Quando IAM crea un utente, un gruppo di utenti, un ruolo, una policy, un profilo dell'istanza o un certificato server, assegna a ciascuna risorsa un ID univoco. L'ID univoco ha il seguente aspetto:

```
AIDAJQABLZS4A3QDU576Q
```

In linea di massima, vengono utilizzati nomi descrittivi e ARN quando si lavora con le risorse IAM. In questo modo non è necessario conoscere l'ID univoco per una risorsa specifica. Tuttavia, a volte l'ID univoco può risultare utile, se l'utilizzo di nomi descrittivi non risulta pratico.

Un esempio riutilizza i nomi descrittivi nel tuo. Account AWS All'interno dell'account, il nome descrittivo di un utente, di un gruppo di utenti, di un ruolo o di una policy deve essere univoco. Ad esempio, potresti creare un utente IAM denominato `John`. La tua azienda utilizza Amazon S3 e dispone di un bucket con cartelle per ogni dipendente. L'utente IAM `John` è un membro di un gruppo di utenti IAM denominato `User-S3-Access` con autorizzazioni che consentono agli utenti di accedere solo alle loro cartelle nel bucket. Per un esempio di come creare una policy basata sull'identità che consenta agli utenti IAM di accedere al loro oggetto del bucket in S3 utilizzando il nome descrittivo degli utenti, consulta Amazon S3: consente agli utenti IAM di accedere alla propria directory home S3, in modo programmatico e nella console.

Supponiamo che il dipendente denominato David lasci l'azienda e che il suo utente IAM denominato `John` venga eliminato. Successivamente, viene assunto un altro dipendente con lo stesso nome e viene creato un nuovo utente IAM, anch'esso denominato `John`. Aggiungi il nuovo utente IAM denominato `John` al gruppo di utenti IAM esistente `User-S3-Access`. Se la policy associata al gruppo di utente specifica il nome descrittivo dell'utente IAM `John`, la policy consente al nuovo John di accedere alle informazioni lasciate dal John precedente.

Come regola generale, ti consigliamo di specificare la ARN per la risorsa nelle policy invece del suo ID univoco. Tuttavia, ogni utente IAM dispone di un ID univoco, anche se crei un nuovo utente IAM che riutilizza un nome descrittivo che avevi eliminato in precedenza. In questo esempio, il vecchio utente IAM `John` e il nuovo utente IAM `John` hanno ID univoci differenti. È possibile creare policy basate sulle risorse che concedono l'accesso in base all'ID univoco e non solo per nome utente. In questo modo si riduce la possibilità che si possa inavvertitamente concedere l'accesso alle informazioni che un dipendente non dovrebbe avere.

L'esempio seguente mostra come specificare ID univoci nell'elemento `Principal` di una policy basata su risorse.

```
"Principal": {
  "AWS": [
    "arn:aws:iam::111122223333:role/role-name",
    "AIDACKCEVSQ6C2EXAMPLE",
    "AROADBQP57FF2AEXAMPLE"
  }
```

L'esempio seguente mostra come specificare ID univoci nell'elemento `Condition` di una policy che utilizza la chiave di condizione globale `aws:userid`.

```
"Condition": {
    "StringLike": {
      "aws:userId": [
        "AIDACKCEVSQ6C2EXAMPLE",
        "AROADBQP57FF2AEXAMPLE:role-session-name",
        "AROA1234567890EXAMPLE:*",
        "111122223333"
      ]
    }
  }
```

Un altro esempio è dato dagli ID utente che possono tornare utili quando ti trovi a gestire un database personale (o un altro archivio) con informazioni sul ruolo o sugli utenti IAM. L'ID univoco può fornire un identificatore univoco per ogni ruolo o utente IAM che viene creato. Questo è il caso di ruoli o utenti IAM che riutilizzano un nome, come nell'esempio precedente.

Approfondimento dei prefissi ID univoci

IAM usa i seguenti prefissi per indicare il tipo di risorsa a cui si applica ciascun ID univoco. I prefissi possono variare in base a quando sono stati creati.

Prefix	Tipo di risorsa
ABIA	AWS STS token service bearer
ACCA	Context-specific credenziale
AGPA	Gruppo di utenti
AIDA	Utente IAM

Prefix	Tipo di risorsa
AIPA	Profilo dell'istanza Amazon EC2
AKIA	Chiave di accesso
ANPA	Policy gestita
ANVA	Versione in una policy gestita
APKA	Chiavi pubbliche
AROA	Ruolo
ASCA	Certificato
ASIA	Gli ID chiave di accesso temporanea (AWS STS) utilizzano questo prefisso, ma sono univoci solo in combinazione con la chiave di accesso segreta e il token di sessione.

Ottenere l'identificatore univoco

L'ID univoco per una risorsa IAM non è disponibile nella console IAM. Per ottenere l'ID univoco, puoi utilizzare i seguenti AWS CLI comandi o chiamate API IAM.

AWS CLI:

- get-caller-identity
- get-group
- get-role
- get-user
- get-policy
- get-instance-profile
- get-server-certificate

API IAM:

- GetCallerIdentity
- GetGroup
- GetRole
- GetUser
- GetPolicy
- GetInstanceProfile
- GetServerCertificate

IAM e AWS STS quote

AWS Identity and Access Management (IAM) e Servizio di token di sicurezza AWS (STS) hanno quote che limitano le dimensioni degli oggetti. Questi servizi limitano anche la modalità di denominazione di un oggetto, il numero di oggetti che è possibile creare e il numero di caratteri che è possibile utilizzare quando si passa un oggetto.

> **Note**
>
> Per ottenere informazioni a livello di account sull'utilizzo e sulle quote IAM, utilizza l'operazione GetAccountSummary API o il comando get-account-summary. https://docs.aws.amazon.com/cli/latest/reference/iam/get-account-summary.html AWS CLI

Requisiti del nome IAM

I nomi IAM sono caratterizzati dalle limitazioni e dai requisiti seguenti:

- I documenti delle policy possono contenere solo i seguenti caratteri Unicode: tabulatore orizzontale (U+0009), avanzamento riga (U+000A), ritorno a capo (U+000D) e i caratteri compresi fra U+0020 a U+00FF.
- I nomi di utenti, gruppi, ruoli, policy, profili dell'istanza, certificati server e percorsi devono essere alfanumerici, inclusi i seguenti caratteri comuni: più (+), uguale (=), virgola (,), punto (.), a (@), trattino basso (_) e trattino (-). I nomi dei percorsi devono iniziare e terminare con una barra (/).
- I nomi di utenti, gruppi, ruoli e profili di istanze devono essere univoci all'interno dell'account. Non viene applicata la distinzione fra maiuscole e minuscole. Ad esempio, non è possibile creare un gruppo **`ADMINS`** e un altro **`admins`**.

- Il valore dell'ID esterno che una terza parte utilizza per assumere un ruolo deve avere un minimo di 2 caratteri e un massimo di 1.224 caratteri. Il valore deve essere alfanumerico senza spazi. Può anche includere i seguenti simboli: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), due punti (:), barra (/) e trattino (-). Per ulteriori informazioni sull'ID esterno, consulta Accesso a Account AWS di proprietà di terzi.
- I nomi delle policy in linea devono essere univoci per l'utente, gruppo o ruolo in cui sono incorporati. I nomi possono contenere qualsiasi carattere latino di base (ASCII), ad eccezione di alcuni caratteri riservati: barra rovesciata (\), barra (/), asterisco (*), punto interrogativo (?) e spazio. Questi caratteri sono riservati in base a RFC 3986, sezione 2.2.
- Le password utente (profili di accesso) possono contenere tutti i caratteri latini di base (ASCII).
- Account AWS Gli alias ID devono essere univoci per tutti i AWS prodotti e devono essere alfanumerici secondo le convenzioni di denominazione DNS. Un alias deve essere in lettere minuscole, non può iniziare o terminare con un trattino, non può contenere due trattini consecutivi e non può essere un numero di 12 cifre.

Per un elenco dei caratteri latini di base (ASCII), consulta la Tabella dei codici latini di base (ASCII) della Library of Congress.

IAM Quote oggetto

Le quote, note anche come limiti in AWS, sono i valori massimi per le risorse, le azioni e gli elementi presenti nel tuo. Account AWSÈ possibile utilizzare Service Quotas per gestire le quote IAM.

Per l'elenco degli endpoint e delle Service Quotas di IAM, consulta Endpoint e quote di AWS Identity and Access Management nella Riferimenti generali di AWS

Richiesta di un aumento delle quote

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere a AWS nella Guida per l'AWS Sign-In utente per accedere al. Console di gestione AWS
2. Apri la console Service Quotas.
3. Nel pannello di navigazione, scegli Servizi AWS (servizi AWS).
4. Sulla barra di navigazione, selezionare la regione US East (N. Virginia). Quindi cercare **`IAM`**.
5. Scegli AWS Identity and Access Management (IAM), seleziona una quota e segui le istruzioni per richiedere un aumento di quota.

Per ulteriori informazioni, consulta Richiesta di un aumento di quota nel Guida per l'utente di Service Quotas.

Per un esempio di come richiedere un aumento della quota IAM utilizzando la console Service Quotas, guarda il video seguente.

Richiedi un aumento della quota IAM utilizzando la console Service Quotas.

Puoi richiedere un aumento delle quote predefinite per le quote IAM regolabili. Le richieste fino a maximum quota vengono automaticamente approvate e completate in pochi minuti.

La tabella seguente elenca le risorse per le quali gli aumenti delle quote possono essere approvati automaticamente.

Risorsa	Quota predefinita	Quota massima
Policy gestite dal cliente per account	1500	10000
Gruppi per account	300	500
Profili di istanza per account	1000	10000
Policy gestite per ruolo	20	25
Policy gestite per utente	10	20
Policy gestite per gruppo	10	10
Lunghezza della policy di attendibilità del ruolo	2048 caratteri	8192 caratteri
Ruoli per account	1000	10000
Certificati server per account	20	20
OpenId connetti i provider per account	100	700

Quote Sistema di analisi degli accessi IAM

Per l'elenco degli endpoint e delle Service Quotas di Sistema di analisi degli accessi IAM, consulta Endpoint e quote di Sistema di analisi degli accessi IAM nella Riferimenti generali di AWS.

Quote di IAM Roles Anywhere

Per l'elenco degli endpoint e delle Service Quotas di IAM Roles Anywhere, consulta Endpoint e quote di AWS Identity and Access Management Roles Anywhere nella Riferimenti generali di AWS.

Quote di richiesta STS

La Servizio di token di sicurezza AWS (AWS STS) applica le seguenti quote di richiesta.

Per AWS STS le richieste effettuate utilizzando AWS le credenziali, la quota di richieste predefinita è di 600 richieste al secondo, per account, per regione. Le seguenti AWS STS operazioni condividono questa quota:

- AssumeRole
- DecodeAuthorizationMessage
- GetAccessKeyInfo
- GetCallerIdentity
- GetFederationToken
- GetSessionToken

> **Note**
>
> Le richieste AWS STS inviate dai responsabili del AWS servizio, ad esempio quelle utilizzate per assumere ruoli da utilizzare con un AWS servizio, non consumano la quota di richieste STS al secondo nei tuoi account.

Ad esempio, se an Account AWS effettua 100 GetCallerIdentity richieste al secondo e 100 AssumeRole chiamate al secondo nella stessa regione, quell'account sta consumando 200 delle 600 richieste STS disponibili al secondo per quella regione.

Per AssumeRole le richieste tra account, solo l'account che effettua la AssumeRole richiesta influisce sulla quota STS. L'account di destinazione non ha alcuna quota consumata.

Per richiedere un aumento delle quote di richiesta STS, apri un ticket con l'assistenza AWS .

> **Note**
>
> Con le imminenti modifiche all'endpoint AWS STS globale (`https://sts.amazonaws.com`), le richieste all'endpoint globale non condivideranno una quota di richieste al secondo (RPS) con gli endpoint AWS STS regionali nelle regioni abilitate per impostazione predefinita. https://docs.aws.amazon.com/accounts/latest/reference/manage-acct-regions.html Quando una richiesta all'endpoint AWS STS globale proviene da una singola regione, verrà conteggiata nella quota RPS dell'endpoint globale. Tuttavia, quando le richieste provengono da più Regioni, ogni Regione aggiuntiva riceverà la propria quota RPS indipendente. Per ulteriori informazioni sulle modifiche AWS STS globali agli endpoint, vedere. AWS STS cambiamenti globali degli endpoint

Limiti di caratteri di IAM e STS

Di seguito sono riportati i numeri massimi di caratteri e i limiti di dimensione per IAM e AWS STS. Non puoi richiedere un aumento per i seguenti limiti.

Description	Limite
Alias per un ID Account AWS	3-63 caratteri
Per policy inline	Non esiste un limite per le policy inline che puoi aggiungere a un utente, a un gruppo o a un ruolo IAM. Tuttavia, la dimensione cumulativa della policy (ovvero, la somma delle dimension i di tutte le policy in linea) per ciascuna entità non può superare i seguenti limiti: • La dimensione della policy dell'utente non può superare i 2.048 caratteri. • La dimensione della policy del ruolo non può superare i 10.240 caratteri. • La dimensione della policy del gruppo non può superare i 5.120 caratteri.

Description	Limite
	Note IAM non conta gli spazi nel calcolo per determinare le dimensioni di una policy rispetto a tali limiti.
Per le politiche gestite dal cliente	• La dimensione di ciascuna policy gestita non può superare i 6.144 caratteri. **Note** IAM non conta gli spazi nel calcolo per determinare le dimensioni di una policy rispetto a tale limite.
Group name (Nome gruppo)	128 caratteri
Nome del profilo dell'istanza	128 caratteri
Password per un profilo di accesso	1-128 caratteri
Path	512 caratteri
Nome policy	128 caratteri
Role Name (Nome ruolo)	64 caratteri **Important** Se intendi utilizzare un ruolo con la funzione Cambia ruolo in Console di gestione AWS, la somma `Path` dei ruoli non `RoleName` può superare i 64 caratteri.

Description	Limite
Durata della sessione del ruolo	12 ore Quando assumi un ruolo dall'API AWS CLI or, puoi utilizzare il parametro `duration-seconds` CLI o il parametro `DurationSeconds` API per richiedere una sessione di ruolo più lunga. È possibile specificare un valore compreso fra 900 secondi (15 minuti) fino all'impostazione massima della durata consentita per il ruolo, che può variare da 1 a 12 ore. Se non specifichi un valore per il parametro `DurationSeconds` le tue credenziali di sicurezza rimarranno valide per un'ora. Agli utenti IAM che cambiano ruoli nella console viene concessa la durata massima della sessione o il tempo rimanente nella sessione dell'utente, a seconda di quale sia minore. L'impostazione di durata massima delle sessioni non limita le sessioni assunte dai servizi AWS . Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Aggiornamento della durata massima della sessione per un ruolo.
Nome della sessione del ruolo	64 caratteri

Description	Limite
Policy di sessione del ruolo	• La dimensione del documento di policy JSON inviato e tutti i caratteri ARN delle policy gestite inviate non possono superare i 2.048 caratteri. • È possibile passare un massimo di 10 ARN delle policy gestite quando si crea una sessione. • Puoi passare un solo documento di policy JSON quando crei a livello di codice una sessione temporanea per un ruolo o AWS STS un'entità utente federata. • Inoltre, i criteri di sessione e i tag di sessione aumentano la dimensione del token di sessione, che ha un limite separato. L'elemento di `SessionTokenUtilization` risposta mostra la percentuale del limite di dimensione utilizzata dal token. L'elemento di `PackedPolicySize` risposta è obsoleto. Usare invece `SessionTokenUtilization` . • Ti consigliamo di passare le policy di sessione utilizzando l'API AWS CLI or AWS . Console di gestione AWS Potrebbe aumentare la dimensione del token di sessione aggiungendo informazioni sulla sessione.

Description	Limite
Tag di sessione per il ruolo	• I tag di sessione devono soddisfare il limite della chiave del tag di 128 caratteri e il limite del valore del tag di 256 caratteri. • È possibile passare fino a 50 tag di sessione. • I criteri di sessione e i tag di sessione si aggiungono alla dimensione del token di sessione, che ha un limite separato. Puoi passare i tag di sessione utilizzando l' AWS API AWS CLI or. L'elemento di `SessionTokenUtilization` risposta mostra la percentuale del limite di dimensione utilizzata dal token. L'elemento di `PackedPolicySize` risposta è obsoleto. Usare invece `SessionTokenUtilization` .
Risposta di autenticazione SAML codificata con base64	100.000 caratteri Questo limite di caratteri si applica all'operazione della CLI `assume-role-with-saml` o dell'API `AssumeRoleWithSAML` .
Chiave tag	128 caratteri Questo limite di caratteri si applica ai tag sulle risorse IAM e ai tag di sessione.
Valore tag	256 caratteri Questo limite di caratteri si applica ai tag sulle risorse IAM e ai tag di sessione. I valori dei tag possono essere vuoti, ciò significa che possono avere una lunghezza di 0 caratteri.

Description	Limite
ID univoci creati da IAM	128 caratteri Ad esempio: • ID utente che iniziano con `AIDA` • ID di gruppi che iniziano con `AGPA` • ID di ruoli che iniziano con `AROA` • ID di policy gestite che iniziano con `ANPA` • ID di certificati server che iniziano con `ASCA` **Note** Questo elenco non è completo e non fornisce alcuna garanzia che gli ID di un determinato tipo inizino sempre con la combinazione di lettere specificata.
Nome utente	64 caratteri

Supporto per endpoint dual-stack

Gli endpoint dual-stack consentono ai client di comunicare con AWS i servizi utilizzando uno o più indirizzi. IPv4 IPv6 Sia IAM che AWS STS forniscono supporto dual-stack per endpoint. Per ulteriori informazioni su come configurare il tuo VPC per la modalità dual-stack, consulta il supporto IPv6 per il tuo VPC nella Amazon VPC User Guide.

Se utilizzi policy IAM che includono le chiavi di `aws:vpcSourceIp` condizione `aws:sourceIp` or, devi aggiornare queste policy per supportare i clienti. IPv6 Per ulteriori informazioni sull' IPv6 indirizzamento per le tue sottoreti VPCs, consulta Indirizzamento IP per VPCs le tue sottoreti nella Amazon VPC User Guide.

Supporto per endpoint dual-stack con IAM

IAM fornisce un endpoint pubblico dual-stack che supporta sia i client che i client. IPv4 IPv6 È possibile accedere all'endpoint pubblico IAM dual-stack anche privatamente dal cloud privato virtuale

(VPC) utilizzando. AWS PrivateLink Per ulteriori informazioni sulla creazione di endpoint VPC di interfaccia privati per IAM, consulta Creare un endpoint VPC per IAM.

L'endpoint pubblico IAM dual-stack è. `https://iam.global.api.aws`

L'endpoint pubblico IAM at `https://iam.amazonaws.com`, a differenza dell'endpoint pubblico dual-stack, supporta solo client. IPv4 Se vi si accede privatamente dal tuo VPC AWS PrivateLink utilizzando, l'endpoint pubblico IAM può supportare IPv4 sia i client che quelli. IPv6

AWS STS supporto per endpoint dual-stack

AWS STS fornisce endpoint regionali dual-stack che supportano sia i client che quelli. IPv4 IPv6 È possibile accedere agli endpoint regionali AWS STS dual-stack anche privatamente dal cloud privato virtuale (VPC) utilizzando. AWS PrivateLink Per ulteriori informazioni sulla creazione di endpoint VPC con interfaccia privata, vedere. AWS STSCrea un endpoint VPC per AWS STS

La tabella seguente mostra gli endpoint regionali AWS STS dual-stack per partizione:

Partizione	URL dell'endpoint dual-stack
Regioni AWS commerciali	Endpoint regolari: `https://sts.` *`<region>`*`.api.aws` Endpoint FIPS: `https://sts-fips.` *`<region>`*`.api.aws`
AWS GovCloud (US) Regioni	Endpoint regolari: `https://sts.` *`<region>`*`.api.aws`
Regioni della Cina	Endpoint regolari: `https://sts.` *`<region>`*`.api.amazonwebservices.com.cn`

> **Note**
>
> Gli endpoint FIPS si applicano solo alle regioni degli Stati Uniti e del Canada.

Limitazioni

Le seguenti limitazioni si applicano al supporto AWS STS dual-stack per endpoint:

- IPv6 i client non sono supportati sull'endpoint globale. `https://sts.amazonaws.com` È necessario utilizzare endpoint regionali dual-stack per l'assistenza clienti. IPv6

- IPv6-non sono AWS STS supportati solo gli endpoint VPC. Gli endpoint VPC possono essere configurati per IPv4 la connettività dual-stack.

Endpoint VPC di interfaccia

Se utilizzi Amazon Virtual Private Cloud (Amazon VPC) per ospitare AWS le tue risorse, puoi stabilire una connessione privata tra il tuo VPC e AWS Identity and Access Management (IAM) o Servizio di token di sicurezza AWS (AWS STS). Puoi utilizzare questa connessione per abilitare IAM o AWS STS per comunicare con le tue risorse nel tuo VPC senza accedere alla rete Internet pubblica.

Amazon VPC è un AWS servizio che puoi utilizzare per avviare AWS risorse in una rete virtuale da te definita. Con un VPC, detieni il controllo delle impostazioni della rete, come l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete. Per connettere il tuo VPC a IAM or AWS STS, definisci un endpoint VPC di interfaccia per ogni servizio. L'endpoint fornisce una connettività affidabile e scalabile a IAM AWS STS senza richiedere un gateway Internet, un'istanza NAT (Network Address Translation) o una connessione VPN. Per ulteriori informazioni, consulta Che cos'è Amazon VPC? nella Guida per l'utente di Amazon VPC

Gli endpoint Interface VPC sono alimentati da AWS PrivateLink una AWS tecnologia che consente la comunicazione privata tra AWS servizi utilizzando un'interfaccia di rete elastica con indirizzi IP privati. Per ulteriori informazioni, consulta la sezione Servizi AWS PrivateLink . AWS

Le informazioni seguenti sono per gli utenti di Amazon VPC. Per ulteriori informazioni, consulta Nozioni di base su Amazon VPC nella Guida per l'utente di Amazon VPC.

Argomenti

- Disponibilità dell'endpoint VPC
- Creare un endpoint VPC per IAM
- Crea un endpoint VPC per AWS STS
- Crea un endpoint VPC per AWS STS Discovery OIDC
- Controlla l'accesso a AWS STS con politiche per gli endpoint VPC
- Usa le chiavi di condizione VPC per controllare l'accesso federato

Disponibilità dell'endpoint VPC

> **Important**
>
> Gli endpoint VPC dell'interfaccia per IAM possono essere creati solo nella regione in cui si trova il piano di controllo IAM. Se il tuo VPC si trova in una regione diversa dalla regione del piano di controllo IAM, devi AWS Transit Gateway utilizzarlo per consentire l'accesso all'endpoint VPC dell'interfaccia IAM da un'altra regione. Per ulteriori informazioni, consulta Creare un endpoint VPC per IAM.

IAM al momento supporta endpoint VPC nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Cina (Pechino)
- AWS GovCloud (US-West)

AWS STS attualmente supporta gli endpoint VPC nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacifico (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Melbourne)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Osaka)
- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)

- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Canada occidentale (Calgary)
- Cina (Pechino)
- Cina (Ningxia)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Spagna)
- Europa (Stoccolma)
- Europa (Zurigo)
- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Creare un endpoint VPC per IAM

Per iniziare a usare IAM con il VPC, crea un endpoint VPC dell'interfaccia per IAM. Per ulteriori informazioni, consulta Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia nella Amazon VPC User Guide.

Gli endpoint VPC dell'interfaccia per IAM possono essere creati solo nella regione in cui si trova il piano di controllo IAM. In ambito commerciale Regioni AWS, l'aereo di controllo IAM si trova nella regione degli Stati Uniti orientali (Virginia settentrionale) (us-east-1). Il nome del servizio endpoint VPC di AWS PrivateLink interfaccia per IAM è. `com.amazonaws.iam` Per un elenco degli endpoint VPC Regioni AWS che supportano IAM, consulta. Disponibilità dell'endpoint VPC

Se il tuo VPC si trova in una regione diversa dalla regione del piano di controllo IAM, devi utilizzare AWS Transit Gateway per consentire l'accesso all'endpoint VPC dell'interfaccia IAM da un'altra regione.

Per accedere a un endpoint VPC con interfaccia IAM da un VPC in una regione diversa utilizzando AWS Transit Gateway

1. Crea un gateway di transito o utilizza un gateway di transito esistente per interconnettere i tuoi cloud privati virtuali (). VPCs Un gateway di transito è necessario per ogni regione. Per ulteriori informazioni, consulta Creazione di un gateway di transito nella Guida per AWS Transit Gateway .
2. Crea allegati VPC del gateway di transito per collegare ogni VPC al gateway di transito. Per ulteriori informazioni, consulta Creazione di un collegamento del gateway di transito a un VPC nella Guida per AWS Transit Gateway .
3. Crea un allegato di peering VPC del gateway di transito per instradare il traffico tra i peer. VPCs Per ulteriori informazioni, consulta Creare un allegato di peering nella Guida per AWS Transit Gateway .

> **Note**
>
> Le connessioni peering VPC possono anche instradare il traffico tra connessioni peer VPCs, ma questo metodo non è scalabile bene con un numero elevato di. VPCs Invece del peering VPC, consigliamo di utilizzare gli allegati di peering AWS Transit Gateway che migliorano la gestione del VPC e della rete on-premises attraverso un hub centrale scalabile. Per ulteriori informazioni sulle connessioni peering VPC, consulta Utilizzo di connessioni peering VPC nella Guida al peering di Amazon VPC.

Crea un endpoint VPC per AWS STS

Per iniziare a utilizzarlo AWS STS con il tuo VPC, crea un'interfaccia VPC endpoint per. AWS STS Per ulteriori informazioni, consulta Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia nella Amazon VPC User Guide.

Dopo aver creato l'endpoint VPC, devi utilizzare l'endpoint regionale corrispondente per inviare le tue richieste. AWS STS AWS STS consiglia di utilizzare entrambi i `setEndpoint` metodi `setRegion` and per effettuare chiamate a un endpoint regionale. Puoi utilizzare il metodo `setRegion` da solo per

regioni abilitate manualmente, ad esempio Asia Pacifico (Hong Kong). In questo caso, le chiamate sono indirizzate all'endpoint regionale STS. Per ulteriori informazioni su come abilitare manualmente una regione, consulta Gestione delle regioni AWS in Riferimenti generali di AWS. Se utilizzi il metodo `setRegion` da solo per regioni abilitate per default, le chiamate vengono indirizzate all'endpoint globale di `https://sts.amazonaws.com` .

Quando utilizzi endpoint regionali, AWS STS chiama altri AWS servizi utilizzando endpoint pubblici o endpoint VPC con interfaccia privata, a seconda di quali siano in uso. Ad esempio, supponiamo di aver creato un endpoint VPC di interfaccia AWS STS e di aver già richiesto credenziali temporanee alle risorse che si trovano nel tuo VPC. AWS STS In tal caso, queste credenziali iniziano a fluire attraverso l'endpoint VPC dell'interfaccia per impostazione predefinita. Per ulteriori informazioni su come effettuare richieste regionali utilizzando, consulta. AWS STSManage (Gestione) AWS STS in un Regione AWS

Crea un endpoint VPC per AWS STS Discovery OIDC

Servizio di token di sicurezza AWS (AWS STS) Il rilevamento OIDC è costituito da una coppia di endpoint pubblici che forniscono i metadati e le chiavi di verifica necessarie alle parti interessate per convalidare i JSON Web Token (JWT) emessi dalla federazione delle identità in uscita. ???

Quando abiliti la federazione delle identità in uscita, pubblica un URL dell'emittente specifico dell'account. AWS STS Questo URL ospita due endpoint di rilevamento OpenID Connect (OIDC). I relying party recuperano questi endpoint per ottenere i metadati e le chiavi pubbliche di cui hanno bisogno per verificare i JWT che emettono: GetWebIdentityToken

- `{issuer_url}/.well-known/openid-configuration`— Il documento di scoperta dell'OIDC
- `{issuer_url}/.well-known/jwks.json`— Il JSON Web Key Set (JWKS) che contiene le chiavi di verifica della firma del token

Per impostazione predefinita, per raggiungere questi endpoint dal tuo VPC sono necessari un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. Per mantenere invece questo traffico sulla AWS rete, create un endpoint VPC di interfaccia per l'endpoint di rilevamento STS OIDC.

> **Note**
>
> L'endpoint di rilevamento STS OIDC serve solo i due endpoint sopra elencati. Non serve ad alcuna operazione API. AWS STS Per chiamare `GetWebIdentityToken`

o eseguire qualsiasi altra AWS STS operazione in privato, utilizza il servizio `com.amazonaws.region.sts` endpoint. `AssumeRole` Consulta Crea un endpoint VPC per AWS STS.

Scelta del servizio endpoint giusto

La federazione delle identità in uscita utilizza due percorsi di rete distinti. La maggior parte dei carichi di lavoro che emettono e verificano token richiedono entrambi i servizi endpoint.

Operazione	Servizio endpoint	Autenticazione
Crea un JWT () `sts:GetWebIdentityToken`	`com.amazonaws.region.sts`	AWS SIG v4
Leggi `/.well-known/openid-configuration`	`com.amazonaws.region.sts-oidc`	Nessuno (non autenticato)
Leggi `/.well-known/jwks.json`	`com.amazonaws.region.sts-oidc`	Nessuno (non autenticato)

Non è necessario l'endpoint di rilevamento STS OIDC se l'unica parte che verifica i token è un servizio esterno. AWS Tale servizio recupera l'URL dell'emittente sulla rete Internet pubblica e il relativo traffico non entra nel tuo VPC.

Considerazioni

Considerate quanto segue prima di creare un endpoint VPC di interfaccia per il rilevamento STS OIDC.

- Le policy degli endpoint VPC non sono supportate. Entrambe le API di rilevamento servono dati disponibili al pubblico e l'endpoint di rilevamento STS OIDC non applica le policy degli endpoint VPC. Utilizza i gruppi di sicurezza e il posizionamento delle sottoreti per controllare quali risorse possono raggiungere l'endpoint.
- Sono serviti solo due percorsi. L'endpoint accetta `GET` richieste HTTPS per `/.well-known/openid-configuration` e `/.well-known/jwks.json` solo. Tutti gli altri percorsi e metodi restituiscono un errore.

Creazione di un endpoint VPC di interfaccia per il rilevamento STS OIDC

Puoi creare un endpoint VPC per l'endpoint di rilevamento STS OIDC utilizzando la console Amazon Virtual Private Cloud o il. AWS Command Line Interface Per ulteriori informazioni, consulta Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia nella Amazon VPC User Guide.

Crea un endpoint VPC per il rilevamento STS OIDC utilizzando il seguente nome di servizio:

- `com.amazonaws.region.sts-oidc`

Se abiliti il DNS privato per l'endpoint, puoi inviare richieste `sts-oidc.region.amazonaws.com` e queste verranno risolte alle interfacce di rete dell'endpoint all'interno del tuo VPC.

Invio di richieste tramite l'endpoint

Quando un relying party nel tuo VPC deve verificare un JWT, recupera il documento di scoperta OIDC e i JWKS dall'URL dell'emittente. Il nome host dell'emittente (ad esempio,`uuid.tokens.sts.global.api.aws`) utilizza un dominio diverso da quello dell'endpoint (), quindi per impostazione predefinita non si risolve nell'endpoint `sts-oidc.region.amazonaws.com` VPC.

Per indirizzare le richieste per il nome host dell'emittente attraverso l'endpoint, crea una zona ospitata privata di Amazon Route 53 per il nome host dell'emittente e aggiungi un record alias che punti all'endpoint VPC. Dopo aver creato la zona ospitata, le risorse nel VPC risolvono il nome host dell'emittente nelle interfacce di rete dell'endpoint. Le librerie OIDC standard possono quindi recuperare i documenti di rilevamento senza modifiche al codice. Per ulteriori informazioni sulla creazione di una zona ospitata privata, consulta la sezione relativa all'utilizzo di zone ospitate private nella Guida per gli sviluppatori di Amazon Route 53.

> **Note**
>
> Il servizio endpoint è regionale, ma un singolo endpoint serve le richieste di rilevamento per qualsiasi UUID dell'emittente nella partizione. Crea endpoint nelle regioni in cui vengono eseguiti i verificatori; non ne hai bisogno uno per account dell'emittente.

Controlla l'accesso a AWS STS con politiche per gli endpoint VPC

Quando crei un endpoint VPC di interfaccia per Servizio di token di sicurezza AWS (AWS STS), puoi allegare una policy sugli endpoint. La policy controlla quali principali possono utilizzare l'endpoint e quali AWS STS azioni possono eseguire. Se non alleghi una policy, l'endpoint utilizza la policy predefinita che consente l'accesso illimitato a tutte le AWS STS azioni per tutti i principali.

Le policy degli endpoint VPC non concedono autorizzazioni da sole. Agiscono come un limite aggiuntivo che funziona insieme ad altre politiche. Sia la policy sugli endpoint che le politiche applicabili del chiamante devono consentire l'esito positivo di una richiesta.

Per ulteriori informazioni sulle politiche degli endpoint VPC, consulta Control access to VPC endpoint using endpoint policy nella Amazon VPC User Guide.

Argomenti

- Policy di endpoint VPC predefinita
- Considerazioni importanti per AWS STS Policy di endpoint VPC
- Chiavi di condizione disponibili per AWS STS Policy di endpoint VPC
- Esempio: consenti tutto AWS STS azioni per la tua organizzazione
- Esempio: consenti tutto AWS STS azioni per account specifici
- Esempio: limita a uno specifico AWS STS actions
- Esempio: negare l'accesso a soggetti esterni all'organizzazione pur consentendo la federazione

Policy di endpoint VPC predefinita

Se non alleghi una policy personalizzata quando crei l'endpoint, allega la seguente policy predefinita. AWS Questa policy consente l'accesso illimitato all'endpoint.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*"
        }
    ]
```

```
}
```

Per limitare l'accesso all'endpoint, allega una policy personalizzata per gli endpoint.

Considerazioni importanti per AWS STS Policy di endpoint VPC

AWS STS gestisce le richieste provenienti da due tipi di chiamanti fondamentalmente diversi. La tua policy sugli endpoint VPC deve tenere conto di entrambi i tipi per evitare di bloccare involontariamente le richieste legittime.

AWS Principal autenticati

Utenti IAM e ruoli IAM che firmano le richieste con AWS Signature Version 4 (Sigv4). Questi chiamanti dispongono di chiavi di condizione standard come `aws:PrincipalOrgIDaws:PrincipalAccount`, e `aws:PrincipalArn` disponibili nel contesto della richiesta.

Chiamanti federati

Principal SAML 2.0 e OpenID Connect (OIDC) che chiamano o. `AssumeRoleWithSAML` `AssumeRoleWithWebIdentity` Questi chiamanti si autenticano con asserzioni SAML o JSON Web Tokens (JWT), non con firme SIGv4. Poiché non hanno un' AWS identità al momento della richiesta, il contesto della richiesta non include chiavi di condizione basate sui principali come, e. `aws:PrincipalOrgID` `aws:PrincipalAccount` `aws:PrincipalArn`

> **Important**
>
> Se la tua policy sugli endpoint VPC si basa esclusivamente sulla `aws:PrincipalOrgID` possibilità di accesso, le chiamate federate `AssumeRoleWithSAML` e `AssumeRoleWithWebIdentity` le chiamate vengono implicitamente negate perché la chiave di condizione è assente per i non principali.AWS

In che modo AWS STS valuta le politiche degli endpoint VPC per i chiamanti federati

Quando un chiamante federato richiama `AssumeRoleWithSAML` o `AssumeRoleWithWebIdentity` tramite un endpoint VPC, vale quanto segue:

- Il chiamante non è un principale. AWS Chiavi di condizione come `aws:PrincipalOrgIDaws:PrincipalAccount`, e non `aws:PrincipalArn` sono disponibili nel contesto della richiesta.
- I chiamanti federati non dispongono della chiave di condizione `aws:PrincipalIsAWSService` nel contesto della richiesta.
- Il ruolo assunto è una AWS risorsa. Resource-based le chiavi di condizione come `aws:ResourceOrgID` e `aws:ResourceAccount` sono disponibili e si riferiscono al ruolo di destinazione.
- La politica di fiducia del ruolo rimane la porta di autorizzazione principale per l'accesso federato. La politica degli endpoint VPC fornisce un limite aggiuntivo a livello di rete.

Consigliamo di utilizzare `aws:ResourceOrgID` o `aws:ResourceAccount` quando si scrivono dichiarazioni di policy sugli endpoint VPC che devono essere applicate ai chiamanti federati, poiché questi chiamanti non dispongono di chiavi di condizione basate sull'indirizzo principale.

Chiavi di condizione disponibili per AWS STS Policy di endpoint VPC

La tabella seguente mostra le chiavi di condizione comunemente utilizzate e la loro disponibilità nel contesto della richiesta per ogni tipo di chiamante quando si effettuano richieste tramite un endpoint AWS STS VPC. Sono disponibili chiavi di condizione aggiuntive oltre a quelle elencate qui.

Disponibilità della chiave condizionale per tipo di chiamante

Chiave di condizione	Responsabili autenticati AWS	Chiamanti federati	Description
`aws:PrincipalOrgID`	Sì	No	L'ID organizzativo del principale chiamante
`aws:PrincipalAccount`	Sì	No	L'ID dell'account del principale chiamante
`aws:PrincipalArn`	Sì	No	L'ARN del principale chiamante

Chiave di condizione	Responsabili autenticati AWS	Chiamanti federati	Description
`aws:PrincipalIsAWSService`	Sì (risulta falso)	No (la chiave è assente)	Se il chiamante è un AWS titolare del servizio
`aws:ResourceOrgID`	Sì	Sì	L'ID dell'organizzazione dell'account proprietario della risorsa richiesta
`aws:ResourceAccount`	Sì	Sì	L'ID dell'account proprietario della risorsa richiesta

> **Note**
>
> Per AWS i principali autenticati (utenti e ruoli IAM), `aws:PrincipalIsAWSService` è presente nel contesto della richiesta e viene valutato come falso. Per i chiamanti federati, questa chiave è completamente assente dal contesto della richiesta. Una condizione che verifica non `"Bool": {"aws:PrincipalIsAWSService": "false"}` corrisponde ai chiamanti federati perché la chiave non è presente.

Esempio: consenti tutto AWS STS azioni per la tua organizzazione

La seguente politica sugli endpoint limita l'endpoint AWS STS VPC alla tua organizzazione supportando al contempo l'accesso federato. Consente tutte le AWS STS azioni per i principali autenticati dell'organizzazione e consente separatamente ai chiamanti federati di assumere ruoli all'interno dell'organizzazione. I chiamanti federati (`AssumeRoleWithSAML`e`AssumeRoleWithWebIdentity`) richiedono una dichiarazione separata perché non ne hanno una nel contesto della richiesta. `aws:PrincipalOrgID`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOrganizationPrincipals",
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": "*"
            },
            "Action": "sts:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "o-exampleorgid"
                }
            }
        },
        {
            "Sid": "AllowFederatedAssumeRole",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "sts:AssumeRoleWithSAML",
                "sts:AssumeRoleWithWebIdentity"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceOrgID": "o-exampleorgid"
                }
            }
        }
    ]
}
```

La prima istruzione utilizza `"Principal": {"AWS": "*"}` with `aws:PrincipalOrgID` per consentire l'autenticazione AWS dei principali dell'organizzazione. La seconda istruzione viene utilizzata `"Principal": "*"` per abbinare i chiamanti federati e limita il ruolo di destinazione utilizzato dall'organizzazione. `aws:ResourceOrgID` La politica di fiducia del ruolo rimane il controllo principale per il quale le identità federate possono assumere quali ruoli.

Per ulteriori informazioni sull'implementazione dei controlli perimetrali di rete, consulta Creazione di un perimetro di dati AWS e gli esempi di policy sul perimetro dei dati sul sito Web. GitHub

Esempio: consenti tutto AWS STS azioni per account specifici

La seguente politica sugli endpoint consente tutte le AWS STS azioni per i committenti in account specifici. Utilizza questa politica quando i tuoi account non fanno parte di un'organizzazione.

I chiamanti federati possono inserire una dichiarazione separata perché non ne hanno una `aws:PrincipalAccount` nel contesto della richiesta.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSpecificAccountPrincipals",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "sts:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": [
                        "111122223333",
                        "444455556666"
                    ]
                }
            }
        },
        {
            "Sid": "AllowFederatedAssumeRole",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "sts:AssumeRoleWithSAML",
                "sts:AssumeRoleWithWebIdentity"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": [
                        "111122223333",
                        "444455556666"
                    ]
                }
            }
        }
    ]
}
```

La prima istruzione utilizza `"Principal": {"AWS": "*"}` with `aws:PrincipalAccount` per consentire l'autenticazione dei AWS principali dagli account specificati. La seconda istruzione viene utilizzata `"Principal": "*"` per abbinare i chiamanti federati e limita il ruolo di destinazione agli stessi account utilizzati. `aws:ResourceAccount` La politica di fiducia del ruolo rimane il controllo principale per il quale le identità federate possono assumere quali ruoli.

Esempio: limita a uno specifico AWS STS actions

La seguente policy sugli endpoint è valida solo `AssumeRole` per i principali autenticati e `AssumeRoleWithSAML` per i chiamanti federati, entrambi riservati ai ruoli nell'organizzazione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAssumeRoleByOrgPrincipals",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "sts:AssumeRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "o-exampleorgid"
                }
            }
        },
        {
            "Sid": "AllowSAMLFederation",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "sts:AssumeRoleWithSAML",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceOrgID": "o-exampleorgid"
                }
            }
        }
    ]
}
```

Questa policy blocca altre AWS STS azioni come`GetCallerIdentity`, `GetSessionToken` e attraverso questo endpoint. `AssumeRoleWithWebIdentity` Modifica gli `Action` elementi in base alle tue esigenze.

Esempio: negare l'accesso a soggetti esterni all'organizzazione pur consentendo la federazione

La seguente politica sugli endpoint utilizza un rifiuto esplicito per bloccare i principali autenticati esterni all'organizzazione, preservando al contempo l'accesso per i chiamanti federati. Questo approccio inizia con un'ampia autorizzazione e aggiunge una dichiarazione di rifiuto mirata.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAll",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*"
        },
        {
            "Sid": "DenyNonOrgPrincipals",
            "Effect": "Deny",
            "Principal": {
                "AWS": "*"
            },
            "Action": "sts:*",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:PrincipalOrgID": "o-exampleorgid"
                }
            }
        }
    ]
}
```

L'istruzione deny utilizza`"Principal": {"AWS": "*"}`, che la limita solo ai principali AWS autenticati. I chiamanti federati (SAML e OIDC) non sono AWS principali e non corrispondono a questo `Principal` elemento, quindi il rifiuto non si applica a loro. Questo approccio evita la

necessità di ricorrere a `Bool` condizioni complesse per individuare eccezioni per i `Null` chiamanti federati.

> **Note**
>
> La prima dichiarazione consente tutte le azioni per tutti i principali. La negazione contenuta nella seconda dichiarazione ha la precedenza per i AWS committenti autenticati esterni all'organizzazione. I chiamanti federati sono consentiti dalla prima istruzione e non sono interessati dalla negazione perché l'`Principal`elemento nell'istruzione di rifiuto non corrisponde a loro.

Usa le chiavi di condizione VPC per controllare l'accesso federato

Gli utenti federati possono assumere ruoli utilizzando `AssumeRoleWithSAML` o `AssumeRoleWithWebIdentity` tramite un endpoint VPC. Quando lo fanno, puoi utilizzare le chiavi di VPC-specific condizione nella politica di fiducia del ruolo o nelle politiche di controllo delle risorse (RCP) per limitare la provenienza di queste richieste. Ciò fornisce un limite a livello di rete oltre ai controlli a livello di identità nella politica di fiducia.

Argomenti

- Come funzionano le chiavi di condizione VPC per le richieste federate
- Esempio: limita la federazione OIDC a un VPC specifico
- Esempio: limita la federazione SAML a un endpoint VPC specifico
- Esempio: limita la federazione OIDC per intervallo IP privato all'interno di un VPC
- Esempio: limita la federazione SAML a più VPC
- Esempio: Combina VPC e condizioni specifiche del provider
- Esempio: utilizzare un RCP per limitare l'accesso federato a un VPC specifico

Come funzionano le chiavi di condizione VPC per le richieste federate

Le chiavi di condizione disponibili nel contesto della richiesta dipendono dal percorso di rete della richiesta federata.

Disponibilità delle chiavi di condizione VPC per percorso di rete

Chiave di condizione	Tramite endpoint VPC	Tramite Internet pubblico	Description
`aws:SourceVpc`	Sì	No	ID VPC attraversato dalla richiesta
`aws:SourceVpcArn`	Sì	No	ARN del VPC attraversato dalla richiesta
`aws:SourceVpce`	Sì	No	ID dell'endpoint VPC attraversato dalla richiesta
`aws:VpcSourceIp`	Sì	No	Indirizzo IP privato del chiamante all'interno del VPC
`aws:SourceIp`	No	Sì	Indirizzo IP pubblico del chiamante

Se la tua policy di fiducia utilizza `aws:SourceVpc` o è contenuta `aws:SourceVpce` in una `Allow` dichiarazione e la richiesta non proviene da un endpoint VPC, la condizione non corrisponde e nega AWS implicitamente la richiesta. `aws:SourceVpcArn` Ciò richiede effettivamente che le richieste federate attraversino l'endpoint VPC per avere successo.

> **Important**
>
> Per le richieste effettuate tramite un endpoint VPC, non `aws:SourceIp` è compilato. Utilizzatelo invece `aws:VpcSourceIp` per le restrizioni IP-based .

Si consiglia di utilizzare politiche `aws:SourceVpcArn` anziché `aws:SourceVpc` attendibili. Gli ID VPC sono unici a livello regionale ma non globale, mentre `aws:SourceVpcArn` includono la regione e l'account, fornendo un'identificazione univoca a livello globale del VPC.

> **Note**
>
> Il valore audience (`aud`) negli esempi OIDC seguenti utilizza`sts.amazonaws.com`, che è l'impostazione predefinita per Amazon EKS. Per altri provider OIDC, sostituiscilo con il

valore di audience configurato per la tua applicazione (ad esempio, un ID client o un URI dell'applicazione).

Esempio: limita la federazione OIDC a un VPC specifico

La seguente politica di fiducia consente a un provider OIDC di assumere il ruolo solo quando la richiesta proviene da un VPC specifico tramite un endpoint VPC.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOIDCFromVpc",
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:oidc-provider/idp.example.com"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "idp.example.com:aud": "sts.amazonaws.com",
                    "aws:SourceVpcArn": "arn:aws:ec2:us-west-2:111122223333:vpc/
vpc-111bbb22"
                }
            }
        }
    ]
}
```

AWS nega implicitamente le richieste dall'esterno del VPC specificato perché la `aws:SourceVpcArn` condizione non corrisponde. Non è necessaria alcuna dichiarazione di negazione esplicita.

Esempio: limita la federazione SAML a un endpoint VPC specifico

La seguente politica di fiducia consente a un provider SAML di assumere il ruolo solo quando la richiesta arriva attraverso un endpoint VPC specifico.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "AllowSAMLFromVpce",
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:saml-provider/ExampleProvider"
            },
            "Action": "sts:AssumeRoleWithSAML",
            "Condition": {
                "StringEquals": {
                    "SAML:aud": "https://signin.aws.amazon.com/saml",
                    "aws:SourceVpce": "vpce-0abcdef1234567890"
                }
            }
        }
    ]
}
```

Using `aws:SourceVpce` instead of `aws:SourceVpc` fornisce un controllo più granulare quando si hanno più endpoint VPC nello stesso VPC e si desidera limitare l'accesso a un endpoint specifico.

Esempio: limita la federazione OIDC per intervallo IP privato all'interno di un VPC

La seguente politica di attendibilità limita la federazione alle richieste provenienti da una sottorete specifica all'interno del VPC. Utilizzalo `aws:VpcSourceIp` per IP-based le restrizioni sulle richieste effettuate tramite un endpoint VPC.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOIDCFromSubnet",
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:oidc-provider/idp.example.com"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "idp.example.com:aud": "sts.amazonaws.com",
                    "aws:SourceVpcArn": "arn:aws:ec2:us-west-2:111122223333:vpc/
vpc-111bbb22"
                },
```

```
                "IpAddress": {
                    "aws:VpcSourceIp": "10.0.1.0/24"
                }
            }
        }
    ]
}
```

> **Important**
>
> Non utilizzare `aws:SourceIp` con intervalli IP privati per le richieste tramite un endpoint VPC. La `aws:SourceIp` chiave non è disponibile per le richieste degli endpoint VPC. Usare invece `aws:VpcSourceIp`.

Esempio: limita la federazione SAML a più VPC

La seguente politica di fiducia consente a un provider SAML di assumere il ruolo di uno qualsiasi dei diversi VPC. Ciò è utile quando l'organizzazione ha più VPC in ambienti o regioni diversi che richiedono l'accesso federato allo stesso ruolo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSAMLFromMultipleVpcs",
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:saml-provider/ExampleProvider"
            },
            "Action": "sts:AssumeRoleWithSAML",
            "Condition": {
                "StringEquals": {
                    "SAML:aud": "https://signin.aws.amazon.com/saml",
                    "aws:SourceVpcArn": [
                        "arn:aws:ec2:us-west-2:111122223333:vpc/vpc-111aaa11",
                        "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-222bbb22"
                    ]
                }
            }
        }
    ]
```

```
}
```

Esempio: Combina VPC e condizioni specifiche del provider

Puoi combinare le chiavi delle condizioni del VPC con condizioni specifiche del provider come audience (`aud`) e subject (`sub`) per un controllo granulare degli accessi. La seguente politica di attendibilità limita l'assunzione del ruolo a un utente federato specifico e richiede che la richiesta provenga da un VPC specifico.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOIDCUserFromVpc",
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:oidc-provider/idp.example.com"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "idp.example.com:aud": "sts.amazonaws.com",
                    "idp.example.com:sub": "user123",
                    "aws:SourceVpcArn": "arn:aws:ec2:us-west-2:111122223333:vpc/
vpc-111bbb22"
                }
            }
        }
    ]
}
```

Ciò combina controlli a livello di identità (limitati a un utente federato specifico) con controlli a livello di rete (limitati a un VPC specifico), fornendo una difesa approfondita.

Esempio: utilizzare un RCP per limitare l'accesso federato a un VPC specifico

La seguente politica di controllo delle risorse (RCP) `AssumeRoleWithSAML` nega le `AssumeRoleWithWebIdentity` richieste a meno che non provengano da un VPC specifico. A differenza delle politiche di fiducia che si applicano per ruolo, una politica di controllo delle risorse (RCP) si applica a tutti i ruoli negli account di destinazione. Ciò fornisce un limite di rete centralizzato.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyFederatedAccessOutsideVpc",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "sts:AssumeRoleWithWebIdentity",
                "sts:AssumeRoleWithSAML"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:SourceVpcArn": "arn:aws:ec2:us-west-2:111122223333:vpc/
vpc-111bbb22"
                },
                "Null": {
                    "aws:SourceVpcArn": "false"
                }
            }
        }
    ]
}
```

La `Null` condizione garantisce che la negazione si applichi solo quando `aws:SourceVpcArn` è presente nel contesto della richiesta (ovvero, la richiesta proviene da un endpoint VPC ma dal VPC sbagliato). Senza di essa, anche le richieste sulla rete Internet pubblica (dove `aws:SourceVpcArn` è assente) verrebbero respinte da. `StringNotEquals` Se rimuovi il `Null` segno di spunta, il rifiuto si applica anche alle richieste che non attraversano alcun endpoint VPC, incluse le richieste sulla rete Internet pubblica. Consigliamo di eseguire test approfonditi prima di rimuovere il `Null` controllo negli ambienti di produzione.

AWS servizi supportati da IAM

I AWS servizi elencati di seguito sono raggruppati in ordine alfabetico e includono informazioni sulle funzionalità IAM che supportano:

- Servizio: puoi scegliere il nome di un servizio per visualizzare la AWS documentazione sull'autorizzazione e l'accesso IAM a quel servizio.
- Operazioni: è possibile specificare singole operazioni in una policy. Se il servizio non supporta questa funzionalità, vengono selezionate Tutte le operazioni nell'editor visivo. In un documento di

policy JSON, è necessario utilizzare * nell'elemento `Action`. Per un elenco delle azioni in ogni servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.

- Resource-level autorizzazioni: è possibile utilizzare gli ARN per specificare singole risorse nella politica. Se il servizio non supporta questa funzionalità, viene selezionata l'opzione Tutte le risorse nell'editor visivo della policy. In un documento di policy JSON, è necessario utilizzare * nell'elemento `Resource`. Alcune operazioni, ad esempio le operazioni `List*`, non supportano la specifica di un ARN perché sono progettate per restituire più risorse. Se un servizio supporta questa funzionalità per alcune risorse ma non per altre, il limite viene indicato con la dicitura Partial (Parziale) nella tabella. Per ulteriori informazioni, consulta la documentazione per quel servizio.
- Resource-based politiche: è possibile allegare politiche basate sulle risorse a una risorsa all'interno del servizio. Resource-based le policy includono un `Principal` elemento per specificare quali identità IAM possono accedere a quella risorsa. Per ulteriori informazioni, consulta Identity-based politiche e politiche basate sulle risorse.
- ABAC (autorizzazione basata su tag): per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento di condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`. Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Partial (Parziale).

  Per ulteriori informazioni sulla definizione delle autorizzazioni basate su attributi quali i tag, consulta Definire le autorizzazioni basate su attributi con l'autorizzazione ABAC. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC).
- Credenziali temporanee: puoi utilizzare credenziali a breve termine ottenute quando accedi utilizzando IAM Identity Center, account access manager, cambi ruolo nella console o generate utilizzando AWS STS l' AWS CLI API or. AWS È possibile accedere ai servizi con un valore No solo utilizzando le credenziali utente IAM a lungo termine. Questo include un nome utente e una password o le chiavi di accesso utente. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM.
- Service-linked ruoli: un ruolo collegato ai servizi è un tipo speciale di ruolo di servizio che concede al servizio l'autorizzazione ad accedere alle risorse di altri servizi per tuo conto. Selezionare il collegamento Sì o Parziale per visualizzare la documentazione per i servizi che supportano questi ruoli. Questa colonna non indica se il servizio utilizza ruoli di servizio standard. Per ulteriori informazioni, consulta Service-linked ruoli.

- Ulteriori informazioni: se un servizio non supporta completamente una funzionalità, è possibile esaminare le note a piè di pagina per visualizzare le limitazioni e i collegamenti alle informazioni correlate.

Servizi supportati da IAM

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Gestore degli accessi all'account	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	Sì
Gestione dell'account AWS	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Consigli d'azione	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Activate Console	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Operazioni Amazon AI	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Amplify Amministratore	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Amplify	✓ Sì	✓ Sì	✗ No	⚠ Parziale	✓ Sì	✗ No
AWS Amplify Generatore di interfacce utente	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
API di Apache Kafka per i cluster di Amazon MSK	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Gateway Amazon API	✓ Sì	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì
Gestione di Gateway Amazon API	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Gestione di Gateway Amazon API V2	Sì	Sì	No	Sì	Sì	No
AWS App Studio	Sì	No	No	No	Sì	No
AWS App2Container	Sì	No	No	No	Sì	No
AWS AppConfig	Sì	Sì	No	Sì	Sì	No
AWS AppFabric	Sì	Sì	No	Sì	Sì	No
Amazon AppFlow	Sì	Sì	No	Sì	Sì	No
Amazon AppIntegrations	Sì	Sì	No	Sì	Sì	Sì
Application Auto Scaling	Sì	Sì	No	Sì	Sì	Sì
AWS Applicazione Discovery Arsenal	Sì	No	No	No	Sì	No
AWS Application Discovery Service	Sì	No	No	No	Sì	Sì
AWS Transform MGN	Sì	Sì	No	Sì	Sì	Sì
Amazon Application Recovery Controller - Spostamento zonale	Sì	Sì	No	No	Sì	No
AWS Servizio di trasformazione delle applicazioni	Sì	No	No	No	Sì	No
AWS App Mesh	Sì	Sì	No	Sì	Sì	Sì
AWS App Mesh Anteprima	Sì	Sì	No	No	Sì	Sì
AWS App Runner	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Amazon AppStream 2.0	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS AppSync	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Cambio Regione di Amazon ARC	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Artifact	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Amazon Athena	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Audit Manager	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Aurora DSQL	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Auto Scaling	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✓ Sì
AWS Scambio di dati B2B	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Backup	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì
AWS Backup Gateway	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Backup Cerca	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Backup archiviazione	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Batch	✓ Sì	⚠ Parziale	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Bedrock	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Bedrock AgentCore	✓ Sì	✓ Sì	✗ No	⚠ Parziale	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Gestione costi e fatturazione AWS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Dashboard di gestione costi e fatturazione AWS	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Gestione costi e fatturazione AWS Esportazioni di dati	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Gestione costi e fatturazione AWS Calcolatore dei prezzi	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Gestione costi e fatturazione AWS Azioni consigliate	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Billing Conductor	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Braket	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Servizio di bilancio	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Certificate Manager (ACM)	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Q Developer nelle applicazioni di chat	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Chime	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Clean Rooms	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Clean Rooms ML	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Client VPN	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✓ Sì
AWS Cloud9	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Cloud AWS API di controllo	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Directory del cloud Amazon	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
CloudFormation	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon CloudFront	✓ Sì	✓ Sì	✗ No	⚠ Parziale	✓ Sì	✓ Sì
Amazon CloudFront KeyValueStore	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS CloudHSM	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Cloud Map	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon CloudSearch	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS CloudShell	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS CloudTrail	✓ Sì	✓ Sì	⚠ Parziale (Informazioni)	✓ Sì	✓ Sì	✓ Sì
AWS CloudTrail Dati	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon CloudWatch	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	⚠ Parziale (Informazioni)
Informazioni approfondite sulle CloudWatch applicazioni Amazon	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Segnali CloudWatch applicativi Amazon	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon CloudWatch Internet Monitor	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
CloudWatch Registri Amazon	✓ Sì	✓ Sì	✓ Sì	⚠ Parziale	✓ Sì	✓ Sì
Monitor sintetico di CloudWatch rete Amazon	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon CloudWatch Observability Access Manager	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Servizio Amazon CloudWatch Observability Admin	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon CloudWatch RUM	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon CloudWatch Synthetics	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS CodeArtifact	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✗ No
AWS CodeBuild	✓ Sì	✓ Sì	✓ Sì (Informazioni)	⚠ Parziale (Informazioni)	✓ Sì	✗ No
Amazon CodeCatalyst	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS CodeCommit	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS CodeConnections	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS CodeDeploy	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS CodeDeploy servizio di comandi host sicuro	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Amazon CodeGuru Profiler	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
CodeGuru Revisore Amazon	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
CodeGuru Sicurezza Amazon	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS CodePipeline	✓ Sì	⚠ Parziale	✗ No	✓ Sì	✓ Sì	✗ No
AWS CodeStar	✓ Sì	⚠ Parziale	✗ No	✓ Sì	✓ Sì	✗ No
AWS CodeStar Connessioni	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Notifiche AWS CodeStar	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon CodeWhisperer	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Cognito	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Cognito Sync	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✓ Sì
Pool di utenti Amazon Cognito	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Comprehend	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Comprehend Medical	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Compute Optimizer	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✓ Sì
AWS Config	✓ Sì	⚠Parziale (Informazioni)	✗ No	✓ Sì	✓ Sì	✓ Sì

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Cliente Amazon Connect	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Casi di clienti Amazon Connect	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Profili dei clienti Amazon Connect	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Campagne in uscita per i clienti Amazon Connect	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
ID vocale del cliente Amazon Connect	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Connect Health	✓ Sì	✓ Sì	✗ No	⚠ Parziale	✓ Sì	✗ No
AWS Console Mobile Application	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Fatturazione consolidata	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Catalogo di controllo AWS	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Control Tower	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Cost and Usage Report	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Cost Explorer	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Centrale ottimizzazione costi AWS	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Servizio di verifica clienti AWS	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS Database Migration Service	Sì	Sì	No (Informazioni)	Sì	Sì	Sì
Database Query Metadata Service	Sì	No	No	No	Sì	No
AWS Data Exchange	Sì	Sì	No	Sì	Sì	Sì
Amazon Data Lifecycle Manager	Sì	Sì	No	Sì	Sì	No
AWS Data Pipeline	Sì	Sì	No	Parziale	Sì	No
AWS DataSync	Sì	Sì	No	Sì	Sì	Sì
Amazon DataZone	Sì	Sì	No	Sì	Sì	No
AWS Deadline Cloud	Sì	Sì	No	Sì	Sì	No
AWS DeepRacer	Sì	Sì	No	Sì	Sì	Sì
Amazon Detective	Sì	Sì	No	Sì	Sì	No
AWS Device Farm	Sì	Sì	No	Sì	Sì	Sì
Amazon DevOps Guru	Sì	Sì	No	No	Sì	Sì
Strumenti di diagnostica AWS	Sì	Sì	No	Sì	Sì	No
AWS Direct Connect	Sì	Sì	No	Sì	Sì	Sì
Servizio di directory AWS	Sì	Sì	No	Sì	Sì	Sì
AWS Dati del Directory Service	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Cluster elastici Amazon DocumentDB	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon DynamoDB Accelerator (DAX)	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✓ Sì
Amazon DynamoDB	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì
Amazon Elastic Compute Cloud (Amazon EC2)	✓ Sì	⚠ Parziale	✗ No	✓ Sì	✓ Sì	⚠ Parziale (Informazioni)
Amazon EC2 Auto Scaling	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
EC2 Image Builder	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon EC2 Instance Connect	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Elastic Beanstalk	✓ Sì	⚠ Parziale	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Elastic Block Store (Amazon EBS)	✓ Sì	⚠ Parziale	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Elastic Container Registry (Amazon ECR)	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì
Amazon Elastic Container Registry Pubblico (Amazon ECR pubblico)	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Elastic Container Service (Amazon ECS)	✓ Sì	⚠ Parziale (Informazioni)	✗ No	✓ Sì	✓ Sì	✓ Sì

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS Elastic Disaster Recovery	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Elastic File System (Amazon EFS)	✓ Sì	✓ Sì	✓ Sì	⚠ Parziale	✓ Sì	✓ Sì
Amazon Elastic Kubernetes Service (Amazon EKS)	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Auth Amazon Elastic Kubernetes Service (Amazon EKS)	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Elastic Load Balancing	✓ Sì	⚠ Parziale	✗ No	⚠ Parziale	✓ Sì	✓ Sì
Amazon Elastic VMware Service	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon ElastiCache	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Servizio di attivazione software e dispositivi elementari	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Apparecchiature e software elementali	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Elemental MediaConnect	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✓ Sì
AWS Elemental MediaConvert	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Elemental MediaLive	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Elemental MediaPackage	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	⚠ Parziale(Informazioni)

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS Elemental MediaPackage V2	Sì	Sì	No	Sì	Sì	No
AWS Elemental MediaPackage VOD	Sì	Sì	No	Sì	Sì	Parziale(Informazioni)
AWS Elemental MediaTailor	Sì	Sì	No	Sì	Sì	Sì
AWS Casi Elemental Support	Sì	Sì	No	Sì	Sì	No
AWS Contenuti di Elemental Support	Sì	No	No	No	Sì	No
Amazon EMR	Sì	Sì	No	Sì	Sì	Sì
Amazon EMR su EKS	Sì	Sì	No	Sì	Sì	Sì
Amazon EMR Serverless	Sì	Sì	No	Sì	Sì	Sì
AWS End User Messaging SMS e Voice V2	Sì	Sì	No	Sì	Sì	Sì
AWS Messaggistica sociale per utenti finali	Sì	Sì	No	Sì	Sì	Sì
AWS Entity Resolution	Sì	Sì	Sì	Sì	Sì	No
Amazon EventBridge	Sì	Sì	Sì	Sì	Sì	No
EventBridge Tubi Amazon	Sì	Sì	No	Sì	Sì	No
Amazon EventBridge Scheduler	Sì	Sì	No	Sì	Sì	No
EventBridge Schemi Amazon	Sì	Sì	Sì	Sì	Sì	No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS Servizio di iniezione dei guasti	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon FinSpace	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon FinSpace API	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Firewall Manager	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	⚠ Parziale
Fleet Hub for AWS IoT Device Management	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Forecast	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Fraud Detector	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
FreeRTOS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Livello gratuito	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Amazon FSx	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
GameLift Server Amazon	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon GameLift Stream	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Global Accelerator	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Glue	✓ Sì	✓ Sì	⚠ Parziale	⚠ Parziale	✓ Sì	✗ No
AWS Glue DataBrew	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Ground Station	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Ground Truth Labeling	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Amazon GuardDuty	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Health API e notifiche	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS HealthImaging	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS HealthLake	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS HealthOmics	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Centro identità AWS IAM	✓ Sì	✓ Sì	✗ No	⚠ Parziale	✓ Sì	✓ Sì
Directory di IAM Identity Center	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Archivio identità di IAM Identity Center	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Servizio OIDC IAM Identity Center	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Identity and Access Management (IAM)	✓ Sì	✓ Sì	⚠ Parziale (Informazioni)	⚠ Parziale (Informazioni)	⚠ Parziale (Informazioni)	✗ No
AWS Identity and Access Management e Access Analyzer	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	⚠ Parziale
AWS Identity and Access Management Ruoli ovunque	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Autenticazione di Identity Store	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Sincronizzazione identità	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS Import/Export	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Amazon Inspector	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Inspector Classic	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✓ Sì
Amazon InspectorScan	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Amazon Interactive Video Service	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Interactive Video Service Chat	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Fatturazione AWS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS IoT Analytics	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS IoT	✓ Sì	✓ Sì	⚠ Parziale (Informazioni)	✓ Sì	✓ Sì	✗ No
AWS IoT Core Consulente per dispositivi	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS IoT Tester per dispositivi	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS IoT Events	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS IoT FleetWise	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS IoT Greengrass	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS IoT Greengrass V2	✓ Sì	✓ Sì	✗ No	⚠ Parziale	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS IoT Jobs DataPlane	Sì	Sì	No	No	Sì	No
AWS IoT Servizio di integrazioni gestite	Sì	Sì	No	Sì	Sì	Sì
AWS IoT SiteWise	Sì	Sì	No	Sì	Sì	Sì
AWS IoT TwinMaker	Sì	Sì	No	Sì	Sì	Sì
Wireless AWS IoT	Sì	Sì	No	Sì	Sì	No
Amazon Kendra	Sì	Sì	No	Sì	Sì	No
Classificazione intelligente di Amazon Kendra	Sì	Sì	No	Sì	Sì	No
AWS Key Management Service (AWS KMS)	Sì	Sì	Sì	Sì	Sì	Sì
Amazon Keyspaces (per Apache Cassandra)	Sì	Sì	No	Sì	Sì	Sì
Servizio gestito da Amazon per Apache Flink	Sì	Sì	No	Sì	Sì	No
Servizio gestito da Amazon per Apache Flink V2	Sì	Sì	No	Sì	Sì	No
Amazon Data Firehose	Sì	Sì	No	Sì	Sì	No
Flusso di dati Amazon Kinesis	Sì	Sì	Sì	Sì	Sì	No
Amazon Kinesis Video Streams	Sì	Sì	No	Sì	Sì	No
AWS Lake Formation	Sì	No	No	No	Sì	Sì

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS Lambda	Sì	Sì	Sì	Parziale (Informazioni)	Sì	Parziale (Informazioni)
AWS Launch Wizard	Sì	No	No	No	Sì	No
Amazon Lex	Sì	Sì	No	Sì	Sì	Sì
Amazon Lex V2	Sì	Sì	Sì	Sì	No	Sì
AWS License Manager	Sì	Sì	No	Sì	Sì	Sì
AWS License Manager Gestore di abbonamenti Linux	Sì	No	No	No	Sì	No
AWS License Manager Sottoscrizioni utente	Sì	No	No	No	Sì	Sì
Amazon Lightsail	Sì	Parziale (Informazioni)	No	Parziale (Informazioni)	Sì	Sì
Servizio di posizione Amazon	Sì	Sì	No	Sì	Sì	No
Mappe del servizio di posizione Amazon	Sì	Sì	No	No	Sì	No
Luoghi del servizio di posizione Amazon	Sì	Sì	No	No	Sì	No
Route del servizio di posizione Amazon	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Amazon Lookout per le apparecchiature	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Lookout per le metriche	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Machine Learning	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Amazon Macie	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Modernizzazione del mainframe AWS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Modernizzazione del mainframe AWS Test delle applicazioni	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Blockchain gestita da Amazon	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Query su Blockchain gestita da Amazon	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Grafana gestito da Amazon	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Managed Service per Prometheus	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Managed Streaming per Apache Kafka (MSK)	✓ Sì	✓ Sì	⚠ Parziale (Informazioni)	✓ Sì	✓ Sì	✓ Sì
Amazon Managed Streaming per Kafka Connect	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Flussi di lavoro gestiti da Amazon per Apache Airflow	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Marketplace AWS	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✓ Sì
Marketplace AWS Catalogo	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Marketplace Commerce Analytics	✓ Sì	✗ No	✗ No	✗ No	✗ No	✗ No
Servizio di implementazione Marketplace AWS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Marketplace AWS Individuazione	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Marketplace Entitlement Service	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Marketplace AWS Servizio di creazione di immagini	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Portale di gestione Marketplace AWS	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Marketplace Metering Service	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Marketplace AWS Marketplace privato	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Marketplace AWS Integrazione dei sistemi di approvvigionamento	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Marketplace AWS Creazione di report	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Marketplace AWS Rapporti sui venditori	Sì	Sì	No	No	Sì	No
Marketplace AWS Informazioni sui fornitori	Sì	Sì	No	Sì	Sì	No
Amazon MediaImport	Sì	No	No	No	No	No
Amazon MemoryDB	Sì	Sì	No	Sì	Sì	Sì
Amazon Message Delivery Service	Sì	No	No	No	Sì	No
Amazon Message Gateway Service	Sì	No	No	No	Sì	No
AWS Microservice Extractor for .NET	Sì	No	No	No	Sì	No
AWS Crediti del Migration Acceleration Program	Sì	Sì	No	No	Sì	No
AWS Migration Hub	Sì	Sì	No	No	Sì	Sì
AWS Migration Hub Orchestratore	Sì	Sì	No	Sì	Sì	Sì
AWS Migration Hub Refactor Spaces	Sì	Sì	Sì	Sì	Sì	Sì
AWS Migration Hub Strategy Recommendations	Sì	No	No	No	Sì	Sì
Amazon Monitron	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Amazon MQ	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Neptune	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✓ Sì
Multi-partyapprovazione	✓ Sì	✓ Sì	✗ No	⚠Parziale (Informazioni)	✓ Sì	✗ No
Analisi di Amazon Neptune	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Network Firewall	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Monitoraggio flusso di rete	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Network Manager	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì(Informazioni)
AWS Network Manager Chat	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Amazon Nimble Studio	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon One Enterprise	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon OpenSearch	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
OpenSearchIngestione di Amazon	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon OpenSearch Serverless	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
OpenSearch Servizio Amazon	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì
Database Oracle@AWS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS OpsWorks	Sì	Sì	No	No	Sì	No
AWS OpsWorks Gestione della configurazione	Sì	Sì	No	No	Sì	No
AWS Organizations	Sì	Sì	Sì	Sì	No	Sì
AWS Outposts	Sì	Sì	No	Sì	Sì	Sì
AWS Parallel Computing Service	Sì	Sì	No	Sì	Sì	Sì
AWS Gestione degli account Partner Central	Sì	No	No	No	Sì	No
AWS Vendita con Partner Central	Sì	Sì	No	Sì	Sì	No
AWS Payment Cryptography	Sì	Sì	No	Sì	Sì	No
AWS Pagamenti	Sì	No	No	No	Sì	No
AWS Approfondimenti sulle prestazioni	Sì	Sì	No	Sì	Sì	No
Amazon Personalize	Sì	Sì	No	No	Sì	No
Amazon Pinpoint	Sì	Sì	No	Sì	Sì	No
Servizio e-mail Amazon Pinpoint	Sì	Sì	No	Sì	Sì	No
Servizio di SMS e messaggi vocali Amazon Pinpoint	Sì	No	No	No	Sì	No
Amazon Polly	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Listino prezzi AWS	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS 5G privato	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Private CA Connettore per Active Directory	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Private CA Connector per SCEP	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Autorità di certificazione privata (AWS Private CA)	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✗ No
AWS PrivateLink	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Proton	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Console per gli ordini di acquisto	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Q Business	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
App Q Amazon Q Business	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Q Developer	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Q in Connect	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Database Amazon Quantum Ledger (Amazon QLDB)	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Quick	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
API dati di Amazon RDS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Amazon RDS IAM Authentication	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Cestino di riciclaggio	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Redshift	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
API dati di Amazon Redshift	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Amazon Redshift Serverless	✓ Sì	✓ Sì	✓Sì (Informazioni)	✓ Sì	✓ Sì	✗ No
Amazon Rekognition	✓ Sì	✓ Sì	⚠Parziale (Informazioni)	✓ Sì	✓ Sì	✗ No
Amazon Relational Database Service (Amazon RDS) (Info)	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS re:Post Privata	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Resilience Hub	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Resource Access Manager (AWS RAM)	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Esploratore di risorse AWS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Resource Groups	✓ Sì	✓ Sì	✗ No	✓ Sì	⚠Parziale (Informazioni)	✓ Sì
AWS Resource Groups Tagging API	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Amazon RHEL Knowledgebase Portal	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS RoboMaker	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Route 53	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Domini Amazon Route 53	✓ Sì	✗ No	✗ No	✗ No	✗ No	✗ No
Amazon Route 53 Profiles	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Cluster di ripristino di Amazon Route 53	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Configurazione dei controlli di ripristino Amazon Route 53	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Preparazione al ripristino di Amazon Route 53	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Route 53 Resolver	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Tessuto RTB	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon S3 Express	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✗ No
Amazon Glacier	✓ Sì	✓ Sì	✓ Sì	✗ No	✓ Sì	⚠ Parziale
Tabelle di Amazon S3	✓ Sì	✓ Sì	✓ Sì	✗ No	✓ Sì	✗ No
Amazon S3 Vectors	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Amazon SageMaker AI	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	⚠ Parziale(Informazioni)

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Assistente di data science di Amazon SageMaker AI	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Funzionalità geospaziali di Amazon SageMaker AI	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon SageMaker AI con MLFlow	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS Savings Plans	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Secrets Manager	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✗ No
AWS Security Hub CSPM	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Risposta agli incidenti di sicurezza di AWS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Security Lake	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✓ Sì
Servizio di token di sicurezza AWS (AWS STS)	✓ Sì	⚠ Parziale (Informazioni)	✗ No	✓ Sì	⚠ Parziale (Informazioni)	✗ No
AWS Serverless Application Repository	✓ Sì	✓ Sì	✓ Sì	✗ No	✓ Sì	✗ No
AWS Service Catalog	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Servizio per la gestione delle funzionalità dell'esperienza utente AWS della console	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Service Quotas	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS Shield	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Shield Network Security Director	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Signer	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Sign-In	✓ Sì	✗ No	✓ Sì	✗ No	✓ Sì	✗ No
Amazon SimpleDB	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Amazon Simple Email Service - Gestione posta	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon Simple Email Service (Amazon SES) v2	✓ Sì	⚠ Parziale (Informazioni)	✓ Sì	✓ Sì	⚠ Parziale (Informazioni)	✓ Sì
Amazon Simple Notification Service (Amazon SNS)	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✗ No
Amazon Simple Queue Service (Amazon SQS)	✓ Sì	✓ Sì	✓ Sì	✓ Parziale	✓ Sì	✗ No
Amazon Simple Storage Service (Amazon S3)	✓ Sì	✓ Sì	✓ Sì	⚠ Parziale (Informazioni)	✓ Sì	⚠ Parziale (Informazioni)
Oggetto Amazon Simple Storage Service (Amazon S3) Lambda	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Amazon Simple Storage Service (Amazon S3) su AWS Outposts	✓ Sì	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì
Amazon Simple Workflow Service (Amazon SWF)	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS SimSpaceWeaver	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Site-to-Site VPN	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✓ Sì
AWS Snowball Edge	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Snowball Edge Edge	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Snowball Edge Device Management	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS SQL Workbench	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Step Functions	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Gateway di archiviazione AWS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Catena di approvvigionamento di AWS	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Supporto AWS	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✓ Sì
AWS Support App in Slack	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Supporto AWS Console	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Supporto AWS Piani	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Sostenibilità	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS Systems Manager	✓ Sì	✓ Sì	⚠ Parziale	✓ Sì	✓ Sì	✓ Sì
AWS Systems Manager per SAP	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Systems Manager GUI Connect	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Strumento di gestione degli incidenti AWS Systems Manager	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì	✓ Sì
Strumento di gestione degli incidenti AWS Systems Manager Contatti	✓ Sì	✓ Sì	✓ Sì	✗ No	✓ Sì	✗ No
AWS Systems Manager Configurazione rapida	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Editor di tag	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Impostazioni fiscali	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
AWS Telco Network Builder	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Textract	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Amazon Timestream	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon Timestream Influxdb	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS API Tiros (per Reachability Analyzer)	✓ Sì	✗ No	✗ No	✗ No	✗ No	✗ No

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
Amazon Transcribe	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Transfer Family	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Transform (Trasforma)	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Amazon Translate	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Trusted Advisor	⚠ Parziale (Informazioni)	✓ Sì	✗ No	✗ No	⚠ Parziale	✓ Sì
AWS Notifiche per gli utenti	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS Notifiche utente e contatti	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Sottoscrizioni degli utenti	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Accesso verificato da AWS	✓ Sì	✗ No	✗ No	✗ No	✓ Sì	✗ No
Autorizzazioni verificate da Amazon	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Amazon Virtual Private Cloud (Amazon VPC)	✓ Sì	⚠ Parziale (Informazioni)	⚠ Parziale (Informazioni)	✓ Sì	✓ Sì	⚠ Parziale (Informazioni)
Amazon VPC Lattice	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Servizi Amazon VPC Lattice	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
AWS WAF	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS WAF Classic	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
AWS WAF Regionale	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì

Servizio	Azioni	Resource-level permissions	Resource-based politiche	ABAC	Credenziali temporanee	Service-linked ruoli
AWS Well-Architected Tool	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS Wickr	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Amazon WorkMail	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Flusso di WorkMail messaggi Amazon	✓ Sì	✓ Sì	✗ No	✗ No	✓ Sì	✗ No
Amazon WorkSpaces	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS WorkSpaces Istanze gestite	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
Browser WorkSpaces sicuro Amazon	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✓ Sì
Amazon WorkSpaces Thin Client	✓ Sì	✓ Sì	✗ No	✓ Sì	✓ Sì	✗ No
AWS X-Ray	✓ Sì	⚠ Parziale (Informazioni)	✗ No	⚠ Parziale (Informazioni)	✓ Sì	✗ No

Ulteriori informazioni

AWS CloudTrail

CloudTrail supporta politiche basate sulle risorse su archivi di dati di eventi, dashboard e canali di CloudTrail Lake utilizzati per le integrazioni con fonti di eventi esterne a. AWS

Amazon CloudWatch

CloudWatch i ruoli collegati ai servizi non possono essere creati utilizzando la Console di gestione AWS funzionalità Alarm Actions e supportano solo la funzionalità Alarm Actions.

AWS CodeBuild

CodeBuild supporta la condivisione di risorse tra account utilizzando. AWS RAM

CodeBuild supporta ABAC per le azioni basate su progetti.

AWS Config

AWS Config supporta le autorizzazioni a livello di risorsa per l'aggregazione e le regole di dati multiaccount e più regioni. AWS Config Per un elenco delle risorse supportate, consulta la sezione Aggregazione Multi-Account Multi-Region dei dati e la sezione Regole della Guida API.AWS ConfigAWS Config

AWS Database Migration Service

È possibile creare e modificare le policy allegate alle chiavi di AWS KMS crittografia create per crittografare i dati migrati verso gli endpoint di destinazione supportati. Gli endpoint di destinazione supportati includono Amazon Redshift e Amazon S3. Per ulteriori informazioni, consulta Creazione e utilizzo di AWS KMS chiavi per crittografare i dati di destinazione di Amazon Redshift e AWS KMS Creazione di chiavi per crittografare oggetti di destinazione Amazon S3 nella Guida per l'utente.AWS Database Migration Service

Amazon Elastic Compute Cloud

I ruoli collegati ai servizi EC2 possono essere utilizzati solo per le seguenti funzionalità: Richieste di istanze Spot, Richieste di serie di istanze spot, Parchi istanze Amazon EC2 e Avvio rapido di istanze Windows.

Amazon Elastic Container Service

Solo alcune operazioni Amazon ECS supportano le autorizzazioni a livello di risorse.

AWS Elemental MediaPackage

MediaPackage supporta ruoli collegati ai servizi per la pubblicazione dei log di accesso dei clienti ma non per CloudWatch altre azioni API.

AWS Identity and Access Management

IAM supporta solo un tipo di policy basata su risorse detta policy di attendibilità del ruolo, collegata a un ruolo IAM. Per ulteriori informazioni, consulta Concedere le autorizzazioni agli utenti per cambiare ruoli.

IAM supporta il controllo degli accessi basato su tag per la maggior parte delle risorse IAM. Per ulteriori informazioni, consulta Tag per AWS Identity and Access Management risorse.

Solo alcune delle operazioni API per IAM possono essere chiamate con credenziali temporanee. Per ulteriori informazioni, consulta la sezione di confronto delle opzioni API.

AWS IoT

I dispositivi connessi AWS IoT vengono autenticati utilizzando X.509 certificati o Amazon Cognito Identities. Puoi allegare AWS IoT policy a un X.509 certificato o Amazon Cognito Identity per controllare a cosa il dispositivo è autorizzato a fare. Per ulteriori informazioni, consulta Sicurezza e identità per AWS IoT nella Guida per gli sviluppatori di AWS IoT .

AWS Lambda

Lambda supporta il controllo degli accessi basato su attributi (ABAC) per funzioni, strumenti di mappatura dell'origine degli eventi e configurazioni di firma del codice. I livelli non sono supportati. Per ulteriori informazioni, consulta Utilizzo del controllo degli accessi basato su attributi in Lambda.

Lambda non dispone di ruoli collegati al servizio, a differenza di Lambda@Edge. Per ulteriori informazioni, consulta Service-Linked Roles for Lambda @Edge nella Amazon CloudFront Developer Guide.

Amazon Lightsail

Lightsail supporta parzialmente le autorizzazioni a livello di risorsa e ABAC. Per ulteriori informazioni, consulta la sezione Operazioni, risorse e chiavi della condizione di Amazon Lightsail.

Amazon Managed Streaming per Apache Kafka (MSK)

Puoi collegare una policy del cluster a un cluster Amazon MSK configurato per la connettività multi-VPC.

Multi-party approvazione

Per le Multi-party fonti di identità di approvazione e i team di approvazione, puoi utilizzare tutte e tre le chiavi di condizione ABAC: `aws:ResourceTag/`*`key-name`*`aws:RequestTag/`*`key-name`*, `eaws:TagKeys`.

Per le sessioni, è possibile utilizzare solo la chiave di condizione `aws:ResourceTag/`*`key-name`* perché:

- Le sessioni non sono risorse direttamente taggabili.
- Le sessioni ereditano i tag dal team di approvazione ad esse associato.

AWS Network Manager

AWS Cloud WAN supporta anche ruoli collegati ai servizi. Per ulteriori informazioni, consulta i ruoli collegati ai servizi AWS Cloud WAN nella Amazon VPC AWS Cloud WAN Guide.

Amazon Redshift Serverless

Redshift Serverless supporta policy basate su risorse per la condivisione di snapshot.

Amazon Relational Database Service

Amazon Aurora è un motore di database relazionale completamente gestito compatibile con MySQL e PostgreSQL. Quando si configurano nuovi server di database mediante Amazon RDS, come motore di database è possibile scegliere Aurora MySQL o Aurora PostgreSQL. Per ulteriori informazioni, consulta la sezione Gestione di identità e accessi per Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Amazon Rekognition

Resource-based le politiche sono supportate solo per la copia dei modelli Amazon Rekognition Custom Labels.

AWS Resource Groups

Gli utenti possono assumere un ruolo con una policy che consente operazioni con Resource Groups.

Amazon SageMaker AI

Service-linked i ruoli sono attualmente disponibili per i lavori di formazione su SageMaker SageMaker AI Studio e AI.

Servizio di token di sicurezza AWS

AWS STS non dispone di «risorse», ma consente di limitare l'accesso in modo simile agli utenti. Per ulteriori informazioni, consulta Rifiutare l'accesso alle credenziali di sicurezza temporanee tramite il nome.

Solo alcune delle operazioni API per AWS STS supportare le chiamate con credenziali temporanee. Per ulteriori informazioni, consulta la sezione di confronto delle opzioni API.

Amazon Simple Email Service

Puoi utilizzare solo le autorizzazioni a livello di risorsa in dichiarazioni di policy che fanno riferimento a operazioni correlate all'invio di e-mail, ad esempio `ses:SendEmail` o `ses:SendRawEmail`. Per le dichiarazioni di policy che fanno riferimento a qualsiasi altra operazione, l'elemento Resource può contenere solo `*`.

Solo l'API Amazon SES supporta le credenziali di sicurezza temporanee. L'interfaccia SMTP Amazon SES non supporta credenziali SMTP derivate da credenziali di sicurezza temporanee.

Amazon Simple Storage Service

Amazon S3 supporta l'autorizzazione basata su tag per bucket di directory, risorse di oggetti, Amazon S3 Storage Lens, gruppi Storage Lens e punti di accesso.

Amazon S3 supporta i ruoli collegati ai servizi per Amazon S3 Storage Lens.

AWS Trusted Advisor

L'accesso all'API Trusted Advisor avviene tramite l' Supporto API ed è controllato dalle politiche Supporto IAM.

Amazon Virtual Private Cloud

Amazon VPC supporta il collegamento di una singola policy di risorse a un endpoint VPC per limitare l'accesso tramite tale endpoint. Per ulteriori informazioni sull'utilizzo di policy basate su risorse per

controllare l'accesso alle risorse da specifici endpoint Amazon VPC, consulta la sezione Controllo dell'accesso ai servizi tramite policy di endpoint nella Guida di AWS PrivateLink .

AWS X-Ray

X-Ray non supporta le autorizzazioni a livello di risorsa per tutte le azioni.

X-Ray supporta il controllo degli accessi basato su tag per i gruppi e le regole di campionamento.

AWS Signature versione 4 per le richieste API

> **Important**
>
> Se utilizzi uno strumento AWS SDK (vedi Sample Code and Libraries) o AWS Command Line Interface (AWS CLI) a cui inviare richieste API AWS, puoi saltare il processo di firma, poiché i client SDK e CLI autenticano le tue richieste utilizzando le chiavi di accesso fornite. A meno che tu non abbia una buona ragione per non farlo, ti consigliamo di utilizzare sempre un SDK o una CLI.
> Nelle regioni che supportano più versioni di firma, la firma manuale delle richieste implica che è necessario specificare quale versione della firma utilizzare. Quando fornisci richieste agli Multi-Region Access Point, gli SDK e la CLI passano automaticamente all'utilizzo della versione 4A di Signature senza configurazione aggiuntiva.

Le informazioni di autenticazione inviate in una richiesta devono includere una firma. AWS Signature Version 4 (Sigv4) è il protocollo di AWS firma per aggiungere informazioni di autenticazione alle richieste AWS API.

Non utilizzare la chiave di accesso segreta per firmare le richieste API. Si utilizza invece il processo di firma SigV4. La firma delle richieste implica:

1. Creazione di una richiesta canonica basata sui dettagli della richiesta.
2. Calcolo di una firma utilizzando le tue credenziali. AWS
3. L'aggiunta di questa firma alla richiesta come intestazione di autorizzazione.

AWS quindi replica questo processo e verifica la firma, concedendo o negando l'accesso di conseguenza.

Symmetric SIGv4 richiede di derivare una chiave con ambito per un singolo AWS servizio, in una singola regione, in un determinato giorno. AWS Ciò rende la chiave e la firma calcolata diverse per ogni regione, il che significa che è necessario conoscere la regione a cui è destinata la firma.

Il Signature Version 4 (SigV4A) asimmetrico è un'estensione che supporta la firma con un nuovo algoritmo e la generazione di firme individuali verificabili in più di una regione AWS . Con SigV4a, puoi firmare una richiesta per più regioni, con routing e failover senza interruzioni tra le regioni. Quando si utilizza l' AWS SDK o si richiama una funzionalità che richiede AWS CLI la firma in più regioni, il tipo di firma viene automaticamente modificato per utilizzare Sigv4a. Per informazioni dettagliate, vedi In che modo AWS SigV4a funziona.

In che modo AWS Sigv4 funziona

La procedura riportata di seguito illustra il processo generale di calcolo di una firma con SigV4:

1. La stringa da firmare dipende dal tipo di richiesta. Ad esempio, quando utilizzi l'intestazione dell'autorizzazione HTTP o i parametri della query per l'autenticazione, utilizzi una combinazione di elementi della richiesta per creare la stringa da firmare. Per una richiesta HTTP POST, la policy `POST` nella richiesta è la stringa che firmi.
2. La chiave di firma è una serie di calcoli, con il risultato di ogni passaggio inserito nel successivo. Il passaggio finale è la chiave di firma.
3. Quando un AWS servizio riceve una richiesta autenticata, ricrea la firma utilizzando le informazioni di autenticazione contenute nella richiesta. Se le firme corrispondono, il servizio elabora la richiesta. In caso contrario, la richiesta viene respinta.

Per ulteriori informazioni, consulta Elementi di un AWS Firma della richiesta API.

In che modo AWS SigV4a funziona

SigV4a utilizza firme asimmetriche basate sulla crittografia a chiave pubblica-privata. Sigv4a segue un processo di derivazione delle credenziali con ambito simile a SigV4, tranne per il fatto che Sigv4a utilizza la stessa chiave per firmare tutte le richieste senza dover derivare una chiave di firma distinta in base alla data, al servizio e alla regione. Una coppia di chiavi ECDSA (Elliptic Curve Digital Signature Algorithm) può essere derivata dalla chiave di accesso segreta esistente. AWS

Il sistema utilizza la crittografia asimmetrica per verificare le firme multiregionali, quindi AWS deve solo archiviare le chiavi pubbliche. Le chiavi pubbliche non sono segrete e non possono essere

utilizzate per firmare le richieste. Le firme asimmetriche sono necessarie per le richieste API in più regioni, ad esempio con i punti di accesso Amazon S3. Multi-Region

La procedura riportata di seguito illustra il processo generale di calcolo di una firma con SigV4a:

1. La stringa da firmare dipende dal tipo di richiesta. Ad esempio, quando utilizzi l'intestazione dell'autorizzazione HTTP o i parametri della query per l'autenticazione, utilizzi una combinazione di elementi della richiesta per creare la stringa da firmare. Per una richiesta HTTP POST, la policy `POST` nella richiesta è la stringa che firmi.
2. La chiave di firma deriva da una chiave di accesso segreta AWS tramite una serie di calcoli, con il risultato di ogni passaggio inserito nel successivo. Il passaggio finale produce la coppia di chiavi.
3. Quando un AWS servizio riceve una richiesta firmata con Sigv4a, AWS verifica la firma utilizzando solo la metà pubblica della coppia di chiavi. Se la firma è valida, la richiesta viene autenticata e il servizio elabora la richiesta. Le richieste con firme non valide vengono rifiutate.

Per ulteriori informazioni su Sigv4a per le richieste API multi-regione, consulta il progetto sigv4a-signing-examples su. https://github.com/aws-samples/sigv4a-signing-examples GitHub

Quando firmare le richieste

Quando scrivi codice personalizzato che invia richieste API a, devi includere il codice che firma le AWS richieste. Potrebbe essere necessario scrivere codice personalizzato perché:

- Utilizzi un linguaggio di programmazione per il quale non esiste un SDK AWS .
- È necessario il controllo completo sulla modalità di invio delle richieste AWS.

Mentre le richieste API autenticano l'accesso con AWS Sigv4, AWS gli SDK AWS CLI autenticano le tue richieste utilizzando le chiavi di accesso fornite. Per ulteriori informazioni sull'autenticazione con SDK e altri, consulta. AWS AWS CLIAltre risorse

Perché le richieste vengono firmate

Il processo di firma aiuta a proteggere le richieste, poiché consente di:

- Verificare l'identità del richiedente

  Le richieste autenticate richiedono una firma che hai creato utilizzando le tue chiavi di accesso (ID chiave di accesso, chiave di accesso segreta). Se stai utilizzando credenziali di sicurezza

temporanee, i calcoli della firma richiedono anche un token di sicurezza. Per ulteriori informazioni, consulta AWS accesso programmatico con credenziali di sicurezza.

- Proteggere i dati in transito

  Per evitare che una richiesta venga modificata mentre è in transito, alcuni elementi della richiesta stessa vengono utilizzati per calcolare un hash (digest) e il valore hash risultante è incluso come parte della richiesta. Quando an Servizio AWS riceve la richiesta, utilizza le stesse informazioni per calcolare un hash e lo confronta con il valore hash nella richiesta. Se i valori non corrispondono, AWS rifiuta la richiesta.

- Garantire la protezione da possibili attacchi di tipo replay

  Nella maggior parte dei casi, una richiesta deve pervenire AWS entro cinque minuti dal timestamp indicato nella richiesta. In caso contrario, AWS rifiuta la richiesta.

AWS SigV4 può essere espresso nell'intestazione di autorizzazione HTTP o come stringa di query nell'URL. Per ulteriori informazioni, consulta Metodi di autenticazione.

Altre risorse

- Per ulteriori informazioni sul processo di firma SigV4 per i diversi servizi, consulta Richiesta di esempi di firma.
- Per configurare le credenziali per l'accesso programmatico per la AWS CLI, vedere Credenziali di autenticazione e accesso nella Guida per l'utente dell'interfaccia a riga di comando. AWS
- Gli AWS SDK includono il codice sorgente per la firma delle richieste API GitHub . AWS Per gli esempi di codice, consulta Esempi di progetti in AWS archivio di esempi.
  - AWS SDK per .NET – AWSSigV4Signer.cs
  - AWS SDK per C++ – AWSAuthV4Signer.cpp
  - AWS SDK per Go — sigv4.go https://github.com/aws/smithy-go/blob/a4c9efcda6aa54c75d1a130d1320a2709eebf51d/aws-http-auth/sigv4/sigv4.go
  - AWS SDK per Java – BaseAws4Signer.java
  - AWS SDK per JavaScript — firma-v4
  - AWS SDK per PHP – SignatureV4.php
  - AWS SDK per Python (Boto) — signers.py
  - AWS SDK per Ruby — signer.rb

- Se riscontri errori durante la firma delle richieste, consulta. Risolvi i problemi relativi alla firma della versione 4 di Signature per AWS Richieste API

Elementi di un AWS Firma della richiesta API

> **Important**
>
> A meno che tu non stia utilizzando AWS gli SDK o la CLI, devi scrivere codice per calcolare le firme che forniscono informazioni di autenticazione nelle tue richieste. Il calcolo delle AWS firme nella versione 4 di Signature può essere un'impresa complessa e ti consigliamo di utilizzare gli AWS SDK o la CLI quando possibile.

Ogni HTTP/HTTPS richiesta che utilizza la firma Signature Version 4 deve contenere questi elementi.

Elementi

- Specifica dell'endpoint
- Azione
- Parametri dell'operazione
- Data
- Informazioni sull'autenticazione

Specifica dell'endpoint

Specifica il nome DNS dell'endpoint a cui inviare la richiesta. Questo nome di solito contiene il codice del servizio e la regione. Ad esempio, l'endpoint Amazon DynamoDB per la regione `us-east-1` è `dynamodb.us-east-1.amazonaws.com`.

Per HTTP/1.1 le richieste, è necessario includere l'`Host`intestazione. Per HTTP/2 le richieste, puoi includere l'`:authority`intestazione o l'`Host`intestazione. Utilizza solo l'`:authority`intestazione per la conformità alle specifiche. HTTP/2 Non tutti i servizi supportano HTTP/2 le richieste.

Per gli endpoint supportati da ciascun servizio, consulta Endpoint e quote del servizio nella Riferimenti generali di AWS.

Azione

Specifica un'operazione API per il servizio. Ad esempio, l'operazione `CreateTable` di DynamoDB o l'operazione `DescribeInstances` di Amazon EC2.

Per le operazioni supportate da ciascun servizio, consulta la Guida di riferimento per l'autorizzazione del servizio.

Parametri dell'operazione

Specifica i parametri per l'operazione specificata nella richiesta. Ogni azione AWS API ha una serie di parametri obbligatori e opzionali. La versione dell'API di solito è un parametro obbligatorio.

Per i parametri supportati da un'operazione API, consulta la Documentazione di riferimento delle API per il servizio.

Data

Specifica la data e l'ora della richiesta. Con la data e l'ora nella richiesta puoi evitare che le terze parti intercettino la tua richiesta e la inviino in un secondo momento. La data specificata nell'ambito delle credenziali deve corrispondere alla data della richiesta.

Il timestamp deve essere in UTC e deve avere il seguente formato ISO 8601: YYYYMMDDTHHMMSSZ. Ad esempio, `20220830T123600Z`. Non includere i millisecondi nel time stamp.

Si può utilizzare un'intestazione `date`, un'intestazione `x-amz-date` o includere `x-amz-date` come parametro di query. Se non riusciamo a trovare un'intestazione `x-amz-date`, cerchiamo un'intestazione `date`.

Informazioni sull'autenticazione

Ogni richiesta inviata deve includere le seguenti informazioni. AWS utilizza queste informazioni per garantire la validità e l'autenticità della richiesta.

- Algoritmo: l'algoritmo utilizzato come parte del processo di firma.
  - SigV4: utilizza `AWS4-HMAC-SHA256` per specificare Signature Version 4 con l'algoritmo hash `HMAC-SHA256`.
  - SigV4a: utilizza `AWS4-ECDSA-P256-SHA256` per specificare l'algoritmo hash `ECDSA-P256-SHA-256`.

- Credenziale: una stringa che è formata concatenando i componenti ID chiave di accesso e ambito delle credenziali.
  - SigV4: l'ambito delle credenziali include l'ID della chiave di accesso, la data in formato AAAAMMGG, il codice della Regione, il codice del servizio e la stringa di chiusura `aws4_request`, separati da barre (/). Devi utilizzare caratteri minuscoli per la regione, il codice del servizio e la stringa di chiusura.

    ```
    AKIAIOSFODNN7EXAMPLE/YYYYMMDD/region/service/aws4_request
    ```

  - SigV4a: l'ambito delle credenziali include la data in formato AAAAMMGG, il nome del servizio e la stringa di chiusura `aws4_request`, separati da barre (/). Ricorda che l'ambito delle credenziali non include la Regione, poiché è indicata in un'intestazione separata `X-Amz-Region-Set`.

    ```
    AKIAIOSFODNN7EXAMPLE/YYYYMMDD/service/aws4_request
    ```

- Intestazioni firmate: le intestazioni HTTP da includere nella firma, separate da punto e virgola (;). Ad esempio, `host;x-amz-date`.

  Per SigV4a, è necessario includere un'intestazione dell'insieme di Regioni che specifichi l'insieme delle Regioni in cui la richiesta sarà valida. L'intestazione `X-Amz-Region-Set` è specificata come un elenco di valori separati da virgole. L'esempio seguente mostra un'intestazione di Regione che consente di effettuare una richiesta sia nella Regione us-east-1 sia in us-west-1.

  ```
  X-Amz-Region-Set=us-east-1,us-west-1
  ```

  È possibile utilizzare i caratteri jolly (*) nelle Regioni per specificare più Regioni. Nell'esempio seguente, l'intestazione consente di effettuare una richiesta sia in us-west-1 sia in us-west-2.

  ```
  X-Amz-Region-Set=us-west-*
  ```

- Firma: una stringa con codifica esadecimale che rappresenta la firma calcolata. Devi calcolare la firma utilizzando l'algoritmo specificato nel parametro `Algorithm`.

Per ulteriori informazioni, consulta Metodi di autenticazione

Metodi di autenticazione

> **Important**
>
> A meno che tu non stia utilizzando AWS gli SDK o la CLI, devi scrivere codice per calcolare le firme che forniscono informazioni di autenticazione nelle tue richieste. Il calcolo delle AWS firme nella versione 4 di Signature può essere un'impresa complessa e ti consigliamo di utilizzare gli AWS SDK o la CLI quando possibile.

Puoi esprimere le informazioni di autenticazione utilizzando uno dei seguenti metodi.

Intestazione HTTP di autorizzazione

L'intestazione `Authorization` HTTP è il metodo più comune per autenticare una richiesta. Tutte le operazioni del REST API (ad eccezione dei caricamenti basati su browser che utilizzano richieste `POST`) richiedono questa intestazione.

Gli esempi seguenti mostrano il valore dell'intestazione `Authorization` per SigV4 e SigV4a. Le interruzioni di riga vengono aggiunte a questo esempio solo per migliorare la leggibilità. Nel tuo codice, l'intestazione deve essere una stringa continua. Non c'è una virgola tra l'algoritmo e le credenziali, ma gli altri elementi devono essere separati da virgole.

Example SigV4

```
Authorization: AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,
SignedHeaders=host;range;x-amz-date,
Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024
```

Example SigV4a

```
Authorization: AWS4-ECDSA-P256-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20130524/s3/aws4_request,
SignedHeaders=host;range;x-amz-date;x-amz-region-set,
Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024
```

La tabella seguente descrive i vari componenti del valore dell'intestazione di autorizzazione nell'esempio precedente:

Componente	Description
Autorizzazione	L'algoritmo utilizzato per calcolare la firma. • SigV4: utilizza `AWS4-HMAC-SHA256` . Questa stringa identifica AWS Sigv4 () e l'algoritmo. `AWS4 HMAC-SHA256` • SigV4a: utilizza `AWS4-ECDSA-P256-SHA256` . Questa stringa identifica AWS SigV4 () e l'algoritmo. `AWS4 ECDSA-P256-SHA-256`

Credential	L'ID della chiave di accesso e le informazioni sull'ambito.
	• SigV4: include la data, la Regione e il servizio utilizzati per calcolare la firma. Questa stringa ha il seguente formato:
	`<your-access-key-id>/<date>/ <aws-region>/<aws-service>/ aws4_request`
	• SigV4a: include la data e il servizio utilizzat i per calcolare la firma. Questa stringa ha il seguente formato:
	`<your-access-key-id>/<date>/ <aws-service>/aws4_request`
	Il valore <date> è specificato utilizzando il formato AAAAMMGG. Il valore <aws-service> è S3 quando si invia una richiesta ad Amazon S3.

SignedHeaders	Un elenco separato da punto e virgola delle intestazioni di richiesta che hai utilizzato per il calcolo della firma. L'elenco include solo i

Componente	Description
	nomi delle intestazioni e i nomi delle intestazioni devono essere in minuscolo. Ad esempio: `host;range;x-amz-date` Per SigV4a, è necessario includere un'intestazione dell'insieme di Regioni che specifichi l'insieme delle Regioni in cui la richiesta sarà valida. L'intestazione X-Amz-Region-Set è specificata come un elenco di valori separati da virgole.
Firma	La firma a 256 bit espressa come 64 caratteri esadecimali minuscoli. Ad esempio:`fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024` Tieni presente che i calcoli della firma variano a seconda dell'opzione scelta per trasferire il payload.

Parametri della stringa di query

È possibile utilizzare una stringa di query per esprimere una richiesta interamente in un URL. In questo caso, utilizzi i parametri di interrogazione per fornire le informazioni sulla richiesta, incluse le informazioni di autenticazione. Poiché la firma richiesta è parte dell'URL, questo tipo di URL viene chiamato URL pre-firmato. Puoi utilizzare gli URL preimpostati per incorporare link cliccabili in HTML; questi link saranno validi per un massimo di sette giorni. Per ulteriori informazioni, consulta Authenticating Requests: Using Query Parameters (AWS Signature Version 4) nell'Amazon S3 API Reference.

Gli esempi seguenti mostrano gli URL prefirmati per SigV4 e SigV4a. Le interruzioni di riga vengono aggiunte a questo esempio solo per migliorare la leggibilità.

Example SigV4

```
https://s3.amazonaws.com/amzn-s3-demo-bucket/test.txt ?
```

```
X-Amz-Algorithm=AWS4-HMAC-SHA256 &
X-Amz-Credential=<your-access-key-id>/20130721/<region>/s3/aws4_request &
X-Amz-Date=20130721T201207Z &
X-Amz-Expires=86400 &
X-Amz-SignedHeaders=host &X-Amz-Signature=<signature-value>
```

Example SigV4a

```
https://s3.amazonaws.com/amzn-s3-demo-bucket/test.txt ?
X-Amz-Algorithm=AWS4-ECDSA-P256-SHA256 &
X-Amz-Credential=<your-access-key-id>/20240721/s3/aws4_request &
X-amz-Region-Set=<regionset> &
X-Amz-Date=20240721T201207Z &
X-Amz-Expires=86400 &
X-Amz-SignedHeaders=host;x-amz-region-set &
X-Amz-Signature=<signature-value>
```

> **Note**
>
> Il valore `X-Amz-Credential` nell'URL mostra il carattere "/" solo per leggibilità. In pratica, dovrebbe essere codificato come %2F. Ad esempio: `&X-Amz-Credential=<your-access-key-id>%2F20130721%2Fus-east-1%2Fs3%2Faws4_request`

La tabella seguente descrive i parametri della query nell'URL che forniscono informazioni di autenticazione.

Nome parametro stringa di query	Description
X-Amz-Algorithm	La versione della AWS firma e l'algoritmo utilizzato per calcolare la firma. • SigV4: utilizza `AWS4-HMAC-SHA256` . Questa stringa identifica AWS Sigv4 (`AWS4`) e l'algoritmo. `HMAC-SHA256` • SigV4a: utilizza `AWS4-ECDSA-P256-SHA256` . Questa stringa identifica AWS

Nome parametro stringa di query	Description
	SigV4 () e l'algoritmo. `AWS4 ECDSA-P256-SHA-256`
X-Amz-Credential	Oltre all'ID della chiave di accesso, questo parametro fornisce anche l'ambito per cui la firma è valida. Questo valore deve corrispon dere all'ambito utilizzato nei calcoli della firma, illustrato nella sezione seguente. • SigV4: la forma generale per questo valore di parametro è la seguente: `<your-access-key-id>/<date>/<AWS Region>/<AWS-service>/aws4_request` Ad esempio: `AKIAIOSFODNN7EXAMPLE/20130721/us-east-1/s3/aws4_request` • SigV4a: la forma generale per questo valore di parametro è la seguente: `<your-access-key-id>/<date>/<AWS-service>/aws4_request` Ad esempio: `AKIAIOSFODNN7EXAMPLE/20130721/s3/aws4_request` La Regione per SigV4a è definita nell'inte stazione dell'insieme di Regioni `X-Amz-Region-Set` . Per un elenco delle stringhe AWS regionali , vedere Regional Endpoints nella Guida generale. AWS

Nome parametro stringa di query	Description
X-Amz-Region-Set	L'insieme delle Regioni in cui la richiesta sarà valida. L'intestazione x-amz-region-set è specificata come un elenco di valori separati da virgole.
X-Amz-Date	Il formato di data e ora deve seguire lo standard ISO 8601 e deve essere formattato con il formato `yyyyMMddTHHmmssZ` . Ad esempio, se la data e l'ora erano "08/01/2016 15:32:41.982-700", devono prima essere convertite in UTC (Coordinated Universal Time) e quindi inviate come «20160801T223241Z».
X-Amz-Expires	Fornisce il periodo di tempo, in secondi, per il quale l'URL prefirmato generato è valido. Ad esempio, 86400 (24 ore). Questo valore è un numero intero. Il valore minimo che puoi impostare è 1 e il massimo è 604800 (sette giorni). Un URL prefirmato può essere valido per massimo sette giorni perché la chiave di firma utilizzata nel calcolo della firma è valida per un massimo di sette giorni.

Nome parametro stringa di query	Description
X-Amz-SignedHeaders	Elenca le intestazioni che hai utilizzato per calcolare la firma. Per i calcoli delle firme sono necessarie le seguenti intestazioni: • L'intestazione dell'host HTTP. • Qualsiasi intestazione x-amz-* che intendi aggiungere alla richiesta. • Per Sigv4a, è necessario specificare le regioni in cui è possibile effettuare la richiesta . X-Amz-Region-Set Per maggiore sicurezza, devi firmare tutte le intestazioni della richiesta che intendi includere nella richiesta.
X-Amz-Signature	Fornisce la firma per autenticare la richiesta . Questa firma deve corrispondere alla firma calcolata dal servizio; in caso contrario, il servizio rifiuta la richiesta. Ad esempio, `733255ef022bec3f2a8701cd61d 4b371f3f28c9f193a1f02279211 d48d5193d7` I calcoli delle firme sono descritti nella sezione seguente.
X-Amz-Security-Token	Parametro opzionale delle credenziali se si utilizzano credenziali provenienti dal servizio STS.

Crea un documento firmato AWS Richiesta API

> **Important**
>
> Se utilizzi uno strumento AWS SDK (vedi Sample Code and Libraries AWS CLI) o AWS Command Line Interface () a cui inviare richieste API AWS, puoi saltare questa sezione perché i client SDK e CLI autenticano le tue richieste utilizzando le chiavi di accesso fornite. A meno che tu non abbia una buona ragione per non farlo, ti consigliamo di utilizzare sempre un SDK o una CLI.
> Nelle regioni che supportano più versioni di firma, per le richieste di firma manuale è necessario specificare quale versione della firma viene utilizzata. Quando fornisci richieste agli Multi-Region Access Point, gli SDK e la CLI passano automaticamente all'utilizzo della versione 4A di Signature senza configurazione aggiuntiva.

Puoi utilizzare il protocollo di firma AWS SIGv4 per creare una richiesta firmata per le richieste API. AWS

1. Creazione di una richiesta canonica basata sui dettagli della richiesta.
2. Calcolo di una firma utilizzando le tue credenziali. AWS
3. L'aggiunta di questa firma alla richiesta come intestazione di autorizzazione.

AWS quindi replica questo processo e verifica la firma, concedendo o negando l'accesso di conseguenza.

Per vedere come puoi usare AWS Sigv4 per firmare le richieste API, consulta. Richiesta di esempi di firma

La tabella seguente descrive le funzioni utilizzate nel processo di creazione di una richiesta firmata. Per queste funzioni devi implementare il codice. Per ulteriori informazioni, consulta gli esempi di codice negli SDK AWS.

Funzione	Description
`Lowercase()`	Converte la stringa in minuscolo.
`Hex()`	Codifica in base 16 minuscola.

Funzione	Description
`SHA256Hash()`	Funzione hash crittografica Secure Hash Algorithm (SHA).
`HMAC-SHA256()`	Calcola HMAC utilizzando l'algoritmo SHA256 con la chiave di firma fornita. Questa è la firma definitiva quando firmi con SigV4.
`ECDSA-Sign`	Firma ECDSA (Elliptic Curve Digital Signature Algorithm) calcolata utilizzando firme asimmetri che basate sulla crittografia a chiave pubblica/ privata.
`KDF(K, Label, Context, L)`	Un NIST SP800-108 KDF in modalità Counter che utilizza la funzione PRF HMAC-SHA256 come definita in NIST SP 800-108r1. https://doi.org/10.6028/NIST.SP.800-108r1-upd1
`Oct2Int(byte[ ])`	Una funzione da ottetto a numero intero come descritta in ANSI. X9.62
`Trim()`	Rimuove eventuali spazi bianchi all'inizio o alla fine della stringa.

Funzione	Description
`UriEncode()`	L'URI codifica ogni byte. UriEncode() deve applicare le seguenti regole: • L'URI codifica ogni byte tranne i caratteri senza riserve: 'A'-'Z', 'a'-'z', '0'-'9', '-', ' . ','_' e '~'. • Il carattere di spazio è un carattere riservato e deve essere codificato come "%20" (e non come "+"). • Ogni byte codificato in URI è formato da una '%' e dal valore esadecimale a due cifre del byte. • Le lettere nel valore esadecimale devono essere maiuscole, ad esempio "%1A". • Codifica la barra, '/', ovunque tranne nel nome della chiave dell'oggetto. Ad esempio, se il nome della chiave dell'oggetto è `photos/Jan/sample.jpg` , la barra nel nome della chiave non è codificata. **Important** Le UriEncode funzioni standard fornite dalla piattaforma di sviluppo potrebbero non funzionare a causa delle differenz e di implementazione e della relativa ambiguità nelle RFC sottostanti. Ti consigliamo di scrivere una UriEncode funzione personalizzata per assicurarti che la codifica funzioni.

Funzione	Description
	Per vedere un esempio di UriEncode funzione in Java, vedi Java Utilities sul GitHub sito Web.

> **Note**
>
> Per firmare le richieste, puoi utilizzare AWS SigV4 o SigV4a. AWS La differenza fondamentale tra le due versioni è determinata dalla modalità di calcolo della firma. Con SigV4a, il set di Regioni è incluso nella stringa da firmare, ma non fa parte della fase di derivazione delle credenziali.

Firma delle richieste con credenziali di sicurezza provvisorie

Invece di utilizzare credenziali a lungo termine per firmare una richiesta, puoi utilizzare le credenziali di sicurezza temporanee fornite da (). Servizio di token di sicurezza AWS AWS STS

Quando si utilizzano credenziali di sicurezza temporanee, è necessario aggiungere `X-Amz-Security-Token` all'intestazione di autorizzazione o includerlo nella stringa di query per contenere il token di sessione. Alcuni servizi richiedono l'aggiunta di `X-Amz-Security-Token` alla richiesta canonica. Per gli altri servizi, aggiungi il parametro `X-Amz-Security-Token` alla fine, dopo aver calcolato la firma. Consulta la documentazione di ciascuno Servizio AWS per i requisiti specifici.

Riepilogo delle fasi di firma

Creare una richiesta canonica

Disponi i contenuti della tua richiesta (host, operazione, intestazioni, ecc.) in un formato standard (canonico). La richiesta canonica è uno degli input utilizzati per creare la stringa da firmare. Per i dettagli sulla creazione della richiesta canonica, consulta Elementi di un AWS Firma della richiesta API

Creare un hash della richiesta canonica

Crea un hash della richiesta canonica con lo stesso algoritmo utilizzato per creare l'hash del payload. L'hash della richiesta canonica è una stringa di caratteri esadecimali minuscoli.

Creare una stringa da firmare

Crea una stringa da firmare con la richiesta canonica e informazioni aggiuntive, ad esempio l'algoritmo, la data della richiesta, l'ambito delle credenziali e l'hash della richiesta canonica.

Derivare una chiave di firma

Usa la chiave di accesso segreta per derivare la chiave utilizzata per firmare la richiesta.

Calcolare la firma

Esegui un'operazione di hash con chiave sulla stringa da firmare utilizzando la chiave di firma derivata come chiave hash.

Aggiungere la firma alla richiesta

Aggiungi la firma calcolata a un'intestazione HTTP o alla stringa di query della richiesta.

Creare una richiesta canonica

Per creare una richiesta canonica concatena le seguenti stringhe, separate da caratteri di nuova riga. Ciò consente di verificare che la firma calcolata possa corrispondere alla firma AWS calcolata.

```
<HTTPMethod>\n
<CanonicalURI>\n
<CanonicalQueryString>\n
<CanonicalHeaders>\n
<SignedHeaders>\n
<HashedPayload>
```

- *HTTPMethod*— Il metodo HTTP, ad esempio`GET`, `PUT`, `HEAD`, e`DELETE`.
- *CanonicalUri*— La URI-encoded versione dell'URI del componente del percorso assoluto, a partire da `/` quello che segue il nome di dominio e fino alla fine della stringa o fino al carattere del punto interrogativo (`?`) se si dispone di parametri della stringa di query. Se il percorso assoluto è vuoto, usa una barra (`/`). L'URI nell'esempio seguente, `/amzn-s3-demo-bucket/myphoto.jpg`, è il percorso assoluto e non devi codificare la `/` nel percorso assoluto:

  ```
  http://s3.amazonaws.com/amzn-s3-demo-bucket/myphoto.jpg
  ```

- *CanonicalQueryString*— I parametri della stringa di URI-encoded interrogazione. URI-encode Ciascuno nomina e valore singolarmente. È inoltre necessario ordinare i parametri nella stringa di

query canonica in ordine alfabetico in base al nome della chiave. L'ordinamento avviene dopo la codifica. La stringa di query nell'esempio di URI seguente è:

```
http://s3.amazonaws.com/amzn-s3-demo-bucket?prefix=somePrefix&marker=someMarker&max-keys=2
```

La stringa di query canonica è la seguente (le interruzioni di riga vengono aggiunte a questo esempio a fini di leggibilità):

```
UriEncode("marker")+"="+UriEncode("someMarker")+"&"+
UriEncode("max-keys")+"="+UriEncode("20") + "&" +
UriEncode("prefix")+"="+UriEncode("somePrefix")
```

Quando una richiesta ha come target una sottorisorsa, il valore del parametro di query corrispondente sarà una stringa vuota (""). Ad esempio, il seguente URI identifica la sottorisorsa ACL sul bucket `amzn-s3-demo-bucket`:

```
http://s3.amazonaws.com/amzn-s3-demo-bucket?acl
```

In questo caso, CanonicalQueryString sarebbero:

```
UriEncode("acl") + "=" + ""
```

Se l'URI non include un ?, la richiesta non contiene una stringa di query e occorre impostare la stringa di query canonica su una stringa vuota (""). Dovrai comunque includere il carattere di nuova riga ("\n").

- *CanonicalHeaders*— Un elenco di intestazioni delle richieste con i relativi valori. Le singole coppie di nome e valore dell'intestazione sono separate dal carattere di nuova riga ("\n"). Di seguito è riportato un esempio di CanonicalHeader:

```
Lowercase(<HeaderName1>)+":"+Trim(<value>)+"\n"
Lowercase(<HeaderName2>)+":"+Trim(<value>)+"\n"
...
Lowercase(<HeaderNameN>)+":"+Trim(<value>)+"\n"
```

CanonicalHeaders l'elenco deve includere quanto segue:

- Intestazione `host` HTTP.
- Se l'`Content-Type`intestazione è presente nella richiesta, è necessario aggiungerla all'*`CanonicalHeaders`*elenco.
- Devi aggiungere anche qualsiasi intestazione `x-amz-*` che desideri includere nella richiesta. Ad esempio, se utilizzi credenziali di sicurezza temporanee, nella tua richiesta devi includere `x-amz-security-token`. È necessario aggiungere questa intestazione nell'elenco di. *`CanonicalHeaders`*
- Per SigV4a, è necessario includere un'intestazione dell'insieme di Regioni che specifichi l'insieme delle Regioni in cui la richiesta sarà valida. L'intestazione `X-Amz-Region-Set` è specificata come un elenco di valori separati da virgole. L'esempio seguente mostra un'intestazione di Regione che consente di effettuare una richiesta sia nella Regione us-east-1 sia in us-west-1.

  ```
  X-Amz-Region-Set=us-east-1,us-west-1
  ```

  È possibile utilizzare i caratteri jolly (*) nelle Regioni per specificare più Regioni. Nell'esempio seguente, l'intestazione consente di effettuare una richiesta sia in us-west-1 sia in us-west-2.

  ```
  X-Amz-Region-Set=us-west-*
  ```

> **Note**
>
> L'intestazione `x-amz-content-sha256` è necessaria per le richieste AWS Amazon S3. Fornisce un hash del payload di richiesta. Se non è presente alcun payload, devi indicare l'hash di una stringa vuota.

Il nome di ogni intestazione deve:

- usare caratteri minuscoli.
- in ordine alfabetico.
- seguiti da due punti (`:`).

Per i valori, devi:

- eliminare eventuali spazi all'inizio o alla fine.
- convertire gli spazi sequenziali in uno spazio singolo.
- separare i valori per un'intestazione multivalore con virgole.

- È necessario includere l'intestazione host (HTTP/1.1) o l'intestazione :authority (HTTP/2) e qualsiasi `x-amz-*` intestazione nella firma. Facoltativamente puoi includere altre intestazioni standard nella firma, ad esempio content-type.

Le funzioni `Lowercase()` e `Trim()` utilizzate in questo esempio sono descritte nella sezione precedente.

Di seguito è riportata una stringa `CanonicalHeaders` di esempio. I nomi di intestazione sono in caratteri minuscoli e in ordine alfabetico.

```
host:s3.amazonaws.com
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20130708T220855Z
```

> **Note**
>
> Ai fini del calcolo di una firma di autorizzazione, sono richiesti solo l'host e qualsiasi altra intestazione `x-amz-*`; tuttavia, per evitare la manomissione dei dati, nel calcolo della firma è consigliabile includere intestazioni aggiuntive.
> Non includere le intestazioni hop-by-hop, che vengono spesso alterate durante il transito su un sistema complesso. Questo comprende tutte le intestazioni di trasporto volatili che vengono modificate da proxy, bilanciatori del carico e nodi in un sistema distribuito, tra cui `connection`, `x-amzn-trace-id`, `user-agent`, `keep-alive`, `transfer-encoding`, `TE`, `trailer`, `upgrade`, `proxy-authorization` e `proxy-authenticate`.

- *`SignedHeaders`*— Un elenco in ordine alfabetico e separato da punto e virgola di nomi minuscoli delle intestazioni delle richieste. Le intestazioni della richiesta nell'elenco sono le stesse che hai incluso nella stringa `CanonicalHeaders`. Per l'esempio precedente, il valore di sarebbe il seguente: *`SignedHeaders`*

```
host;x-amz-content-sha256;x-amz-date
```

- *`HashedPayload`*— Una stringa creata utilizzando il payload nel corpo della richiesta HTTP come input per una funzione hash. Questa stringa utilizza caratteri esadecimali minuscoli.

```
Hex(SHA256Hash(<payload>))
```

Se non è presente alcun payload nella richiesta, calcola un hash della stringa vuota, ad esempio quando recupera un oggetto utilizzando una richiesta `GET`, non è presente nulla nel payload.

```
Hex(SHA256Hash(""))
```

> **Note**
>
> Per Amazon S3, includi la stringa letterale `UNSIGNED-PAYLOAD` durante la creazione di una richiesta canonica e imposta lo stesso valore dell'intestazione `x-amz-content-sha256` quando invii la richiesta.
> `Hex(SHA256Hash("UNSIGNED-PAYLOAD"))`

Creare un hash della richiesta canonica

Crea un hash (digest) della richiesta canonica con lo stesso algoritmo utilizzato per creare l'hash del payload. L'hash della richiesta canonica è una stringa di caratteri esadecimali minuscoli.

Creare una stringa da firmare

Per creare una stringa, concatenare le seguenti stringhe, separate da caratteri di nuova riga. Non terminare questa stringa con un carattere di nuova riga.

```
Algorithm \n
RequestDateTime \n
CredentialScope  \n
HashedCanonicalRequest
```

- *`Algorithm`*— L'algoritmo utilizzato per creare l'hash della richiesta canonica.
  - SigV4: utilizza `AWS4-HMAC-SHA256` per specificare l'algoritmo hash `HMAC-SHA256`.
  - SigV4a: utilizza `AWS4-ECDSA-P256-SHA256` per specificare l'algoritmo hash `ECDSA-P256-SHA-256`.
- *`RequestDateTime`*— La data e l'ora utilizzate nell'ambito delle credenziali. Questo valore è l'ora UTC corrente in formato ISO 8601 (ad esempio,`20130524T000000Z`).
- *`CredentialScope`*— L'ambito delle credenziali, che limita la firma risultante alla regione e al servizio specificati.

- SigV4: le credenziali includono una stringa composta dall'ID della chiave di accesso, dalla data in formato YYYYMMDD, dal codice della Regione, dal codice del servizio e dalla stringa di chiusura `aws4_request`, separati da barre (/). Devi utilizzare caratteri minuscoli per la regione, il codice del servizio e la stringa di chiusura. La stringa ha il seguente formato: `YYYYMMDD/region/service/aws4_request`.
- SigV4a: le credenziali includono la data in formato YYYYMMDD, il nome del servizio e la stringa di chiusura `aws4_request`, separati da barre (/). Ricorda che l'ambito delle credenziali non include la Regione, poiché è indicata in un'intestazione separata `X-Amz-Region-Set`. La stringa ha il seguente formato: `YYYYMMDD/service/aws4_request`.
- *`HashedCanonicalRequest`*— L'hash della richiesta canonica, calcolato nel passaggio precedente.

Di seguito è riportata una stringa di esempio da firmare.

```
"<Algorithm>" + "\n" +
timeStampISO8601Format + "\n" +
<Scope> + "\n" +
Hex(<Algorithm>(<CanonicalRequest>))
```

Derivare una chiave di firma

Per derivare una chiave di firma, scegli uno dei seguenti processi per calcolare una chiave di firma per SigV4 o SigV4a.

Derivazione di una chiave di firma per SigV4

Per ricavare una chiave di firma per SIGv4, esegui una successione di operazioni hash con chiave (HMAC) alla data, alla regione e al servizio della richiesta, con la tua chiave di accesso AWS segreta come chiave per l'operazione di hashing iniziale.

Per ogni passaggio, richiama la funzione hash con la chiave e i dati richiesti. Il risultato di ogni chiamata alla funzione hash diventa l'input per la chiamata successiva alla funzione.

L'esempio seguente mostra come derivare l'elemento `SigningKey` utilizzato nella sezione successiva di questa procedura, mostrando l'ordine in cui l'input viene concatenato e sottoposto ad hash. `HMAC-SHA256` è la funzione hash utilizzata per eseguire l'hash dei dati come mostrato.

```
DateKey = HMAC-SHA256("AWS4"+"<SecretAccessKey>", "<YYYYMMDD>")
```

```
DateRegionKey = HMAC-SHA256(<DateKey>, "<aws-region>")
DateRegionServiceKey = HMAC-SHA256(<DateRegionKey>, "<aws-service>")
SigningKey = HMAC-SHA256(<DateRegionServiceKey>, "aws4_request")
```

Input richiesto

- `Key`: una stringa che contiene la tua chiave di accesso segreta.
- `Date`: una stringa che contiene la data utilizzata nell'ambito delle credenziali, nel formato AAAAMMGG.
- `Region`: una stringa che contiene il codice della Regione (ad esempio, `us-east-1`).

  Per un elenco delle stringhe delle regioni, consulta la pagina Endpoint regionali in Riferimenti generali di AWS.
- `Service`: una stringa che contiene il codice del servizio (ad esempio, `ec2`).
- La stringa da firmare creata nel passaggio precedente.

Per derivare una chiave di firma per SigV4

1. Concatena `"AWS4"` e la chiave di accesso segreta. Chiama la funzione hash con la stringa concatenata come stringa di chiave e data come dati.

   ```
   DateKey = hash("AWS4" + Key, Date)
   ```
2. Chiama la funzione hash con il risultato della chiamata precedente come stringa di chiave e regione come dati.

   ```
   DateRegionKey = hash(kDate, Region)
   ```
3. Chiama la funzione hash con il risultato della chiamata precedente come stringa di chiave e servizio come dati.

   Il codice del servizio è definito dal servizio. Puoi utilizzare get-products nella CLI di AWS Pricing per restituire il codice di servizio per un servizio.

   ```
   DateRegionServiceKey = hash(kRegion, Service)
   ```
4. Chiama la funzione hash con il risultato della chiamata precedente come chiave e "aws4_request" come dati.

```
SigningKey = hash(kService, "aws4_request")
```

Derivazione di una chiave di firma per SigV4a

Per creare una chiave di firma per SigV4Aa, utilizza il seguente processo per derivare una coppia di chiavi dalla chiave di accesso segreta. Per un esempio di implementazione di questa derivazione, vedete l'implementazione dell'autenticazione lato client della libreria C99 AWS

```
n = [NIST P-256 elliptic curve group order]
G = [NIST P-256 elliptic curve base point]
label = "AWS4-ECDSA-P256-SHA256"

akid = [AWS access key ID as a UTF8 string]
sk = [AWS secret access Key as a UTF8 Base64 string]

input_key = "AWS4A" || sk
count = 1
while (counter != 255) {
  context = akid || counter // note: counter is one byte
  key = KDF(input_key, label, context, 256)
  c = Oct2Int(key)
  if (c > n - 2) {
    counter++
  } else {
    k = c + 1   // private key
    Q = k * G   // public key
  }
}

if (c < 255) {
  return [k, Q]
} else {
  return FAILURE
}
```

Calcolare la firma

Dopo aver derivato la chiave di firma, calcola la firma da aggiungere alla richiesta. Questa procedura varia in base alla versione della firma utilizzata.

Per calcolare una firma per SigV4

1. Chiama la funzione hash con il risultato della chiamata precedente come chiave e stringa da firmare come dati. Usa la chiave di firma derivata come chiave hash per questa operazione. Il risultato è la firma come valore binario.

```
signature = hash(SigningKey, string-to-sign)
```

2. Converti la firma da rappresentazione binaria a esadecimale, in caratteri minuscoli.

Per calcolare una firma per SigV4a

1. Utilizzando l'algoritmo di firma digitale (ECDSA P-256), firmate la stringa da firmare creata nel passaggio precedente. La chiave utilizzata per questa firma è la chiave asimmetrica privata derivata dalla chiave di accesso segreta come descritto sopra.

```
signature = base16(ECDSA-Sign(k, string-to-sign))
```

2. Converti la firma da rappresentazione binaria a esadecimale, in caratteri minuscoli.

Aggiungere la firma alla richiesta

Aggiungi la firma calcolata alla tua richiesta.

Example Esempio: intestazione di autorizzazione

SigV4

L'esempio seguente mostra un'`Authorization`intestazione per l'`DescribeInstances`azione che utilizza SigV4. AWS Per motivi di leggibilità, questo esempio è formattato con interruzioni di riga. Nel tuo codice, deve essere una stringa continua. Non vi è alcun virgola tra l'algoritmo e `Credential`. Tuttavia, gli altri elementi devono essere separati da virgole.

```
Authorization: AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20220830/us-east-1/ec2/aws4_request,
SignedHeaders=host;x-amz-date,
Signature=calculated-signature
```

SigV4a

L'esempio seguente mostra un'intestazione di autorizzazione per l'azione che utilizza SigV4a. `CreateBucket` AWS Per motivi di leggibilità, questo esempio è formattato con interruzioni di riga. Nel tuo codice, deve essere una stringa continua. Non vi è alcun virgola tra l'algoritmo e le credenziali. Tuttavia, gli altri elementi devono essere separati da virgole.

```
Authorization: AWS4-ECDSA-P256-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20220830/s3/aws4_request,
SignedHeaders=host;x-amz-date;x-amz-region-set,
Signature=calculated-signature
```

Example Esempio: richiesta con parametri di autenticazione nella stringa di query

SigV4

L'esempio seguente mostra una query per l'`DescribeInstances`azione che utilizza AWS Sigv4 che include le informazioni di autenticazione. Per motivi di leggibilità, questo esempio è formattato con interruzioni di riga e non è codificato con l'URL. Nel codice, la stringa di query deve essere una stringa continua con codifica URL.

```
https://ec2.amazonaws.com/?
Action=DescribeInstances&
Version=2016-11-15&
X-Amz-Algorithm=AWS4-HMAC-SHA256&
X-Amz-Credential=AKIAIOSFODNN7EXAMPLE/20220830/us-east-1/ec2/aws4_request&
X-Amz-Date=20220830T123600Z&
X-Amz-SignedHeaders=host;x-amz-date&
X-Amz-Signature=calculated-signature
```

SigV4a

L'esempio seguente mostra una query per l'operazione `CreateBucket` utilizzando AWS SigV4a che include le informazioni di autenticazione. Per motivi di leggibilità, questo esempio è formattato con interruzioni di riga e non è codificato con l'URL. Nel codice, la stringa di query deve essere una stringa continua con codifica URL.

```
https://ec2.amazonaws.com/?
Action=CreateBucket&
Version=2016-11-15&
X-Amz-Algorithm=AWS4-ECDSA-P256-SHA256&
X-Amz-Credential=AKIAIOSFODNN7EXAMPLE/20220830/s3/aws4_request&
X-Amz-Region-Set=us-west-1&
```

```
X-Amz-Date=20220830T123600Z&
X-Amz-SignedHeaders=host;x-amz-date;x-amz-region-set&
X-Amz-Signature=calculated-signature
```

Richiesta di esempi di firma

I seguenti esempi di richieste di AWS firma mostrano come utilizzare Sigv4 per firmare le richieste inviate senza l' AWS SDK o lo strumento a riga di AWS comando.

Caricamento di Amazon S3 basato su browser tramite HTTP POST

Autenticazione delle richieste: Browser-Based Uploads descrive la firma e le informazioni pertinenti utilizzate da Amazon S3 per calcolare la firma alla ricezione della richiesta.

Esempio: il Browser-Based caricamento tramite HTTP POST (Using AWS Signature Version 4) fornisce ulteriori informazioni con un esempio di policy POST e un modulo che puoi utilizzare per caricare un file. La policy di esempio e le credenziali fittizie mostrano il flusso di lavoro e la firma e l'hash della policy risultanti.

Richieste autenticate VPC Lattice

Esempi di richieste autenticate Signature Version 4 (SigV4) fornisce esempi in Python e Java che mostrano come eseguire la firma delle richieste con e senza intercettori personalizzati.

Utilizzo di Signature Version 4 con Amazon Translate

Live Translations in the Metaverse mostra come creare un'applicazione che produce una soluzione di traduzione quasi in tempo reale. Questa soluzione di traduzione vocale utilizza AWS SIGv4 nella codifica del flusso di eventi per produrre trascrizioni in tempo reale.

Utilizzo di Signature Version 4 con Neptune

Esempio: connessione a Neptune utilizzando Python con firma Signature Version 4 mostra come effettuare richieste firmate a Neptune usando Python. Questo esempio include varianti per l'utilizzo di una chiave di accesso o di credenziali temporanee.

Firma delle richieste HTTP ad Amazon Glacier

Esempio di calcolo della firma per l'API di streaming illustra i dettagli della creazione di una firma per il caricamento dell'archivio (POST), una delle due API di streaming di Amazon Glacier.

Invio di richieste HTTP ad Amazon SWF

Invio di richieste HTTP ad Amazon SWF mostra il contenuto dell'intestazione per una richiesta JSON ad Amazon SWF.

Calcolo della firma per le OpenSearch API di streaming in Amazon Service

La firma di una richiesta di ricerca Amazon OpenSearch Service con AWS SDK per PHP versione 3 include un esempio di come inviare richieste HTTP firmate ad Amazon OpenSearch Service.

Esempi di progetti in AWS archivio di esempi

I seguenti progetti di esempio mostrano come firmare le richieste per effettuare richieste API Rest a AWS servizi con linguaggi comuni come Python, Java Node.js, C#, Go e Rust.

Progetti Signature Version 4a

Il progetto sigv4-signing-examples fornisce esempi di come firmare richieste con SigV4 per effettuare richieste API Rest con linguaggi comuni come Python, Java Servizi AWS , C#, Go e Rust. Node.js

Il progetto sigv4a-signing-examples fornisce esempi per la firma di richieste API multi-regione, ad esempio Access Point in Amazon S3. Multi-Region

Pubblica su AWS IoT Core

Il codice Python da pubblicare AWS IoT Core utilizzando il protocollo HTTPS fornisce indicazioni su come pubblicare messaggi AWS IoT Core utilizzando il protocollo HTTPS e l'autenticazione AWS Sigv4. Ha due implementazioni di riferimento: una in Python e l'altra in. NodeJs

L'applicazione.Net Framework per la pubblicazione AWS IoT Core tramite il protocollo HTTPS fornisce indicazioni su come pubblicare messaggi AWS IoT Core utilizzando il protocollo HTTPS e l'autenticazione AWS SigV4. Questo progetto include anche un'implementazione equivalente a .NET core.

Risolvi i problemi relativi alla firma della versione 4 di Signature per AWS Richieste API

> **Important**
>
> A meno che tu non stia utilizzando AWS gli SDK o la CLI, devi scrivere codice per calcolare le firme che forniscono informazioni di autenticazione nelle tue richieste. Il calcolo delle firme SIGv4 può essere un'impresa complessa e ti consigliamo di utilizzare gli SDK o la CLI AWS quando possibile.

Quando sviluppi codice che crea una richiesta firmata, potresti ricevere HTTP 403 da. `SignatureDoesNotMatch` Servizi AWS Questi errori indicano che il valore della firma nella richiesta HTTP AWS non corrisponde alla firma Servizio AWS calcolata. Gli errori HTTP 401 `Unauthorized` vengono restituiti quando le autorizzazioni non consentono al chiamante di effettuare la richiesta.

Le richieste API potrebbero restituire un errore se:

- La richiesta API non è firmata e utilizza l'autenticazione IAM.
- Le credenziali IAM utilizzate per firmare la richiesta non sono corrette o non dispongono delle autorizzazioni per richiamare l'API.
- La firma della richiesta API firmata non corrisponde alla firma calcolata dal servizio AWS .
- L'intestazione della richiesta API non è corretta.

> **Note**
>
> Aggiorna il protocollo di AWS firma dalla versione Signature 2 (Sigv2) alla versione 4 di AWS Signature (Sigv4) prima di esplorare altre soluzioni agli errori. I servizi come Amazon S3 e le regioni non supportano più la firma SigV2.

Possibili cause

- Errori delle credenziali
- Errori nella richiesta canonica e nella stringa di firma
- Errori nell'ambito delle credenziali

- Errori nella chiave di firma

Errori delle credenziali

Assicurati che la richiesta dell'API sia firmata con SigV4. Se la richiesta API non è firmata, potresti ricevere l'errore: `Missing Authentication Token`. Aggiungi la firma mancante e invia nuovamente la richiesta.

Verifica che le credenziali di autenticazione della chiave di accesso e della chiave segreta siano corrette. Se la chiave di accesso non è corretta, potresti ricevere l'errore: `Unauthorized`. Assicurati che l'entità utilizzata per firmare la richiesta sia autorizzata a effettuare la richiesta. Per informazioni dettagliate, vedi Risoluzione dei problemi relativi ai messaggi di errore di accesso rifiutato.

Errori nella richiesta canonica e nella stringa di firma

Se hai calcolato la richiesta canonica in Creare un hash della richiesta canonica o Creare una stringa da firmare, la fase di verifica della firma eseguita dal servizio ha esito negativo con li seguente messaggio di errore:

```
The request signature we calculated does not match the signature you provided
```

Quando il AWS servizio riceve una richiesta firmata, ricalcola la firma. Se sussistono differenze nei valori, le firme non corrispondono. Confronta la stringa e la richiesta canonica con la tua richiesta firmata con il valore nel messaggio di errore. Modifica il processo di firma se riscontri differenze.

> **Note**
>
> Puoi anche verificare di non aver inviato la richiesta tramite una proxy che modifica le intestazioni o la richiesta.

Example Esempio di richiesta canonica

```
GET                                                    -------- HTTP method
/                                                      -------- Path. For API stage
 endpoint, it should be /{stage-name}/{resource-path}
                                                       -------- Query string key-
value pair. Leave it blank if the request doesn't have a query string.
content-type:application/json                          -------- Header key-value
 pair. One header per line.
```

```
host:0123456789.execute-api.us-east-1.amazonaws.com      -------- Host and x-amz-date
 are required headers for all signed requests.
x-amz-date:20220806T024003Z

content-type;host;x-amz-date                              -------- A list of signed
 headers
d167e99c53f15b0c105101d468ae35a3dc9187839ca081095e340f3649a04501        -------- Hash
 of the payload
```

Per verificare che la chiave segreta corrisponda all'ID della chiave di accesso, puoi testarla con un'implementazione funzionante nota. Ad esempio, utilizza un AWS SDK o la AWS CLI per effettuare una richiesta a. AWS

Intestazione della richiesta API

Quando l'intestazione di autorizzazione è vuota, la chiave o la firma della credenziale è mancante o errata, l'intestazione non inizia con il nome di un algoritmo o le coppie chiave-valore non includono un segno uguale, viene visualizzato uno dei seguenti errori:

- L'intestazione dell'autorizzazione non può essere vuota.
- L'intestazione di autorizzazione richiede il parametro "Credential".
- L'intestazione di autorizzazione richiede il parametro "Signature".
- La firma contiene una coppia chiave=valore non valida (segno di uguale mancante) nell'intestazione di autorizzazione.

Assicurati che l'intestazione di autorizzazione SigV4 aggiunta in Calcolare la firma includa la chiave di credenziali corretta e la data della richiesta utilizzando HTTP Date o l'intestazione `x-amz-date`.

Se hai ricevuto un IncompleteSignatureException errore e la struttura della firma è corretta, puoi verificare che l'intestazione di autorizzazione non sia stata modificata durante il transito verso il, Servizio AWS calcolando un SHA-256 hash e una codifica B64 dell'intestazione di autorizzazione nella richiesta lato client.

1. Ottieni l'intestazione di autorizzazione che hai inviato nella richiesta. L'aspetto dell'intestazione dell'autorizzazione è simile all'esempio seguente:

   ```
   Authorization: AWS4-HMAC-SHA256
   Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,
   SignedHeaders=host;range;x-amz-date,
   ```

```
Signature=example-generated-signature
```

2. Calcola un hash dell'intestazione di autorizzazione. SHA-256

```
hashSHA256(rawAuthorizationHeader) = hashedAuthorizationHeader
```

3. Codifica l'intestazione di autorizzazione con hash nel formato Base64.

```
base64(hashedAuthorizationHeader) = encodedHashedAuthorizationHeader
```

4. Confronta la stringa con hash e codificata che hai appena calcolato con la stringa che hai ricevuto nel messaggio di errore. Il messaggio di errore dovrebbe essere simile al seguente esempio:

```
com.amazon.coral.service#IncompleteSignatureException:
The signature contains an in-valid key=value pair (missing equal-sign)
in Authorization header (hashed with SHA-256 and encoded with Base64):
'9c574f83b4b950926da4a99c2b43418b3db8d97d571b5e18dd0e4f3c3ed1ed2c'.
```

- Se i due hash sono diversi, una parte dell'intestazione di autorizzazione è cambiata durante il transito. Questa modifica potrebbe essere dovuta al fatto che i gestori della rete o del client allegano intestazioni firmate o modificano in qualche modo l'intestazione di autorizzazione.
- Se i due hash corrispondono, l'intestazione di autorizzazione inviata nella richiesta corrisponde a quella ricevuta. AWS Controlla il messaggio di errore che hai ricevuto per determinare se il problema è il risultato di credenziali o firme errate. Questi errori sono descritti nelle altre sezioni di questa pagina.

Errori nell'ambito delle credenziali

L'ambito delle credenziali che hai creato in Creare una stringa da firmare limita una firma a una data, una regione e un servizio specifici. Questa stringa ha il seguente formato:

```
YYYYMMDD/region/service/aws4_request
```

> **Note**
>
> Se si utilizza SigV4a, la regione non è inclusa nell'ambito delle credenziali.

Data

Se l'ambito delle credenziali non specifica la stessa data dell'intestazione di x-amz-date, il passaggio di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Date in Credential scope does not match YYYYMMDD from ISO-8601 version of date from
 HTTP
```

Se la richiesta specifica un orario futuro, la fase di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Signature not yet current: date is still later than date
```

Se la richiesta è scaduta, la fase di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Signature expired: date is now earlier than date
```

Region

Se l'ambito delle credenziali non specifica la stessa regione della richiesta, il passaggio di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Credential should be scoped to a valid Region, not region-code
```

Servizio

Se l'ambito delle credenziali non specifica lo stesso servizio dell'intestazione di host, il passaggio di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Credential should be scoped to correct service: 'service'
```

Stringa di terminazione

Se l'ambito delle credenziali termina con aws4_request, il passaggio di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Credential should be scoped with a valid terminator: 'aws4_request'
```

Errori nella chiave di firma

Gli errori causati da un'errata derivazione della chiave di firma o dall'uso improprio della crittografia sono più difficili da risolvere. Dopo aver verificato che la stringa canonica e la stringa da firmare siano corrette, puoi anche verificare la presenza di uno dei seguenti problemi:

- La chiave di accesso segreta non corrisponde all'ID della chiave di accesso specificato.
- Si è verificato un problema con il codice di derivazione della chiave.

Per verificare che la chiave segreta corrisponda all'ID della chiave di accesso, puoi testarla con un'implementazione funzionante nota. Ad esempio, utilizza un AWS SDK o il AWS CLI per effettuare una richiesta a. AWS Per alcuni esempi, consultare Richiesta di esempi di firma.

Riferimento alla policy JSON IAM

Questa sezione presenta la sintassi, le descrizioni e gli esempi dettagliati di elementi, variabili e logica di valutazione delle policy JSON in IAM. Per ulteriori informazioni generali, consulta Panoramica delle policy JSON.

Questo riferimento include le seguenti sezioni.

- Documentazione di riferimento degli elementi delle policy JSON IAM: ulteriori informazioni sugli elementi che è possibile utilizzare durante la creazione di una policy. Visualizzare ulteriori esempi di policy e ulteriori informazioni su condizioni, tipi di dati supportati e il modo in cui vengono utilizzati in vari servizi.
- Logica di valutazione delle policy— Questa sezione descrive AWS le richieste, come vengono autenticate e come vengono AWS utilizzate le policy per determinare l'accesso alle risorse.
- Sintassi del linguaggio della policy JSON IAM: questa sezione presenta una sintassi formale per il linguaggio utilizzato per creare le policy in IAM.
- AWS politiche gestite per le funzioni lavorative: questa sezione elenca tutte le policy gestite di AWS che mappano direttamente a funzioni lavorative nel settore IT. Utilizzare queste policy per concedere le autorizzazioni necessarie per eseguire le attività che ci si aspetta da qualcuno in una determinata funzione lavorativa. Queste policy consolidano le autorizzazioni per molti servizi in una singola policy.
- AWS chiavi di contesto delle condizioni globali— Questa sezione include un elenco di tutte le chiavi di condizione AWS globali che è possibile utilizzare per limitare le autorizzazioni in una policy IAM.

- IAM e AWS STS chiavi di contesto delle condizioni— Questa sezione include un elenco di tutte le chiavi di AWS STS condizione e IAM che puoi usare per limitare le autorizzazioni in una policy IAM.
- Azioni, risorse e chiavi di condizione per AWS i servizi: questa sezione presenta un elenco di tutte le operazioni AWS API che puoi utilizzare come autorizzazioni in una policy IAM. Include anche le chiavi di condizione specifiche del servizio che possono essere utilizzate per ottimizzare ulteriormente la richiesta.

Documentazione di riferimento degli elementi delle policy JSON IAM

I documenti delle policy JSON sono costituiti da elementi. Gli elementi vengono elencati qui nell'ordine generale in cui vengono utilizzati in una policy. L'ordine degli elementi non ha importanza, ad esempio l'elemento `Resource` può venire prima dell'elemento `Action`. Non devi specificare alcun elemento `Condition` nella policy. Per ulteriori informazioni sulla struttura generale e lo scopo di un documento di policy JSON, consulta la pagina Panoramica delle policy JSON.

Alcuni elementi della policy JSON sono reciprocamente esclusivi. Questo significa che non puoi creare una policy che utilizza entrambi. Ad esempio, non è possibile utilizzare `Action` e `NotAction` nella stessa dichiarazione di policy. Altre coppie che si escludono reciprocamente sono `Principal`/`NotPrincipal` e `Resource`/`NotResource`.

I dettagli di ciò che va a comporre una policy variano per ciascun servizio, a seconda di quali operazioni il servizio rende disponibili, quali tipi di risorse contiene e così via. Quando stai scrivendo delle policy per un servizio specifico, è utile consultare esempi di policy per quel servizio. Per un elenco di tutti i servizi che supportano IAM e per i collegamenti alla documentazione di quei servizi che illustrano IAM e le policy, consulta AWS servizi supportati da IAM.

Quando crei o modifichi una policy JSON, IAM può eseguire la convalida delle policy per facilitare la creazione di una policy efficace. IAM identificherà gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente le policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

Argomenti

- Elementi delle policy JSON IAM: Version
- Elementi delle policy JSON IAM: Id
- Elementi delle policy JSON IAM: Statement

- Elementi delle policy JSON IAM: Sid
- Elementi della policy JSON IAM: Effetto
- AWS Elementi della policy JSON: Principale
- AWS Elementi della policy JSON: NotPrincipal
- Elementi della policy JSON IAM: Azione
- Elementi delle policy JSON IAM: NotAction
- Elementi della policy JSON IAM: Risorsa
- Elementi della policy JSON IAM: NotResource
- Elementi della policy JSON IAM: Condizione
- Elementi delle policy IAM: variabili e tag
- Elementi della policy JSON IAM: tipi di dati supportati

Elementi delle policy JSON IAM: Version

> **Chiarimento**
>
> Questo elemento della policy JSON `Version` è diverso da una versione della policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Diversamente, una versione della policy viene creata quando si apportano modifiche alla policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per informazioni sul supporto per versioni multiple disponibile per le policy gestite, consultare the section called "Controllo delle versioni delle policy IAM".

L'elemento della policy `Version` specifica le regole sintattiche di linguaggio che devono essere utilizzate per elaborare una policy. Per utilizzare tutte le funzionalità disponibili della policy, includi il seguente elemento `Version` all'esterno dell'elemento `Statement` in tutte le policy.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    }
  ]
}
```

IAM supporta i seguenti valori degli elementi `Version`:

- `2012-10-17`. Questa è la versione corrente del linguaggio della policy e deve sempre includere un elemento `Version` ed essere impostato su `2012-10-17`. In caso contrario, non è possibile utilizzare caratteristiche come variabili di policy introdotte con questa versione.
- `2008-10-17`. Questa è una versione precedente del linguaggio della policy. Potresti vedere questa versione su policy esistenti meno recenti. Non utilizzare questa versione per le nuove policy o quando si aggiornano policy esistenti. Le caratteristiche più recenti, come variabili di policy, non funzioneranno con la tua policy. Ad esempio, le variabili tipo `${aws:username}` non saranno riconosciute come variabili e verranno trattate come stringhe letterali nella policy.

Elementi delle policy JSON IAM: Id

L'elemento `Id` specifica un identificatore opzionale per la policy. L'ID viene utilizzato in modo diverso in servizi diversi. L'ID è consentito nelle policy basate su risorse, ma non nelle policy basate sulle identità.

Per i servizi che consentono di impostare un elemento `ID`, consigliamo di utilizzare un UUID (GUID) per il valore o incorporare un UUID come parte dell'ID per garantire l'univocità.

JSON

```
{
  "Version":"2012-10-17",
  "Id": "cd3ad3d9-2776-4ef1-a904-4c229d1642ee",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    }
  ]
```

```
}
```

> **Note**
>
> Alcuni AWS servizi (ad esempio Amazon SQS o Amazon SNS) potrebbero richiedere questo elemento e avere requisiti di unicità per esso. Per informazioni specifiche per servizio sulla scrittura di policy, consultare la documentazione per il servizio in uso.

Elementi delle policy JSON IAM: Statement

L'elemento `Statement` è l'elemento principale per una policy. Questa elemento è obbligatorio. L'elemento `Statement` può contenere una singola istruzione o una matrice di singole istruzioni. Ogni singolo blocco di istruzioni deve essere racchiuso tra parentesi graffe { }. In caso di istruzioni multiple, l'array deve essere racchiuso tra parentesi quadre [ ].

```
"Statement": [{...},{...},{...}]
```

L'esempio seguente mostra una policy che contiene una serie di tre istruzioni all'interno di un singolo elemento `Statement`. La policy consente di accedere alla propria "cartella home" nella console Amazon S3. La policy include la variabile `aws:username`, che viene sostituita durante la valutazione della policy con il nome utente dalla richiesta. Per ulteriori informazioni, consulta Introduzione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
```

```
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Condition": {"StringLike": {"s3:prefix": [
        "",
        "home/",
        "home/${aws:username}/"
      ]}}
    },
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/home/${aws:username}",
        "arn:aws:s3:::amzn-s3-demo-bucket/home/${aws:username}/*"
      ]
    }
  ]
}
```

Elementi delle policy JSON IAM: Sid

Puoi fornire un `Sid` (ID istruzione) come identificativo facoltativo per l'istruzione della policy. Puoi assegnare un valore `Sid` a ogni istruzione in una matrice di istruzioni. È possibile utilizzare il valore `Sid` come descrizione per l'istruzione della policy. In servizi che consentono di specificare un elemento `ID`, ad esempio SQS e SNS, il valore `Sid` è semplicemente un ID secondario dell'ID del documento di policy. In IAM, il valore `Sid` deve essere univoco all'interno di una policy JSON.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ExampleStatementID",
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    }
  ]
}
```

L'elemento `Sid` supporta lettere maiuscole ASCII (A-Z), lettere minuscole (a-z) e numeri (0-9).

IAM non utilizza il `Sid` nell'API IAM. Non puoi recuperare una determinata istruzione in base a questo ID.

> **Note**
>
> Alcuni AWS servizi (ad esempio Amazon SQS o Amazon SNS) potrebbero richiedere questo elemento e avere requisiti di unicità per esso. Per informazioni specifiche del servizio sulla scrittura di policy, consulta la documentazione per il servizio in uso.

Elementi della policy JSON IAM: Effetto

L'elemento `Effect` è obbligatorio e specifica se l'istruzione determina un consenso o un rifiuto esplicito. I valori validi di `Effect` sono `Allow` e `Deny`. Il valore `Effect` prevede la distinzione tra lettere maiuscole e minuscole.

```
"Effect":"Allow"
```

Come impostazione predefinita, l'accesso alle risorse è negato. Per consentire l'accesso a una risorsa, è necessario impostare l'elemento `Effect` su `Allow`. Per ignorare un consenso (ad esempio, per ignorare un consenso altrimenti valido), è necessario impostare l'elemento `Effect` su `Deny`. Per ulteriori informazioni, consulta Logica di valutazione delle policy.

AWS Elementi della policy JSON: Principale

Utilizzare l'elemento `Principal` in una policy JSON basata sulle risorse per specificare il principale a cui è consentito o negato l'accesso a una risorsa.

Nelle policy basate sulle risorse devi utilizzare l'elemento `Principal`. Diversi servizi supportano le policy basate sulle risorse, tra cui IAM. Il tipo di policy basata sulle risorse IAM è una policy di attendibilità del ruolo. Nei ruoli IAM, utilizza l'elemento `Principal` nella policy di attendibilità del ruolo per specificare chi può assumere il ruolo. Per l'accesso tra account, è necessario specificare l'identificatore a 12 cifre dell'account affidabile. Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

> **Note**
>
> Dopo aver creato il ruolo, è possibile modificare l'account in "*" per consentire a tutti di assumere il ruolo. In questo caso, è consigliabile limitare gli utenti che possono accedere al ruolo attraverso altri mezzi, ad esempio un elemento `Condition` che limita l'accesso solo a determinati indirizzi IP. Non permettere che il ruolo sia accessibile a tutti.

Altri esempi di risorse che supportano le policy basate sulle risorse includono un bucket Amazon S3 o AWS KMS key.

Non è possibile utilizzare l'`Principal`elemento in una politica basata sull'identità. Identity-based le policy sono policy di autorizzazioni che alleghi alle identità IAM (utenti, gruppi o ruoli). In questi casi, il principale è implicito nell'identità dove è collegata la policy.

Argomenti

- Come specificare un principale
- Account AWS principals
- Principali ruolo IAM
- Principali della sessione come ruolo
- Principali federati OIDC
- Principali federati SAML
- Principali dell'utente IAM
- Principi fondamentali di Centro identità IAM
- AWS STS principi utente federati
- AWS responsabili del servizio
- AWS responsabili del servizio nelle regioni con attivazione obbligatoria
- Tutti i principali
- Ulteriori informazioni

Come specificare un principale

È possibile specificare un principale nell'elemento `Principal` di una policy basata sulle risorse o in chiavi di condizione che supportano i principali.

In una policy è possibile specificare una delle seguenti entità:

- Account AWS e utente root
- Ruoli IAM
- Sessioni come ruolo
- Utenti IAM
- Principali utente federato
- AWS servizi
- Tutti i principali

Non è possibile identificare un gruppo di utenti come principale in una policy (ad esempio una policy basata sulle risorse) perché i gruppi si riferiscono alle autorizzazioni, non all'autenticazione, e i principali sono entità IAM autenticate.

È possibile specificare più di un principale per ciascuno dei tipi di entità nelle sezioni seguenti utilizzando un array. Gli array possono richiedere uno o più valori. Quando si specifica più di un principale in un elemento, si concedono le autorizzazioni a ciascun principale. Questo è un `OR` logico e non un `AND` logico, perché si viene autenticati come un principale alla volta. Se includi più di un valore, utilizza parentesi quadre (`[` e `]`) e delimita con le virgole ogni voce per l'array. La seguente policy di esempio definisce le autorizzazioni per l'account 123456789012 o per l'account 555555555555.

```
"Principal" : {
"AWS": [
  "123456789012",
  "555555555555"
  ]
}
```

> **Note**
>
> Non è possibile utilizzare un carattere jolly per associare parte di un nome di un principale o di un ARN.

Account AWS principals

È possibile specificare Account AWS gli identificatori nell'`Principal`elemento di una policy basata sulle risorse o nelle chiavi di condizione che supportano i principali. In questo modo l'autorità viene delegata all'account. Quando consenti l'accesso a un altro account, un amministratore di tale account deve concedere l'accesso a un'identità (utente o ruolo IAM) in tale account. Quando specifichi un Account AWS, puoi utilizzare l'ARN dell'account (arn:aws:iam: ::root*`account-ID`*) o un modulo abbreviato composto dal prefisso seguito dall'ID dell'account. `"AWS":`

Ad esempio, fornendo un account ID di `123456789012`, è possibile utilizzare uno dei seguenti metodi per specificare l'account nell'elemento `Principal`:

```
"Principal": { "AWS": "arn:aws:iam::123456789012:root" }
```

```
"Principal": { "AWS": "123456789012" }
```

L'ARN dell'account e l'ID dell'account abbreviato si comportano allo stesso modo: entrambi delegano le autorizzazioni all'account. L'utilizzo dell'ARN dell'account nell'elemento `Principal` non limita le autorizzazioni solo per l'utente root dell'account.

> **Note**
>
> Quando salvi una policy basata sulle risorse che include l'ID dell'account abbreviato, il servizio potrebbe convertirlo nell'ARN del principale. Ciò non modifica la funzionalità della policy.

Alcuni AWS servizi supportano opzioni aggiuntive per specificare l'entità dell'account. Ad esempio, Amazon S3 ti consente di specificare un ID utente canonico utilizzando il formato seguente:

```
"Principal": { "CanonicalUser":
 "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be" }
```

Puoi anche specificarne più di uno Account AWS(o ID utente canonico) come principale utilizzando un array. Ad esempio, è possibile specificare un principale in una policy del bucket utilizzando tutti e tre i metodi.

```
"Principal": {
  "AWS": [
```

```
    "arn:aws:iam::123456789012:root",
    "999999999999"
  ],
  "CanonicalUser": "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
}
```

Principali ruolo IAM

È possibile specificare gli ARN dei principali del ruolo IAM nell'elemento `Principal` di una policy basata sulle risorse o in chiavi di condizione che supportano i principali. I ruoli IAM sono identità. In IAM, le identità sono risorse a cui è possibile assegnare autorizzazioni. I ruoli si affidano a un'altra identità autenticata per assumere tale ruolo. Ciò include un principale in AWS o un utente di un provider di identità esterno (IdP). Quando un principal o un'identità assume un ruolo, ricevono credenziali di sicurezza temporanee con le autorizzazioni del ruolo assunto. Quando utilizzano tali credenziali di sessione per eseguire operazioni in AWS, diventano il ruolo principale della sessione.

Quando si specifica un principale del ruolo in una policy basata sulle risorse, le autorizzazioni effettive per il principale sono limitate da qualsiasi tipo di policy che limita le autorizzazioni per il ruolo. Ciò include le policy di sessione e i limiti delle autorizzazioni. Per ulteriori informazioni su come vengono valutate le autorizzazioni effettive per una sessione come ruolo, consulta Logica di valutazione delle policy.

Per specificare l'ARN del ruolo nell'elemento `Principal`, utilizza questo formato:

```
"Principal": { "AWS": "arn:aws:iam::AWS-account-ID:role/role-name" }
```

> **Important**
>
> Se l'elemento `Principal` in una policy di attendibilità del ruolo contiene un ARN che punta a un determinato ruolo IAM, allora l'ARN si trasforma nell'ID principale univoco del ruolo quando si salva la policy. Ciò aiuta a mitigare il rischio che qualcuno aumenti i propri privilegi rimuovendo e ricreando il ruolo. Questa ID nella console non è normalmente presente, in quanto IAM usa una trasformazione inversa verso l'ARN del ruolo quando la policy di affidabilità viene visualizzata.
> Tuttavia, se si elimina il ruolo, la relazione viene interrotta. La policy non è più applicabile, anche se si ricrea il ruolo perché il nuovo ruolo ha un nuovo ID principale che non corrisponde all'ID principale archiviato nella policy di affidabilità. Quando ciò accade, l'ID principale viene visualizzato nelle politiche basate sulle risorse perché non AWS può più ricondurlo a un ARN valido.

Il risultato è che se si elimina e si ricrea un ruolo a cui si fa riferimento in un `Principal` elemento di una policy di fiducia, è necessario modificare il ruolo nella policy per sostituire l'ID principale con l'ARN corretto. L'ARN si trasforma nuovamente nel nuovo ID principale del ruolo quando si salva la policy. Per ulteriori informazioni, consulta Understanding AWS Handling of Deleted IAM role in Policies.

In alternativa, è possibile specificare il principale del ruolo come principale in una policy basata sulle risorse oppure creare una policy di ampia autorizzazione che usa la chiave di condizione `aws:PrincipalArn`. Quando si utilizza questa chiave, al principale della sessione come ruolo vengono concesse le autorizzazioni in base all'ARN del ruolo assunto e non all'ARN della sessione risultante. Poiché AWS non converte gli ARN con chiave di condizione in ID, le autorizzazioni concesse al ruolo ARN persistono se si elimina il ruolo e quindi si crea un nuovo ruolo con lo stesso nome. Identity-based i tipi di policy, come i limiti delle autorizzazioni o i criteri di sessione, non limitano le autorizzazioni concesse utilizzando la chiave `aws:PrincipalArn` condizionale con un carattere jolly (*) nell'`Principal`elemento, a meno che le politiche basate sull'identità non contengano un rifiuto esplicito.

Principali della sessione come ruolo

È possibile specificare le sessioni come ruolo nell'elemento `Principal` di una policy basata sulle risorse o in chiavi in condizione che supportano i principali. Quando un principal o un'identità assume un ruolo, ricevono credenziali di sicurezza temporanee con le autorizzazioni del ruolo assunto. Quando utilizzano tali credenziali di sessione per eseguire operazioni in, diventano il ruolo principale della sessione. AWS

Il formato utilizzato per un responsabile della sessione di ruolo dipende dall' AWS STS operazione utilizzata per assumere il ruolo.

Important

AWS consiglia di utilizzare i role principal IAM nelle policy anziché i ruoli principali delle sessioni di ruolo laddove possibile. Utilizza `Condition` istruzioni e chiavi di condizione per limitare ulteriormente l'accesso quando richiesto.

Per specificare il ruolo ARN principale della sessione nell'`Principal`elemento, utilizzate il seguente formato:

```
"Principal": { "AWS": "arn:aws:sts::AWS-account-ID:assumed-role/role-name/role-session-name" }
```

Inoltre, gli amministratori possono progettare un processo per controllare il modo di emissione delle sessioni come ruolo. Ad esempio, possono fornire una soluzione con un clic per gli utenti che creano un nome della sessione prevedibile. Se l'amministratore compie questa operazione, è possibile utilizzare i principali della sessione come ruolo nelle policy o nelle chiavi di condizione. In caso contrario, è possibile specificare l'ARN del ruolo come principale nella chiave di condizione `aws:PrincipalArn`. Il modo in cui si specifica il ruolo come principale può modificare le autorizzazioni effettive per la sessione risultante. Per ulteriori informazioni, consulta Principali ruolo IAM.

Principali federati OIDC

Un principal federato OIDC è il principale utilizzato quando si chiama l' AWS STS `AssumeRoleWithWebIdentity`API con un token web JSON (JWT) da un IDP conforme a OIDC, noto anche come OpenID Provider (OP), per richiedere credenziali temporanee. AWS Un principale federato OIDC può rappresentare un IDP OIDC nel tuo account o i 4 provider di identità integrati:,, e Amazon Cognito. AWS Login with Amazon Google Facebook https://docs.aws.amazon.com/cognito/latest/developerguide/role-based-access-control.html

Gli utenti, i carichi di lavoro o i sistemi a cui è stato rilasciato un JWT dal proprio IDP OIDC possono chiamare `AssumeRoleWithWebIdentity` utilizzando il JWT per richiedere credenziali di AWS sicurezza temporanee per un ruolo IAM configurato per considerare attendibile l'IDP OIDC che ha emesso il JWT. Il JWT può essere un ID token, un token di accesso o un token JWT fornito con qualsiasi altro metodo purché soddisfi i requisiti elencati da AWS STS. Per ulteriori informazioni, consulta Scenari comuni e Richiesta di credenziali tramite un provider OIDC.

Usa questo tipo principale nella tua policy di fiducia del ruolo per consentire o negare le autorizzazioni di chiamata `AssumeRoleWIthWebIdentity` utilizzando un IDP OIDC presente nel tuo o uno dei quattro IDP integrati. Account AWS Per specificare l'ARN del principale federato OIDC nell'elemento `Principal` di una policy di attendibilità dei ruoli, utilizza uno dei quattro formati seguenti per gli IdP OIDC incorporati:

```
"Principal": { "Federated": "cognito-identity.amazonaws.com" }
```

```
"Principal": { "Federated": "www.amazon.com" }
```

```
"Principal": { "Federated": "graph.facebook.com" }
```

```
"Principal": { "Federated": "accounts.google.com" }
```

Quando utilizzi un provider OIDC che aggiungi al tuo account, ad esempio, specifichi l'ARN del provider nella politica di fiducia del tuo ruolo GitHub. Questa configurazione ti consente di scrivere policy IAM che controllano l'accesso in modo specifico per gli utenti autenticati tramite il tuo provider di identità personalizzato.

```
"Principal": { "Federated": "arn:aws:iam::AWS-account-ID:oidc-provider/full-OIDC-
identity-provider-URL" }
```

Ad esempio, se GitHub è il provider di identità web attendibile, l'ARN della sessione del ruolo OIDC nell'elemento `Principal` di una policy di attendibilità dei ruoli utilizza il seguente formato:

```
"Principal": { "Federated": "arn:aws:iam::AWS-account-ID:oidc-provider/
tokens.actions.githubusercontent.com" }
```

Per ulteriori informazioni, consulta Configuring OpenID Connect in Amazon Web Services.

I principali federati OIDC non sono supportati in tipi di policy diversi dalle policy di attendibilità dei ruoli.

Principali federati SAML

Un principal federato SAML è un principal utilizzato quando si chiama l' AWS STS `AssumeRoleWithSAML`API per richiedere AWS credenziali temporanee utilizzando un'asserzione SAML. È possibile utilizzare un gestore dell'identità digitale (IdP) SAML esterno per accedere e quindi assumere un ruolo IAM utilizzando questa operazione. Analogamente a`AssumeRoleWithWebIdentity`, `AssumeRoleWithSAML` non richiede credenziali per l'autenticazione. AWS Invece, gli utenti si autenticano prima con il proprio provider di identità SAML, quindi effettuano la chiamata `AssumeRoleWithSAML` API utilizzando la loro asserzione SAML o vengono reindirizzati alla pagina per accedere al AWS Sign-In/SAML . Console di gestione AWS Per ulteriori informazioni su quali principali possono assumere un ruolo utilizzando questa operazione, consulta Confronta le AWS STS credenziali.

Utilizza questo tipo di principale nella policy di attendibilità del ruolo per consentire o negare l'autorizzazione in base al provider di identità SAML attendibile. Per specificare l'ARN della sessione

del ruolo dell'identità Web nell'elemento `Principal` di una policy di attendibilità dei ruoli, utilizza questo formato:

```
"Principal": { "Federated": "arn:aws:iam::AWS-account-ID:saml-provider/provider-name" }
```

Principali dell'utente IAM

Puoi specificare gli utenti IAM nell'elemento `Principal` di una policy basata sulle risorse o nelle chiavi della condizione che supportano i principali.

> **Note**
>
> In un elemento `Principal`, la parte del nome utente dell'Amazon Resource Name(ARN) fa distinzione tra maiuscole e minuscole.

```
"Principal": { "AWS": "arn:aws:iam::AWS-account-ID:user/user-name" }
```

```
"Principal": {
  "AWS": [
    "arn:aws:iam::AWS-account-ID:user/user-name-1",
    "arn:aws:iam::AWS-account-ID:user/user-name-2"
  ]
}
```

Quando si specificano gli utenti in un elemento `Principal`, non è possibile utilizzare un carattere jolly (`*`) che indica "tutti gli utenti". I principali devono sempre nominare utenti determinati.

> **Important**
>
> Se l'elemento `Principal` in una policy di attendibilità del ruolo contiene un nome ARN che punta a un determinato utente o IAM, allora IAM trasforma l'ARN nell'ID principale univoco dell'utente quando la policy viene salvata. Ciò aiuta a mitigare il rischio che qualcuno aumenti i propri privilegi rimuovendo e ricreando il ruolo o l'utente. Questa ID nella console non è normalmente presente, in quanto c'è anche una trasformazione inversa verso il nome ARN dell'utente quando la policy di affidabilità viene visualizzata.
> Tuttavia, se si elimina l'utente, la relazione viene interrotta. La policy non è più applicabile, anche se viene ricreato l'utente. Questo perché il nuovo utente ha un nuovo ID principale che

non corrisponde all'ID archiviato nella policy di affidabilità. Quando ciò accade, l'ID principale viene visualizzato nelle politiche basate sulle risorse perché non AWS può più ricondurlo a un ARN valido.
Il risultato è che se si elimina e si ricrea un utente o referenziato in un elemento `Principal` della policy di attendibilità, è necessario modificare il ruolo per sostituire l'ID principale non corretto con il nome ARN corretto. IAM trasforma nuovamente l'ARN nel nuovo ID principale dell'utente quando si salva la policy.

Principi fondamentali di Centro identità IAM

In Centro identità IAM, il principio di una policy basata sulle risorse deve essere definito come principale dell' Account AWS . Per specificare l'accesso, fai riferimento all'ARN del ruolo del set di autorizzazioni nel blocco delle condizioni. Per ulteriori dettagli, consulta la sezione Referenziare i set di autorizzazioni nelle policy delle risorse, in Amazon EKS e in AWS KMS nella Guida per l'utente di Centro identità IAM.

AWS STS principi utente federati

È possibile specificare le sessioni come utente federato nell'elemento `Principal` di una policy basata sulle risorse o in chiavi di condizione che supportano i principali.

Important

AWS consiglia di limitare l'uso delle sessioni utente AWS STS federate. Invece, promuove l'uso di ruoli IAM.

Un'entità utente AWS STS federata viene creata tramite l'`GetFederationToken`operazione richiamata con credenziali IAM di lunga durata. Le autorizzazioni dell'utente federato sono l'intersezione tra il principale che ha effettuato la chiamata `GetFederationToken` e le policy della sessione passate come parametri all'API `GetFederationToken`.

In AWS, gli utenti IAM o un utente Utente root dell'account AWS possono autenticarsi utilizzando chiavi di accesso a lungo termine. Per ulteriori informazioni su quali principali possono eseguire la federazione utilizzando questa operazione, consulta Confronta le AWS STS credenziali.

- Utente federato IAM: un utente IAM esegue la federazione utilizzando l'operazione `GetFederationToken`, che si traduce in una sessione come utente federato per quell'utente IAM.

- Utente root federato: un utente root esegue la federazione usando l'operazione `GetFederationToken`, che si traduce in una sessione come utente federato per quell'utente root.

Quando un utente IAM o un utente root richiede credenziali temporanee per AWS STS utilizzare questa operazione, inizia una sessione utente federata temporanea. L'ARN di questa sessione si basa sull'identità originale federata.

Per specificare l'ARN della sessione come utente federato nell'elemento `Principal`, utilizza questo formato:

```
"Principal": { "AWS": "arn:aws:sts::AWS-account-ID:federated-user/user-name" }
```

AWS responsabili del servizio

È possibile specificare AWS i servizi nell'`Principal`elemento di una policy basata sulle risorse o nelle chiavi di condizione che supportano i principali. Un principale del servizio è un identificatore per un servizio.

I ruoli IAM che possono essere assunti da un AWS servizio sono denominati ruoli di servizio. ??? I ruoli di servizio devono includere una policy di affidabilità. Le Policy di affidabilità sono policy basate su risorse collegate a un ruolo che definisce quali principali possono assumere il ruolo. Alcuni ruoli di servizio hanno policy di affidabilità predefinite. Tuttavia, in alcuni casi, è necessario specificare il principale del servizio nella policy di affidabilità. Il principale del servizio in una policy IAM non può essere `"Service": "*"`.

> **Important**
>
> L'identificatore di un principale del servizio include il nome del servizio ed è solitamente nel formato seguente:
> *service-name*`.amazonaws.com`

Il principale del servizio è definito dal servizio. Puoi trovare l'entità del servizio per alcuni servizi aprendoAWS servizi supportati da IAM, controllando se il servizio ha Sì nella colonna del Service-linked ruolo e aprendo il link Sì per visualizzare la documentazione del ruolo collegato al servizio per quel servizio. Trova la sezione Autorizzazioni del Service-Linked ruolo relativa a quel servizio per visualizzare l'entità del servizio.

L'esempio seguente mostra una policy che può essere collegata a un ruolo del servizio. Questa policy consente a due servizi, Amazon ECS e Elastic Load Balancing, di assumere il ruolo. I servizi possono eseguire qualsiasi attività concesse da una policy di autorizzazioni assegnata al ruolo (non visualizzato). Per specificare più principali del servizio, non si specificano due elementi `Service`, è possibile averne solo uno. Utilizzare invece una serie di principali del servizio come il valore di un elemento singolo `Service`.

```
"Principal": {
    "Service": [
        "ecs.amazonaws.com",
        "elasticloadbalancing.amazonaws.com"
   ]
}
```

AWS responsabili del servizio nelle regioni con attivazione obbligatoria

Puoi lanciare risorse in diverse AWS regioni e alcune di queste regioni devi aderire. Per un elenco completo delle regioni a cui devi aderire, consulta Gestione AWS delle regioni nella Riferimenti generali di AWS guida.

Quando un AWS servizio in una regione opt-in effettua una richiesta all'interno della stessa regione, il formato del nome principale del servizio viene identificato come la versione non regionalizzata del nome principale del servizio:

service-name`.amazonaws.com`

Quando un AWS servizio in una regione con opt-in invia una richiesta interregionale a un'altra regione, il formato del nome principale del servizio viene identificato come la versione regionalizzata del nome principale del servizio:

service-name`.`*{region}*`.amazonaws.com`

Ad esempio, si consideri un argomento Amazon SNS situato nella Regione `ap-southeast-1` e un bucket Amazon S3 nella Regione di adesione `ap-east-1`. Supponiamo che si desideri configurare le notifiche bucket S3 per pubblicare messaggi nell'argomento SNS. Per consentire al servizio S3 di pubblicare messaggi sull'argomento SNS, è necessario concedere l'`sns:Publish`autorizzazione principale del servizio S3 tramite la politica di accesso basata sulle risorse dell'argomento.

Se si specifica la versione non regionalizzata del principale del servizio S3 `s3.amazonaws.com`, nella policy di accesso all'argomento, la richiesta `sns:Publish` dal bucket all'argomento avrà esito

negativo. L'esempio seguente specifica il principale del servizio S3 non regionalizzato nell'elemento della policy `Principal` della policy di accesso all'argomento SNS.

```
"Principal": { "Service": "s3.amazonaws.com" }
```

Poiché il bucket si trova in una regione di adesione e la richiesta viene effettuata al di fuori della stessa regione, il principale del servizio S3 appare come nome del principale del servizio regionalizzato, `s3.ap-east-1.amazonaws.com`. È necessario utilizzare il nome principale del servizio regionalizzato quando un AWS servizio in una regione opt-in effettua una richiesta a un'altra regione. Dopo aver specificato il nome del principale del servizio regionalizzato, se il bucket effettua una richiesta `sns:Publish` all'argomento SNS situato in un'altra regione, la richiesta avrà esito positivo. L'esempio seguente specifica il principale del servizio S3 regionalizzato nell'elemento della policy `Principal` della policy di accesso all'argomento SNS.

```
"Principal": { "Service": "s3.ap-east-1.amazonaws.com" }
```

Le policy di risorse o gli elenchi di autorizzazioni basati sui principali dei servizi per le richieste tra regioni da una regione di adesione a un'altra regione avranno esito positivo solo se si specifica il nome del principale del servizio regionalizzato.

> **Note**
>
> Per le policy di attendibilità dei ruoli IAM, consigliamo di utilizzare il nome del principale del servizio non regionalizzato. Le risorse IAM sono globali e quindi lo stesso ruolo può essere utilizzato in qualsiasi regione.

Tutti i principali

Puoi utilizzare un carattere jolly (*) per specificare tutti i principali nell'elemento `Principal` di una policy basata sulle risorse o nelle chiavi della condizione che supportano tali entità. Resource-based politiche concedono le autorizzazioni e le chiavi della condizione vengono utilizzate per limitare le condizioni di un'istruzione della policy.

> **Important**
>
> Ti consigliamo di non utilizzare un carattere jolly (*) nell'elemento `Principal` di una policy basata sulle risorse con un effetto `Allow` a meno che tu non intenda concedere un

accesso pubblico o anonimo. In caso contrario, specifica i principali, i servizi o gli account AWS previsti nell'elemento `Principal`, quindi limita ulteriormente l'accesso nell'elemento `Condition`. Ciò vale in special modo per le policy di attendibilità del ruolo IAM, perché consentono ad altri principali di diventare un principale nel tuo account.

Per le policy basate sulle risorse, l'utilizzo di un carattere jolly (*) con un effetto `Allow` concede l'accesso a tutti gli utenti, compresi gli utenti anonimi (accesso pubblico). Per gli utenti IAM e i principali del ruolo all'interno del tuo account non sono richieste altre autorizzazioni. Per i principali di altri account, devono inoltre disporre di autorizzazioni basate su identità nel proprio account che consentano loro di accedere alla tua risorsa. Questo è chiamato accesso tra account.

Per gli utenti anonimi, i seguenti elementi sono equivalenti:

```
"Principal": "*"
```

```
"Principal" : { "AWS" : "*" }
```

Non è possibile utilizzare un carattere jolly per associare parte di un nome di un principale o di un ARN.

L'esempio seguente mostra una policy basata sulle risorse che può essere utilizzata al posto di AWS Elementi della policy JSON: NotPrincipal per negare esplicitamente tutti i principali, eccetto quelli specificati nell'elemento `Condition`. Questa policy deve essere aggiunta a un bucket Amazon S3.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "UsePrincipalArnInsteadOfNotPrincipalWithDeny",
      "Effect": "Deny",
      "Action": "s3:*",
      "Principal": "*",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-bucket"
      ],
```

```
      "Condition": {
        "ArnNotEquals": {
          "aws:PrincipalArn": "arn:aws:iam::444455556666:user/user-name"
        }
      }
    }
  ]
}
```

Ulteriori informazioni

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Esempi di policy di bucket nella Guida per l'utente di Amazon Simple Storage Service (Amazon S3)
- Policy di esempio per Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service
- Policy di esempio per Amazon SQS nella Guida per gli sviluppatori di Amazon Simple Queue Service
- Policy delle chiavi nella Guida per gli sviluppatori di AWS Key Management Service
- Identificatori di account nella Riferimenti generali di AWS
- Federazione OIDC

AWS Elementi della policy JSON: NotPrincipal

L'elemento `NotPrincipal` utilizza `"Effect":"Deny"` per negare l'accesso a tutti i principali tranne il principale specificato nell'elemento `NotPrincipal`. Un principale può essere un utente IAM, un utente AWS STS federato, un ruolo IAM, una sessione di ruolo presunta Account AWS, un AWS servizio o un altro tipo principale. Per ulteriori informazioni sui principali, consulta AWS Elementi della policy JSON: Principale.

`NotPrincipal` deve essere usato con `"Effect":"Deny"`. L'uso con `"Effect":"Allow"` non è supportato.

> **Important**
>
> Si sconsiglia l'uso di `NotPrincipal` per nuove policy basate sulle risorse come parte della strategia di sicurezza e autorizzazione. Quando utilizzi `NotPrincipal`, la risoluzione

dei problemi legati agli effetti di più tipi di policy può essere difficile. Con gli operatori di condizione ARN, si consiglia invece di utilizzare la chiave di contesto `aws:PrincipalArn`.

Punti chiave

- L'elemento `NotPrincipal` è supportato nelle policy basate sulle risorse per alcuni servizi AWS , tra cui gli endpoint VPC. Le policy basate su risorse sono policy che vengono incorporate direttamente in una risorsa. Non puoi utilizzare l'elemento `NotPrincipal` in una policy basata sull'identità IAM o in una policy di attendibilità del ruolo IAM.
- Non utilizzare istruzioni di policy basate sulle risorse che includono un elemento di policy `NotPrincipal` con effetto `Deny` per gli utenti o i ruoli IAM ai quali è collegata una policy con limite delle autorizzazioni. L'elemento `NotPrincipal` con effetto `Deny` rifiuterà sempre qualsiasi principale IAM al quale è collegata una policy con limite delle autorizzazioni, indipendentemente dai valori specificati nell'elemento `NotPrincipal`. Ciò fa sì che alcuni utenti o ruoli IAM che altrimenti avrebbero accesso alla risorsa perdano l'accesso. Ti consigliamo di modificare le istruzioni di policy basate sulle risorse di modo che, per limitare l'accesso, utilizzino l'operatore di condizione `ArnNotEquals` con la chiave di contesto `aws:PrincipalArn` anziché l'elemento `NotPrincipal`. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta la pagina Limiti delle autorizzazioni per le entità IAM.
- Quando si utilizza `NotPrincipal`, è necessario specificare anche l'ARN dell'account del principale non rifiutato. In caso contrario, la policy potrebbe rifiutare l'accesso all'intero account contenente il principale. A seconda del servizio che si include nella policy, AWS potrebbe convalidare prima l'account e poi l'utente. Se viene valutato un utente con ruolo presunto (qualcuno che utilizza un ruolo), AWS potrebbe convalidare prima l'account, poi il ruolo e poi l'utente assunto. L'utente con ruolo assunto viene identificato tramite il nome della sessione del ruolo specificato quando l'utente ha assunto il ruolo. Pertanto, è fortemente consigliabile includere esplicitamente l'ARN di un account utente oppure includere sia l'ARN di un ruolo sia l'ARN dell'account che contiene quel ruolo.
- L'elemento `NotPrincipal` non è supportato nelle policy di controllo dei servizi (SCP) e nelle policy di controllo delle risorse (RCP).

Alternative all'elemento **`NotPrincipal`**

Quando si gestisce il controllo degli accessi in AWS, può verificarsi la necessità di negare esplicitamente a tutti i responsabili l'accesso a una risorsa, ad eccezione di uno o più principali

specificati dall'utente. AWS consiglia di utilizzare un'istruzione Deny con tasti contestuali delle condizioni globali per un controllo più preciso e una risoluzione dei problemi più semplice. Gli esempi seguenti mostrano approcci alternativi che utilizzano operatori di condizione come `StringNotEquals` o `ArnNotEquals` per negare l'accesso a tutti i principali, ad eccezione di quelli specificati nell'elemento Condizione.

Scenario di esempio che utilizza un ruolo IAM

Puoi utilizzare una policy basata sulle risorse con un'istruzione Nega per impedire a tutti i ruoli IAM, ad eccezione di quelli specificati nell'elemento Condizione, di accedere alle tue risorse o di manipolarle. Questo approccio segue il principio AWS di sicurezza secondo cui un'esplicita deny ha sempre la precedenza su qualsiasi istruzione di autorizzazione e aiuta a mantenere il principio del privilegio minimo nell'infrastruttura. AWS

Invece di usare `NotPrincipal`, consigliamo di utilizzare un'istruzione Nega con chiavi di contesto di condizione globali e l'operatore di condizione come `ArnNotEquals` per consentire esplicitamente a un ruolo IAM l'accesso alle tue risorse. L'esempio seguente impiega era: PrincipalArn per consentire esplicitamente al ruolo `read-only-role` di accedere ai bucket Amazon S3 nella cartella `Bucket_Account_Audit`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyCrossAuditAccess",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::Bucket_Account_Audit",
        "arn:aws:s3:::Bucket_Account_Audit/*"
      ],
      "Condition": {
        "ArnNotEquals": {
          "aws:PrincipalArn": "arn:aws:iam::444455556666:role/read-only-role"
        }
      }
    }
  ]
```

```
}
```

Scenario di esempio che utilizza un principale del servizio

È possibile utilizzare un'istruzione Nega per impedire a tutti i principali del servizio, ad eccezione di quelli specificati nell'elemento `Condition`, di accedere alle risorse o di manipolarle. Questo approccio è particolarmente utile quando è necessario implementare controlli di accesso granulari o stabilire limiti di sicurezza tra diversi servizi e applicazioni nell'ambiente AWS .

Invece di usare `NotPrincipal`, consigliamo di utilizzare un'istruzione Nega con chiavi di contesto di condizione globali e l'operatore di condizione `StringNotEquals` per consentire esplicitamente a un principale del servizio l'accesso alle tue risorse. L'esempio seguente impiega `era: PrincipalServiceName` per consentire esplicitamente al principale del servizio AWS CodeBuild di accedere ai bucket Amazon S3 nella cartella `BUCKETNAME`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyNotCodeBuildAccess",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::BUCKETNAME",
        "arn:aws:s3:::BUCKETNAME/*"
      ],
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:PrincipalServiceName": "codebuild.amazonaws.com"
        }
      }
    }
  ]
}
```

Elementi della policy JSON IAM: Azione

L'elemento `Action` descrive l'operazione o le operazioni specifiche che saranno concesse o negate. Le istruzioni devono includere un elemento `Action` o un elemento `NotAction`. Ogni AWS servizio ha una propria serie di azioni che descrivono le attività che è possibile eseguire con quel servizio. Ad esempio, l'elenco delle azioni per Amazon S3 è disponibile in Specificare le autorizzazioni in una policy nella Amazon Simple Storage Service User Guide, l'elenco delle azioni per Amazon EC2 è disponibile nell'Amazon EC2 API Reference e l'elenco delle azioni per AWS Identity and Access Management è disponibile nella IAM API Reference. https://docs.aws.amazon.com/IAM/latest/APIReference/API_Operations.html Per trovare l'elenco delle operazioni per altri servizi, consulta la documentazione di riferimento alle API per il servizio.

AWS fornisce inoltre informazioni di riferimento sui servizi in formato JSON per semplificare l'automazione dei flussi di lavoro di gestione delle policy. Con le informazioni di riferimento del servizio, puoi accedere alle azioni, alle risorse e alle chiavi di condizione disponibili tramite file leggibili Servizi AWS dalla macchina. Per ulteriori informazioni, consulta Simplified Servizio AWS information for programmatic access nella documentazione di riferimento sull'autorizzazione dei servizi.

È possibile specificare un valore utilizzando un namespace come prefisso dell'operazione (`iam`, `ec2` `sqs`, `sns`, `s3`, ecc.) seguito dal nome dell'operazione da consentire o negare. Il nome deve corrispondere a un'operazione che è supportata dal servizio. Il prefisso e il nome dell'operazione non fanno distinzione tra maiuscole e minuscole. Ad esempio, `iam:ListAccessKeys` è equivalente a `IAM:listaccesskeys`. I seguenti esempi mostrano gli elementi `Action` per diversi servizi.

Operazione di Amazon SQS

```
"Action": "sqs:SendMessage"
```

Operazione Amazon EC2

```
"Action": "ec2:StartInstances"
```

Operazione IAM

```
"Action": "iam:ChangePassword"
```

Operazioni di Amazon S3

```
"Action": "s3:GetObject"
```

Puoi specificare valori multipli per l'elemento `Action`.

```
"Action": [ "sqs:SendMessage", "sqs:ReceiveMessage", "ec2:StartInstances",
 "iam:ChangePassword", "s3:GetObject" ]
```

Puoi utilizzare i caratteri jolly con corrispondenza a più caratteri (`*`) e i caratteri jolly con corrispondenza a carattere singolo (`?`) per dare accesso a tutte le azioni offerte dal prodotto specifico. AWS Ad esempio, il seguente elemento `Action` si applica a tutte le operazioni S3.

```
"Action": "s3:*"
```

Puoi anche utilizzare caratteri jolly (`*` o `?`) come parte del nome dell'operazione. Ad esempio, il seguente elemento `Action` si applica a tutte le operazioni IAM che includono la stringa `AccessKey`, incluso `CreateAccessKey`, `DeleteAccessKey`, `ListAccessKeys` e `UpdateAccessKey`.

```
"Action": "iam:*AccessKey*"
```

Alcuni servizi ti consentono di limitare le operazioni disponibili. Ad esempio, Amazon SQS consente di rendere disponibile solo un sottoinsieme di tutte le operazioni Amazon SQS possibili. In questo caso, il carattere jolly `*` non ti permette il controllo completo della coda; ti permette solo il sottoinsieme di operazioni che hai condiviso. Per ulteriori informazioni, consulta Informazioni sulle autorizzazioni nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Elementi delle policy JSON IAM: NotAction

`NotAction` è un elemento di policy avanzato che corrisponde esplicitamente a tutte le operazioni tranne quelle specificamente elencate. L'utilizzo di `NotAction` può determinare una policy più breve dal momento che è possibile elencare solo poche operazioni che non devono corrispondere, anziché includere un lungo elenco di operazioni che devono corrispondere. Le azioni specificate in non `NotAction` sono influenzate dall'effetto `Allow` o `Deny` contenuto in una istruzione della policy. Questo significa a sua volta che tutte le operazioni o i servizi applicabili che non sono elencati sono consentiti se utilizzi l'effetto `Allow`. Inoltre, tali operazioni o servizi non elencati vengono negati se utilizzi l'effetto `Deny`. Quando utilizzi `NotAction` con l'elemento `Resource`, fornisci l'ambito della policy. In questo modo si AWS determinano le azioni o i servizi applicabili. Per ulteriori informazioni, consulta la policy di esempio seguente.

NotAction con Allow

È possibile utilizzare l'`NotAction`elemento in un'istruzione con `"Effect": "Allow"` per fornire l'accesso a tutte le azioni di un AWS servizio, ad eccezione delle azioni specificate in`NotAction`. È possibile utilizzarlo con l'elemento `Resource` per fornire l'ambito della policy, limitando le operazioni consentite a quelle che possono essere eseguite sulla risorsa specificata.

L'esempio seguente consente agli utenti di accedere a tutte le operazioni Amazon S3 che possono essere eseguite su qualsiasi risorsa S3 eccetto l'eliminazione di un bucket. Inoltre, questa policy non consente operazioni in altri servizi, perché le operazioni di altri servizi non sono applicabili alle risorse S3.

```
"Effect": "Allow",
"NotAction": "s3:DeleteBucket",
"Resource": "arn:aws:s3:::*",
```

È possibile che talvolta si desideri consentire l'accesso a un numero elevato di operazioni. Utilizzando l'elemento `NotAction` si inverte efficacemente l'istruzione, determinando un elenco più breve di operazioni. Ad esempio, poiché AWS dispone di così tanti servizi, potresti voler creare una policy che consenta all'utente di fare tutto tranne accedere alle azioni IAM.

L'esempio seguente consente agli utenti di accedere a ogni azione in ogni AWS servizio tranne IAM.

```
"Effect": "Allow",
"NotAction": "iam:*",
"Resource": "*"
```

Va prestata attenzione all'utilizzo dell'elemento `NotAction` e `"Effect": "Allow"` nella stessa istruzione o in un'istruzione diversa nella policy. `NotAction` corrisponde a tutti i servizi e le operazioni che non sono esplicitamente elencati o applicabili alla risorsa specificata e può finire col concedere agli utenti più autorizzazioni del previsto.

NotAction con Deny

Puoi utilizzare l'elemento `NotAction` in un'istruzione con `"Effect": "Deny"` per negare l'accesso a tutte le risorse elencate tranne le operazioni specificate nell'elemento `NotAction`. Questa combinazione non consente gli elementi elencati ma invece nega esplicitamente le operazioni non elencate. Devi comunque consentire le operazioni che desideri consentire.

Il seguente esempio condizionale nega l'accesso alle operazioni non IAM se l'utente non ha eseguito l'accesso utilizzando l'autenticazione MFA. Se l'utente ha eseguito l'accesso con l'autenticazione MFA, il test `"Condition"` non riesce e l'istruzione `"Deny"` finale non produce effetti. Nota, tuttavia, che questa istruzione non concederebbe all'utente l'accesso ad alcuna operazione, ma negherebbe solamente in modo esplicito tutte le altre operazioni eccetto le operazioni IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "DenyAllUsersNotUsingMFA",
        "Effect": "Deny",
        "NotAction": "iam:*",
        "Resource": "*",
        "Condition": {"BoolIfExists": {"aws:MultiFactorAuthPresent": "false"}}
    }]
}
```

Per una policy di esempio che nega l'accesso alle operazioni al di fuori di regioni specifiche, ad eccezione delle operazioni di servizi specifici, consulta AWS: nega l'accesso in AWS base alla regione richiesta.

Elementi della policy JSON IAM: Risorsa

In una istruzione di policy IAM, l'elemento `Resource` definisce l'oggetto o gli oggetti a cui si applica l'istruzione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`.

Specifica una risorsa utilizzando un nome della risorsa Amazon (ARN). Il formato dell'ARN dipende dalla Servizio AWS risorsa specifica a cui ti riferisci. Sebbene il formato ARN vari, si utilizza sempre un ARN per identificare una risorsa. Per ulteriori informazioni sul formato di ARN, consulta ARN IAM. Per informazioni su come specificare una risorsa, consulta la documentazione relativa al servizio per la quale desideri scrivere un'istruzione.

> **Note**
>
> Alcuni Servizi AWS non consentono di specificare azioni per singole risorse. In questi casi, tutte le azioni elencate nell'elemento `Action` o `NotAction` si applicano a tutte le risorse di quel servizio. In questo caso, viene utilizzato il carattere jolly (`*`) nell'elemento `Resource`.

L'esempio seguente si riferisce a una determinata coda Amazon SQS.

```
"Resource": "arn:aws:sqs:us-east-2:account-ID-without-hyphens:queue1"
```

L'esempio seguente si riferisce all'utente IAM denominato Bob in un Account AWS.

> **Note**
>
> Nell'elemento `Resource`, il nome utente IAM prevede una distinzione tra lettere minuscole e maiuscole.

```
"Resource": "arn:aws:iam::account-ID-without-hyphens:user/Bob"
```

Utilizzo di caratteri jolly negli ARN delle risorse

È possibile utilizzare caratteri jolly (`*` e `?`) all'interno di singoli segmenti di un ARN (le parti separate da due punti) per rappresentare:

- Qualsiasi combinazione di caratteri (`*`)
- Qualsiasi carattere singolo (`?`)

È possibile utilizzare più caratteri `*` o `?` in ogni segmento. Se il carattere jolly `*` è l'ultimo carattere del segmento dell'ARN di una risorsa, può espandersi fino a superare i limiti dei due punti. Consigliamo di utilizzare i caratteri jolly (`*` e `?`) all'interno dei segmenti dell'ARN separati da due punti.

> **Note**
>
> Non è possibile utilizzare un carattere jolly nel segmento di servizio che identifica il AWS prodotto. Per ulteriori informazioni sui segmenti degli ARN, consulta Identificazione AWS risorse con Amazon Resource Names (ARN)

L'esempio seguente si riferisce a tutti gli utenti IAM il cui percorso è `/accounting`.

```
"Resource": "arn:aws:iam::account-ID-without-hyphens:user/accounting/*"
```

L'esempio seguente si riferisce a tutti gli elementi all'interno di un determinato bucket Amazon S3.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
```

Il carattere asterisco (*) può espandersi per sostituire tutto all'interno di un segmento, inclusi caratteri come una barra (/) che potrebbero sembrare un delimitatore all'interno di un determinato spazio dei nomi del servizio. Ad esempio, considera il seguente ARN di Amazon S3 come la stessa logica di espansione con caratteri jolly si applica a tutti i servizi.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*/test/*"
```

I caratteri jolly nell'ARN si applicano a tutti i seguenti oggetti nel bucket, non solo al primo oggetto elencato.

```
amzn-s3-demo-bucket/1/test/object.jpg
amzn-s3-demo-bucket/1/2/test/object.jpg
amzn-s3-demo-bucket/1/2/test/3/object.jpg
amzn-s3-demo-bucket/1/2/3/test/4/object.jpg
amzn-s3-demo-bucket/1///test///object.jpg
amzn-s3-demo-bucket/1/test/.jpg
amzn-s3-demo-bucket//test/object.jpg
amzn-s3-demo-bucket/1/test/
```

Considera gli ultimi due oggetti dell'elenco precedente. Il nome di un oggetto Amazon S3 può iniziare o terminare validamente con il carattere barra (/) del delimitatore convenzionale. Mentre / funziona come delimitatore, non vi è alcun significato specifico quando questo carattere viene utilizzato all'interno dell'ARN di una risorsa. Viene trattato come qualsiasi altro carattere valido. L'ARN non corrisponde ai seguenti oggetti:

```
amzn-s3-demo-bucket/1-test/object.jpg
amzn-s3-demo-bucket/test/object.jpg
amzn-s3-demo-bucket/1/2/test.jpg
```

Specifica di più risorse

È possibile specificare più risorse nell'elemento `Resource` utilizzando un array di ARN. L'esempio seguente si riferisce a due tabelle DynamoDB.

```
"Resource": [
    "arn:aws:dynamodb:us-east-2:account-ID-without-hyphens:table/books_table",
    "arn:aws:dynamodb:us-east-2:account-ID-without-hyphens:table/magazines_table"
]
```

Utilizzo delle variabili delle policy negli ARN delle risorse

Nell'elemento `Resource`, puoi utilizzare le variabili di policy JSON nella parte dell'ARN che identifica la risorsa specifica (ovvero nella parte finale di ARN). Ad esempio, puoi utilizzare la chiave `{aws:username}` come parte di una risorsa ARN per indicare che l'attuale nome dell'utente deve essere incluso come parte del nome della risorsa. L'esempio seguente mostra come puoi utilizzare la chiave `{aws:username}` in un elemento `Resource`. La policy consente l'accesso a una tabella Amazon DynamoDB che corrisponde al nome dell'utente corrente.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "dynamodb:*",
        "Resource": "arn:aws:dynamodb:us-east-2:111122223333:table/
${aws:username}"
    }
}
```

Per ulteriori informazioni sulle variabili di policy JSON, consultare la pagina Elementi delle policy IAM: variabili e tag.

Elementi della policy JSON IAM: NotResource

`NotResource` è un elemento della policy avanzato che corrisponde esplicitamente a tutte le risorse tranne quelle specificate. L'utilizzo di `NotResource` può risultare in una policy di durata inferiore elencando solo poche risorse che non devono corrispondere, anziché includere un lungo elenco di risorse che corrisponderanno. Ciò è particolarmente utile per le policy che si applicano all'interno di un singolo servizio AWS .

Ad esempio, immaginate di disporre di un gruppo denominato `HRPayroll`. I membri di `HRPayroll` non devono avere il permesso di accedere a qualsiasi risorsa Amazon S3 ad eccezione della cartella `Payroll` nel bucket `HRBucket`. La policy seguente rifiuta esplicitamente l'accesso a tutte le risorse Amazon S3 eccetto a quelle elencate. Tuttavia, questa policy non concede all'utente l'accesso a nessuna risorsa.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "s3:*",
    "NotResource": [
      "arn:aws:s3:::HRBucket/Payroll",
      "arn:aws:s3:::HRBucket/Payroll/*"
    ]
  }
}
```

Di solito, per negare esplicitamente l'accesso a una risorsa è necessario scrivere una policy che utilizza `"Effect":"Deny"` e che include un elemento `Resource` che elenca ogni cartella individualmente. Tuttavia, in tal caso, ogni volta che aggiungi una cartella a `HRBucket` o una risorsa ad Amazon S3 che non deve essere accessibile, è necessario aggiungere il suo nome all'elenco nell'elemento `Resource`. Se si utilizza invece un elemento `NotResource`, agli utenti viene automaticamente negato l'accesso a nuove cartelle a meno che non si aggiungano i nomi delle cartelle all'elemento `NotResource`.

Quando si utilizza `NotResource`, è necessario tenere presente che le risorse specificate in questo elemento sono le uniche risorse a non essere limitate. Questo, a sua volta, limita tutte le risorse che

si applicano all'operazione. Nell'esempio precedente, la policy riguarda solo le operazioni di Amazon S3 e quindi solo le risorse di Amazon S3. Se l'elemento `Action` avesse incluso anche operazioni di Amazon EC2, la policy avrebbe rifiutato l'accesso a qualunque risorsa EC2 non specificata nell'elemento `NotResource`. Per sapere quali azioni in un servizio consentono di specificare l'ARN di una risorsa, vedi Azioni, risorse e chiavi di condizione per i servizi. AWS

NotResource con altri elementi

Non bisognerebbe mai utilizzare insieme gli elementi `"Effect": "Allow"`, `"Action": "*"` e `"NotResource": "arn:aws:s3:::HRBucket"`. Questa dichiarazione è molto pericolosa, perché consente l'accesso a AWS tutte le risorse tranne il bucket `HRBucket` S3. Ciò consentirebbe addirittura a un utente di aggiungere al proprio profilo una policy che gli consenta di accedere a `HRBucket`. Non bisogna farlo.

Va prestata attenzione all'utilizzo dell'elemento `NotResource` e `"Effect": "Allow"` nella stessa istruzione o in un'istruzione diversa nella policy. `NotResource` consente tutti i servizi e risorse che non sono elencati in modo esplicito e può concedere agli utenti più autorizzazioni del previsto. L'utilizzo dell'elemento `NotResource` e `"Effect": "Deny"` nella stessa istruzione nega i servizi e le risorse che non sono elencati in modo esplicito.

Elementi della policy JSON IAM: Condizione

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni di attivazione di una policy. L'elemento `Condition` è facoltativo. Nell'elemento `Condition` è possibile creare espressioni in cui utilizzare operatori condizionali (uguale a, meno di, e altri) per confrontare le chiavi di contesto e i valori della policy rispetto alle chiavi e ai valori del contesto della richiesta. Per ulteriori informazioni sul contesto della richiesta, consultare Componenti di una richiesta.

```
"Condition" : { "{condition-operator}" : { "{condition-key}" : "{condition-value}" }}
```

La chiave di contesto specificata in una condizione della policy può essere una chiave di contesto della condizione globale o una chiave di condizione specifica del servizio. Le chiavi di contesto delle condizioni globali hanno il `aws:` prefisso. Service-specific le chiavi di contesto hanno il prefisso del servizio. Ad esempio, Amazon EC2 consente di scrivere una condizione utilizzando la chiave di contesto `ec2:InstanceType`, univoca per quel servizio. Per visualizzare le chiavi di contesto IAM specifiche del servizio con il prefisso `iam:`, consulta IAM e AWS STS chiavi di contesto delle condizioni.

I nomi delle chiavi di contesto non fanno distinzione tra maiuscole e minuscole. Ad esempio, includere la chiave di `aws:SourceIP` contesto equivale a testare. `AWS:SourceIp` Case-sensitivity il numero di valori delle chiavi di contesto dipende dall'operatore di condizione utilizzato. Ad esempio, la seguente condizione include l'operatore `StringEquals` per rendere possibile la corrispondenza solo delle richieste effettuate da `john`. Agli utenti denominati `John` viene negato l'accesso.

```
"Condition" : { "StringEquals" : { "aws:username" : "john" }}
```

Le seguenti condizione utilizza l'operatore `StringEqualsIgnoreCase` per corrispondere agli utenti denominati `john` o `John`.

```
"Condition" : { "StringEqualsIgnoreCase" : { "aws:username" : "john" }}
```

Alcune chiavi di contesto supportano le coppie chiave-valore che consentono di specificare parte del nome della chiave. Gli esempi includono la chiave di `aws:RequestTag/tag-key` contesto AWS KMS `kms:EncryptionContext:encryption_context_key`, la e la chiave di `ResourceTag/tag-key` contesto supportate da più servizi.

- Se utilizzi la chiave di contesto `ResourceTag/tag-key` per un servizio come Amazon EC2, devi specificare un nome di chiave per la `tag-key`.
- I nomi delle chiavi non fanno distinzione tra maiuscole e minuscole. Questo significa che se specifichi `"aws:ResourceTag/TagKey1": "Value1"` nell'elemento condizione della policy, la condizione corrisponderà a una chiave di tag della risorsa denominata `TagKey1` o `tagkey1`, ma non a entrambe.
- AWS i servizi che supportano questi attributi potrebbero consentire di creare più nomi di chiave che differiscono solo in base alle maiuscole e minuscole. Ad esempio, è possibile applicare un tag a un'istanza Amazon EC2 con `ec2=test1` e `EC2=test2`. Quando utilizzi una condizione come `"aws:ResourceTag/EC2": "test1"` per consentire l'accesso alla risorsa, il nome della chiave corrisponde a entrambi i tag, ma solo a un valore. Questo può causare errori di condizione imprevisti.

> **Important**
>
> Come best practice, verifica che i membri del tuo account seguano una convenzione di denominazione coerente quando assegnano nomi agli attributi di coppie chiave-

valore. Alcuni esempi includono tag o contesti di crittografia AWS KMS . È possibile imporlo utilizzando la chiave di `aws:TagKeys` contesto per l'etichettatura o la chiave `kms:EncryptionContextKeys` per il contesto di AWS KMS crittografia.

- Per un elenco di tutti gli operatori di condizione e per una descrizione del funzionamento di ciascun operatore, consulta Operatori di condizione
- Se non è diversamente specificato, tutte le chiavi di contesto possono avere valori multipli. Per ulteriori informazioni sulla gestione delle chiavi di contesto che dispongono di più valori, consulta Operatori dell'insieme per le chiavi di contesto multivalore
- Per un elenco di tutte le chiavi di contesto disponibili a livello globale, consulta AWS chiavi di contesto delle condizioni globali.
- Per le chiavi di contesto delle condizioni definite da ciascun servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.

Il contesto della richiesta

Quando un principale effettua una richiesta a AWS, AWS raccoglie le informazioni sulla richiesta in un contesto di richiesta. Il contesto della richiesta include informazioni sul principale, sulle risorse, sulle operazioni e su altre proprietà dell'ambiente. La valutazione delle policy confronta le proprietà nella policy con le proprietà inviate nella richiesta per valutare e autorizzare le operazioni che è possibile eseguire in AWS.

È possibile utilizzare l'elemento `Condition` di una policy JSON per testare chiavi di contesto specifiche rispetto al contesto della richiesta. Ad esempio, puoi creare una policy che utilizza la chiave aws: CurrentTime context per consentire a un utente di eseguire azioni solo entro un intervallo di date specifico.

L'esempio seguente mostra una rappresentazione del contesto della richiesta quando Martha Rivera invia una richiesta per disattivare il suo dispositivo MFA.

```
Principal: AROA123456789EXAMPLE
Action: iam:DeactivateMFADevice
Resource: arn:aws:iam::user/martha
Context:
  - aws:UserId=AROA123456789EXAMPLE:martha
  - aws:PrincipalAccount=1123456789012
```

```
- aws:PrincipalOrgId=o-example
- aws:PrincipalARN=arn:aws:iam::1123456789012:assumed-role/TestAR
- aws:MultiFactorAuthPresent=true
- aws:MultiFactorAuthAge=2800
- aws:CurrentTime=...
- aws:EpochTime=...
- aws:SourceIp=...
```

Il contesto della richiesta viene confrontato con una policy che consente agli utenti di rimuovere il proprio dispositivo di autenticazione a più fattori (MFA), ma solo se è stato effettuato l'accesso utilizzando l'MFA nell'ultima ora (3600 secondi).

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Sid": "AllowRemoveMfaOnlyIfRecentMfa",
        "Effect": "Allow",
        "Action": [
            "iam:DeactivateMFADevice"
        ],
        "Resource": "arn:aws:iam::*:user/${aws:username}",
        "Condition": {
            "NumericLessThanEquals": {"aws:MultiFactorAuthAge": "3600"}
        }
    }
}
```

In questo esempio, la policy corrisponde al contesto della richiesta: l'operazione è la stessa, la risorsa corrisponde al carattere jolly "*" e il valore per `aws:MultiFactorAuthAge` è 2800, che è inferiore a 3600, quindi la policy consente questa richiesta di autorizzazione.

AWS valuta ogni chiave di contesto nella policy e restituisce un valore vero o falso. Una chiave di contesto che non è presente nella richiesta è considerata una mancata corrispondenza.

Il contesto della richiesta può restituire i seguenti valori:

- True: se il richiedente ha effettuato l'accesso utilizzando MFA nell'ultima ora o meno, la condizione restituisce true.

- False: se il richiedente ha effettuato l'accesso utilizzando MFA più di un'ora fa, la condizione restituisce false.
  - Non presente: se il richiedente ha effettuato una richiesta utilizzando le proprie chiavi di accesso utente IAM nell' AWS API AWS CLI or, la chiave non è presente. In questo caso, la chiave non è presente e non viene restituita la corrispondenza.

> **Note**
>
> In alcuni casi, quando il valore della chiave della condizione non è presente, la condizione può comunque restituire True. Ad esempio, se si aggiunge il qualificatore `ForAllValues`, la richiesta restituisce True se la chiave di contesto non è inclusa nella richiesta. Per evitare che le chiavi di contesto mancanti o con valori vuoti vengano valutate come True, puoi includere l'Operatore di condizione Null nella tua policy con un valore `false` per verificare che la chiave di contesto esista e che il suo valore non sia Null.

Il blocco condizione

L'esempio seguente mostra il formato di base di un elemento `Condition`:

```
"Condition": {"StringLike": {"s3:prefix": ["jane/*"]}}
```

Un valore dalla richiesta è rappresentato da una chiave di contesto, in questo caso `s3:prefix`. Il valore della chiave di contesto viene confrontato con un valore specificato come valore letterale, ad esempio `jane/*`. Il tipo di confronto da eseguire viene specificato dall'operatore di condizione (in questo caso, `StringLike`). Puoi creare condizioni che confrontano stringhe, date, numeri e altro ancora, utilizzando tipiche comparazioni booleane come ad esempio "uguale a", "maggiore di" e "minore di". Se utilizzi operatori stringa o operatori ARN, puoi utilizzare una variabile di policy nel valore della chiave di contesto. L'esempio seguente include la variabile `aws:username`.

```
"Condition": {"StringLike": {"s3:prefix": ["${aws:username}/*"]}}
```

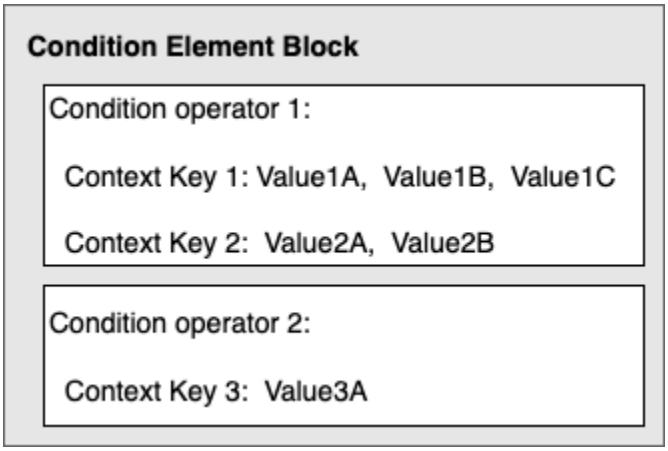
In alcuni casi, le chiavi di contesto possono contenere più valori. Ad esempio, una richiesta ad Amazon DynamoDB potrebbe richiedere la restituzione o l'aggiornamento di più attributi di una tabella. Una policy per l'accesso alle tabelle di DynamoDB può includere la chiave `dynamodb:Attributes` che contiene tutti gli attributi elencati nella richiesta. Puoi testare i vari attributi nella richiesta a fronte di un elenco di attributi consentiti in una policy, utilizzando operatori

predefiniti nell'elemento `Condition`. Per ulteriori informazioni, consulta Operatori dell'insieme per le chiavi di contesto multivalore.

Quando la policy viene valutata durante una richiesta, AWS sostituisce la chiave con il valore corrispondente della richiesta. (In questo esempio, AWS utilizzerebbe la data e l'ora della richiesta.) Dopo la valutazione della condizione, viene restituito un risultato True o False, che viene poi utilizzato per decidere se la policy nel suo complesso deve consentire o rifiutare la richiesta.

Valori multipli in una condizione

Un elemento `Condition` può contenere più operatori di condizioni, ciascuno delle quali può includere a sua volta più coppie chiave-valore. L'immagine seguente illustra questo scenario.



Per ulteriori informazioni, consulta Operatori dell'insieme per le chiavi di contesto multivalore.

Elementi della policy JSON IAM: operatori di condizione

Utilizzare gli operatori di condizione nell'elemento `Condition` per confrontare chiave e valore nella policy con i valori nel contesto della richiesta. Per ulteriori informazioni sull'elemento `Condition`, consultare Elementi della policy JSON IAM: Condizione.

L'operatore di condizione che è possibile utilizzare in una policy dipende dalla chiave di condizione scelta. È possibile scegliere una chiave di condizione globale o una chiave di condizione specifica del servizio. Per informazioni su quale operatore di condizione è possibile utilizzare per una chiave di condizione globale, consultare AWS chiavi di contesto delle condizioni globali. Per sapere quale operatore di condizione è possibile utilizzare per una chiave di condizione specifica del servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi e scegli il servizio che desideri visualizzare.

> **Important**
>
> Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono e la condizione è false. Se la condizione di policy richiede che la chiave sia non abbinata, ad esempio `StringNotLike` o `ArnNotLike` e la chiave giusta non è presente, la condizione è true. Questa logica si applica a tutti gli operatori di condizione tranne... IfExistse controllo ??? Null. Questi operatori testano se la chiave è presente (esiste) nel contesto della richiesta.

Gli operatori di condizione possono essere raggruppati nelle seguenti categorie:

- Stringa
- Numerici
- Data e ora
- Booleano
- Binary
- IP address (Indirizzo IP)
- Amazon Resource Name (ARN) (disponibile solo per alcuni servizi)
- ... IfExists(controlla se il valore della chiave esiste come parte di un altro controllo)
- Verifica Null (controlla se il valore della chiave esiste come controllo autonomo)

Operatori di condizione stringa

Gli operatori di condizioni stringa consentono di creare elementi `Condition` che limitano l'accesso in base al confronto con una chiave con un valore di stringa.

- Variabili di policy: supportate
- Caratteri jolly: supportati

Operatore di condizione	Description
`StringEquals`	Corrispondenza esatta, con distinzione maiuscole/minuscole
`StringNotEquals`	Corrispondenza negativa

Operatore di condizione	Description
`StringEqualsIgnoreCase`	Corrispondenza esatta, senza distinzione maiuscole/minuscole
`StringNotEqualsIgnoreCase`	Corrispondenza negativa, senza distinzione maiuscole/minuscole
`StringLike`	Case-sensitive corrispondente. I valori possono includere una corrispondenza con più caratteri jolly (*) e un singolo carattere jolly (?) in qualsiasi punto della stringa. Per ottenere corrispondenze di stringhe parziali devi specificare caratteri jolly. **Note** Se una chiave contiene più valori, `StringLike` può essere qualificato con gli operatori su set: `ForAllValues:StringLike` e `ForAnyValue:StringLike`. Per ulteriori informazioni, consulta Operatori dell'insieme per le chiavi di contesto multivalore.
`StringNotLike`	Corrispondenza negativa, con distinzione maiuscole/minuscole. I valori possono includere una corrispondenza con più caratteri jolly (*) o un singolo carattere jolly (?) in qualsiasi punto della stringa.

Example operatore di condizione stringa

Ad esempio, l'istruzione seguente contiene un elemento `Condition` che utilizza la chiave `aws:PrincipalTag` per specificare che il principale che effettua la richiesta deve essere contrassegnato con la categoria di processo `iamuser-admin`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
```

```
        "Effect": "Allow",
        "Action": "iam:*AccessKey*",
        "Resource": "arn:aws:iam::111122223333:user/*",
        "Condition": {
            "StringEquals": {
                "aws:PrincipalTag/job-category": "iamuser-admin"
            }
        }
    }
}
```

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. In questo esempio, la chiave `aws:PrincipalTag/job-category` è presente nel contesto della richiesta se il principale utilizza un utente IAM con tag collegati. È inclusa anche per un principale che utilizza un ruolo IAM con tag collegati o tag di sessione. Se un utente senza il tag tenta di visualizzare o modificare una chiave di accesso, la condizione restituisce `false` e la richiesta viene negata implicitamente da questa istruzione.

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"StringEquals": { "aws:PrincipalTag/job-category": "iamuser-admin" }`	`aws:PrincipalTag/job-category: - iamuser-admin`	Match
`"StringEquals": { "aws:PrincipalTag/job-category": "iamuser-admin" }`	`aws:PrincipalTag/job-category: - dev-ops`	Nessuna corrispondenza
`"StringEquals": { "aws:PrincipalTag/job-category": "iamuser-admin"`	`aws:PrincipalTag/job-category` non è	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`}`	presente nel contesto della richiesta.	

Example utilizzo di una variabile di policy con un operatore di condizione stringa

Nell'esempio seguente viene utilizzato l'operatore di condizione `StringLike` per eseguire il confronto con una variabile di policy per creare una policy che consente a un utente IAM di utilizzare la console Amazon S3 per gestire la propria "directory principale" in un bucket Amazon S3. La policy consente le operazioni specificate in un bucket S3 a condizione che `s3:prefix` corrisponda a uno qualsiasi dei modelli specificati.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "",
            "home/",
            "home/${aws:username}/"
          ]
        }
      }
    },
    {
```

```
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/home/${aws:username}",
        "arn:aws:s3:::amzn-s3-demo-bucket/home/${aws:username}/*"
      ]
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica per i diversi utenti in base al aws:username valore nel contesto della richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"StringLike": { "s3:prefix": [ "home/", "home/${aws:username}/" ] }`	`aws:username: - martha_rivera`	`"StringLike": { "s3:prefix": [ "home/", "home/martha_rivera /" ] }`
`"StringLike": { "s3:prefix": [ "home/", "home/${aws:username}/" ] }`	`aws:username: - nikki_wolf`	`"StringLike": { "s3:prefix": [ "home/", "home/nikki_wolf /" ] }`
`"StringLike": { "s3:prefix": [ "home/", "home/${aws:username}/" ] }`	`aws:username` non è presente nel contesto della richiesta.	Nessuna corrispondenza

Per un esempio di una policy che mostra come usare l'elemento `Condition` per limitare l'accesso alle risorse in base a un ID applicazione e un ID utente per la federazione OIDC, consulta Amazon S3: consente agli utenti di Amazon Cognito di accedere a oggetti nel relativo bucket.

Operatori di condizione stringa multivalore

Se una chiave nella richiesta contiene più valori, gli operatori di stringa possono essere qualificati con gli operatori dell'insieme `ForAllValues` e `ForAnyValue`. Per ulteriori informazioni sulla logica di valutazione di più chiavi o valori di contesto, consulta Operatori dell'insieme per le chiavi di contesto multivalore.

Operatore di condizione	Description

`ForAllValues:StringEquals`
 	
`ForAllValues:StringEqualsIgnoreCase`
	Tutti i valori per la chiave di condizione nella richiesta devono corrispondere ad almeno uno dei valori della policy.

`ForAnyValue:StringEquals`
 	
`ForAnyValue:StringEqualsIgnoreCase`
	Almeno un valore per la chiave di condizione nella richiesta deve corrispondere a uno dei valori della policy.

`ForAllValues:StringNotEquals`
 	
`ForAllValues:StringNotEqualsIgnoreCase`
	Corrispondenza negativa.
	
	Nessuno dei valori per la chiave di contesto nella richiesta può corrispondere a nessuno dei valori della chiave di contesto della policy.

`ForAnyValue:StringNotEquals`
 	
`ForAnyValue:StringNotEqualsIgnoreCase`
	Corrispondenza negativa.
	
	Almeno un valore per la chiave di contesto nella richiesta NON deve corrispondere a uno dei valori nella chiave di contesto della policy.

`ForAllValues:StringLike`	Tutti i valori per la chiave di condizione nella richiesta devono corrispondere ad almeno uno dei valori della policy.

Operatore di condizione	Description
`ForAnyValue:StringLike`	Almeno un valore per la chiave di condizione nella richiesta deve corrispondere a uno dei valori della policy.
`ForAllValues:StringNotLike`	Corrispondenza negativa. Nessuno dei valori per la chiave di contesto nella richiesta può corrispondere a nessuno dei valori della chiave di contesto della policy.
`ForAnyValue:StringNotLike`	Corrispondenza negativa. Almeno un valore per la chiave di contesto nella richiesta NON deve corrispondere a uno dei valori nella chiave di contesto della policy.

Example utilizzo di **`ForAnyValue`** con un operatore di condizione stringa

Questo esempio mostra come creare una policy basata sull'identità che consenta di utilizzare l'operazione `CreateTags` di Amazon EC2 per assegnare tag a un'istanza. Quando utilizzi `StringEqualsIgnoreCase`, puoi collegare i tag solo se il tag contiene la chiave `environment` con i valori `preprod` o `storage`. Quando aggiungi `IgnoreCase` all'operatore, consenti a qualsiasi utilizzo delle maiuscole del valore del tag esistente, ad esempio `preprod`, `Preprod` e `PreProd`, di risultare True.

Quando aggiungi il modificatore `ForAnyValue` con la chiave di condizione ha detto: TagKeys, almeno un valore della chiave di tag nella richiesta deve corrispondere al valore `environment`. Il confronto `ForAnyValue` fa distinzione tra maiuscole e minuscole, il che impedisce agli utenti di utilizzare le maiuscole e minuscole errate per la chiave del tag, ad esempio utilizzare `Environment` invece che `environment`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
```

```
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
      "StringEqualsIgnoreCase": {
        "aws:RequestTag/environment": [
          "preprod",
          "storage"
        ]
      },
      "ForAnyValue:StringEquals": {
        "aws:TagKeys": "environment"
      }
    }
  }
}
```

La tabella seguente mostra come AWS valuta questo criterio in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"StringEqualsIgnoreCase": { "aws:RequestTag/environment": [ "preprod", "storage" ] }, "ForAnyValue:StringEquals": { "aws:TagKeys": "environment" }`	`aws:TagKeys: - environment aws:RequestTag/environment: - preprod`	Match
`"StringEqualsIgnoreCase": { "aws:RequestTag/environment": [`	`aws:TagKeys: - environment - costcenter`	Match

Condizione della policy	Contesto della richiesta	Risultato

"preprod",
"storage"
]
},
"ForAnyValue:StringEqu
als": {
"aws:TagKeys":
"environment"
}
	aws:RequestTag/e
	nvironment:
	- PreProd	

"StringEqualsIgnor
eCase": {
"aws:RequestTag/en
vironment": [
"preprod",
"storage"
]
},
"ForAnyValue:StringEqu
als": {
"aws:TagKeys":
"environment"
}
	aws:TagKeys:
	- Environment
	aws:RequestTag/
	Environment:
	- preprod
		Nessuna corrispondenza

"StringEqualsIgnor
eCase": {
"aws:RequestTag/en
vironment": [
"preprod",
"storage"
]
},
"ForAnyValue:StringEqu
als": {
"aws:TagKeys":
"environment"
}
	aws:TagKeys:
	- costcenter
	aws:RequestTag/e
	nvironment:
	- preprod
		Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`"StringEqualsIgnoreCase": { "aws:RequestTag/environment": [ "preprod", "storage" ] }, "ForAnyValue:StringEquals": { "aws:TagKeys": "environment" }`	`aws:TagKeys` non è presente nel contesto della richiesta. `aws:RequestTag/environment: - storage`	Nessuna corrispondenza
`"StringEqualsIgnoreCase": { "aws:RequestTag/environment": [ "preprod", "storage" ] }, "ForAnyValue:StringEquals": { "aws:TagKeys": "environment" }`	`aws:TagKeys: - environment` `aws:RequestTag/environment` non è presente nel contesto della richiesta.	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`"StringEqualsIgnoreCase": { "aws:RequestTag/environment": [ "preprod", "storage" ] }, "ForAnyValue:StringEquals": { "aws:TagKeys": "environment" }`	`aws:TagKeys` non è presente nel contesto della richiesta. `aws:RequestTag/environment` non è presente nel contesto della richiesta.	Nessuna corrispondenza

Corrispondenza dei caratteri jolly

Gli operatori di condizioni di stringa eseguono una corrispondenza senza modello che non impone un formato predefinito. Gli operatori di condizione ARN e Date sono un sottoinsieme di operatori di stringa che impongono una struttura al valore della chiave della condizione.

Ti consigliamo di utilizzare operatori di condizione che corrispondono ai valori con cui stai confrontando le chiavi. Ad esempio, è necessario utilizzare the section called "Operatori di condizione stringa" quando si confrontano le chiavi con i valori delle stringhe. Allo stesso modo, è necessario utilizzare the section called "Operatori di condizione con Amazon Resource Name (ARN)" quando si confrontano le chiavi con i valori ARN.

Example

In questo esempio viene illustrato come creare un limite intorno alle risorse nell'organizzazione. La condizione di questa policy nega l'accesso alle operazioni di Amazon S3 a meno che la risorsa a cui si accede non si trovi in un set specifico di unità organizzative (UO) in AWS Organizations. Un percorso AWS Organizations è una rappresentazione testuale della struttura dell'entità di un'organizzazione.

La condizione richiede che `era: ResourceOrgPaths` contenga uno dei percorsi di unità organizzative elencati. Poiché `aws:ResourceOrgPaths` rappresenta una condizione multivalore,

la policy utilizza l'operatore `ForAllValues:StringNotLike` per confrontare i valori di `aws:ResourceOrgPaths` con l'elenco di unità organizzative contenuto nella policy.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3AccessOutsideMyBoundary",
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringNotLike": {
          "aws:ResourceOrgPaths": [
            "o-acorg/r-acroot/ou-acroot-mediaou/",
            "o-acorg/r-acroot/ou-acroot-sportsou/*"
          ]
        }
      }
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"ForAllValues:StringNotLike": { "aws:ResourceOrgPaths": [ "o-acorg/r-acroot/ou-acroot-mediaou/", "o-acorg/r-acroot/ou-acroot-sportsou/*"`	`aws:ResourceOrgPaths: - o-acorg/r-acroot/ou-acroot-sportsou/costcenter/`	Match

Condizione della policy	Contesto della richiesta	Risultato
] }		

"ForAllValues:StringNotLike": {
"aws:ResourceOrgPaths": [
"o-acorg/r-acroot/ou-acroot-mediaou/",
"o-acorg/r-acroot/ou-acroot-sportsou/*"
]
}
	aws:ResourceOrgPaths:
	- o-acorg/r-acroot/ou-acroot-mediaou/costcenter/
		Nessuna corrispondenza

"ForAllValues:StringNotLike": {
"aws:ResourceOrgPaths": [
"o-acorg/r-acroot/ou-acroot-mediaou/",
"o-acorg/r-acroot/ou-acroot-sportsou/*"
]
}
	Nessun aws:ResourceOrgPaths: nella richiesta.	Nessuna corrispondenza

Operatori di condizione numerici

Gli operatori di condizione numerici consentono di creare elementi `Condition` che limitano l'accesso in base al confronto di una chiave con un valore intero o decimale.

- Variabili di policy: non supportate
- Caratteri jolly: non supportati

Operatore di condizione	Description
`NumericEquals`	Corrispondenza

Operatore di condizione	Description
`NumericNotEquals`	Corrispondenza negativa
`NumericLessThan`	Corrispondenza "Minore di"
`NumericLessThanEqu als`	Corrispondenza "Minore di o uguale a"
`NumericGreaterThan`	Corrispondenza "Maggiore di"
`NumericGreaterThan Equals`	Corrispondenza "Maggiore di o uguale a"

Ad esempio, la seguente istruzione contiene un elemento `Condition` che utilizza l'operatore di condizione `NumericLessThanEquals` con la chiave `s3:max-keys` per specificare che il richiedente può elencare fino a oggetti in `amzn-s3-demo-bucket` alla volta.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
    "Condition": {"NumericLessThanEquals": {"s3:max-keys": "10"}}
  }
}
```

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. In questo esempio, la chiave `s3:max-keys` è sempre presente nella richiesta quando si esegue l'operazione `ListBucket`. Se questa policy consentiva tutte le operazioni Amazon S3, saranno consentite solo le operazioni che includono la chiave di contesto `max-keys` con un valore minore o uguale a 10.

Operatori di condizione data

Gli operatori con condizioni di data consentono di creare `Condition` elementi che limitano l'accesso in base al confronto tra una chiave e un date/time valore. Utilizza questi operatori di condizione con la chiave `aws:CurrentTime` o la chiave `aws:EpochTime`. È necessario specificare date/time i valori con una delle implementazioni W3C dei formati di data ISO 8601 o in epoca (UNIX).

- Variabili di policy: non supportate
- Caratteri jolly: non supportati

Operatore di condizione	Description
`DateEquals`	Corrispondenza con una data specifica
`DateNotEquals`	Corrispondenza negativa
`DateLessThan`	Corrispondenza prima di una determinata data e ora
`DateLessThanEquals`	Corrispondenza a una determinata data e ora
`DateGreaterThan`	Corrispondenza dopo una determinata data e ora
`DateGreaterThanEqu als`	Corrispondenza a una determinata data e ora o successiva

Ad esempio, l'istruzione seguente contiene un elemento `Condition` che utilizza l'operatore di condizione `DateGreaterThan` con la chiave `aws:TokenIssueTime`. Questa condizione specifica che le credenziali di sicurezza temporanee utilizzate per effettuare la richiesta sono state emesse nel 2020. Questa policy può essere aggiornata ogni giorno a livello di codice per garantire che i membri dell'account utilizzino nuove credenziali.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "iam:*AccessKey*",
```

```
        "Resource": "arn:aws:iam::111122223333:user/*",
        "Condition": {
            "DateGreaterThan": {
                "aws:TokenIssueTime": "2020-01-01T00:00:01Z"
            }
        }
    }
}
```

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. La chiave `aws:TokenIssueTime` è presente nel contesto della richiesta solo quando il principale utilizza le credenziali temporanee per effettuare la richiesta. La chiave non è presente nelle AWS CLI richieste AWS API o AWS SDK effettuate utilizzando le chiavi di accesso. In questo esempio, se un utente IAM prova a visualizzare o modificare una chiave di accesso, la richiesta viene rifiutata.

Operatori di condizione booleani

Le condizioni booleane consentono di creare elementi `Condition` che limitano l'accesso in base al confronto con una chiave con valore `true` o `false`.

Se una chiave contiene più valori, gli operatori booleani possono essere qualificati con gli operatori dell'insieme `ForAllValues` e `ForAnyValue`. Per ulteriori informazioni sulla logica di valutazione di più chiavi o valori di contesto, consulta Operatori dell'insieme per le chiavi di contesto multivalore.

- Variabili di policy: supportate
- Caratteri jolly: non supportati

Operatore di condizione	Description
`Bool`	Corrispondenza booleana
`ForAllValues:Bool`	Da utilizzare con il tipo di dati array di valori booleani. Tutti i valori booleani inclusi nei valori chiave di contesto devono corrispondere ai valori booleani della policy.

Operatore di condizione	Description
	Per evitare che gli operatori `ForAllValues` valutino come Consentit a una chiave di contesto mancante o con valore vuoto, puoi includere nella policy l'Operatore di condizione Null.
`ForAnyValue:Bool`	Da utilizzare con il tipo di dati array di valori booleani. Almeno uno dei valori booleani nella chiave di contesto deve corrispondere ai valori booleani della policy.

Example operatore di condizione booleana

La seguente policy basata sull'identità utilizza l'operatore di condizione `Bool` con la chiave `aws:SecureTransport` per negare la replica degli oggetti e dei tag degli oggetti nel bucket di destinazione e i rispettivi contenuti se la richiesta non è su SSL.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "BooleanExample",
      "Action": "s3:ReplicateObject",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
```

```
      }
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"Bool": { "aws:SecureTranspo rt": "false" }`	`aws:SecureTransport: - false`	Match
`"Bool": { "aws:SecureTranspo rt": "false" }`	`aws:SecureTransport: - true`	Nessuna corrispondenza
`"Bool": { "aws:SecureTranspo rt": "false" }`	`aws:SecureTransport` non è presente nel contesto della richiesta.	Nessuna corrispondenza

Operatori di condizione binari

L'operatore di condizione `BinaryEquals` consente di creare elementi `Condition` che testano i valori di chiave in formato binario. Viene effettuato un confronto del valore del byte di chiave specificato per il byte con una rappresentazione codificata in base 64 nella policy. Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono.

- Variabili di policy: non supportate
- Caratteri jolly: non supportati

```
"Condition" : {
  "BinaryEquals": {
    "key" : "QmluYXJ5VmFsdWVJbkJhc2U2NA=="
  }
}
```

Condizione della policy	Contesto della richiesta	Risultato
`"BinaryEquals": { "key" : "QmluYXJ5 VmFsdWVJbkJhc2U2NA==" }`	`key: - QmluYXJ5VmFsdWVJbk Jhc2U2NA==`	Match
`"BinaryEquals": { "key" : "QmluYXJ5 VmFsdWVJbkJhc2U2NA==" }`	`key: - ASIAIOSFODNN7EXAMP LE`	Nessuna corrispondenza
`"BinaryEquals": { "key" : "QmluYXJ5 VmFsdWVJbkJhc2U2NA==" }`	`key` non è presente nel contesto della richiesta.	Nessuna corrispondenza

Operatori di condizione con indirizzo IP

Gli operatori di condizione con indirizzo IP consentono di creare elementi `Condition` che limitano l'accesso in base al confronto di una chiave con un indirizzo IPv4 o IPv6 o un intervallo di indirizzi IP. È possibile utilizzare questi operatori con la chiave `aws:SourceIp`. Il valore deve essere nel formato CIDR standard (ad esempio, 203.0.113). 0/24 o 2001: DB8:1234:5678 :/64). Se si specifica un indirizzo IP senza il prefisso di instradamento associato, IAM utilizza il valore predefinito di prefisso `/32`.

Alcuni AWS servizi supportano l'IPv6, utilizzando:: per rappresentare un intervallo di 0. Per sapere se un servizio supporta IPv6, consultare la documentazione per tale servizio.

- Variabili di policy: non supportate
- Caratteri jolly: non supportati

Operatore di condizione	Description
`IpAddress`	L'indirizzo o l'intervallo IP specificato
`NotIpAddress`	Tutti gli indirizzi IP tranne l'indirizzo o l'intervallo IP specificato

Example Operatore di condizione con indirizzo IP

La seguente istruzione utilizza l'operatore di condizione `IpAddress` con la chiave `aws:SourceIp` per specificare che la richiesta deve provenire dall'intervallo IP da 203.0.113.0 a 203.0.113.255.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "iam:*AccessKey*",
        "Resource": "arn:aws:iam::111122223333:user/*",
        "Condition": {
            "IpAddress": {
                "aws:SourceIp": "203.0.113.0/24"
            }
        }
    }
}
```

La chiave di condizione `aws:SourceIp` risolve l'indirizzo IP da cui ha origine la richiesta. Se le richieste provengono da un'istanza Amazon EC2, `aws:SourceIp` viene valutato con l'indirizzo IP pubblico dell'istanza.

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. La chiave `aws:SourceIp` è sempre presente nel contesto della richiesta, tranne quando il richiedente utilizza un endpoint VPC per effettuare la richiesta. In questo caso, la condizione restituisce `false` e la richiesta è implicitamente rifiutata da questa istruzione.

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"IpAddress": { "aws:SourceIp": "203.0.113.0/24" }`	`aws:SourceIp: - 203.0.113.1`	Match
`"IpAddress": { "aws:SourceIp": "203.0.113.0/24" }`	`aws:SourceIp: - 198.51.100.1`	Nessuna corrispondenza

Nella policy di bucket di esempio riportata di seguito viene mostrato come combinare gli indirizzi IPv4 e IPv6 per coprire tutti gli indirizzi IP validi dell'organizzazione. Ti consigliamo di aggiornare le policy dell'organizzazione con gli intervalli di indirizzi IPv6 in aggiunta agli intervalli IPv4 di cui già disponi per garantire che le policy continuino a funzionare mentre effettui la transizione a IPv6.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "someservice:*",
    "Resource": "*",
    "Condition": {
      "IpAddress": {
        "aws:SourceIp": [
          "203.0.113.0/24",
          "2001:DB8:1234:5678::/64"
        ]
      }
    }
  }
}
```

La chiave di condizione `aws:SourceIp` funziona solo in una policy JSON se si chiama l'API testata direttamente come utente. Se si utilizza invece un servizio per chiamare il servizio di destinazione per conto dell'utente, il servizio di destinazione visualizza l'indirizzo IP del servizio chiamante anziché l'indirizzo IP dell'utente originario. Ciò può accadere, ad esempio, se si utilizza AWS CloudFormation per chiamare Amazon EC2 per creare istanze per tuo conto. Attualmente non è disponibile alcun modo per passare l'indirizzo IP di origine tramite un servizio di chiamata al servizio di destinazione per la valutazione in una policy JSON. Per questi tipi di chiamate API di servizi, non utilizzare la chiave di condizione `aws:SourceIp`.

Operatori di condizione con Amazon Resource Name (ARN)

Gli operatori di condizione con ARN (Amazon Resource Name) consentono di creare elementi `Condition` che limitano l'accesso in base al confronto di una chiave con un ARN. L'ARN è considerato una stringa.

- Variabili di policy: supportate
- Caratteri jolly: supportati

Operatore di condizione	Description
`ArnEquals` , `ArnLike`	Case-sensitive corrispondenza dell'ARN. Ciascuno dei sei component i delimitati da due punti dell'ARN viene verificato separatamente e ognuno di essi può includere più caratteri jolly (*) o un singolo carattere jolly (?) corrispondenti. Gli operatori di condizione `ArnEquals` e `ArnLike` si comportano allo stesso modo.
`ArnNotEquals` , `ArnNotLike`	Corrispondenza negativa per l'ARN. Gli operatori di condizione `ArnNotEquals` e `ArnNotLike` si comportano allo stesso modo.

Example Operatore di condizione ARN

Nell'esempio di policy basata sulle risorse riportato di seguito viene illustrata una policy collegata a una coda Amazon SQS a cui si desidera inviare messaggi SNS. La policy autorizza Amazon SNS a inviare messaggi alla coda (o alle code) di propria scelta, ma solo se il servizio invia i messaggi per conto di un determinato argomento (o argomenti) di Amazon SNS. È possibile specificare la coda nel campo `Resource` e l'argomento Amazon SNS come valore per la chiave `SourceArn`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "sns.amazonaws.com"
        },
        "Action": "SQS:SendMessage",
        "Resource": "arn:aws:sqs:us-east-1:123456789012:QUEUE-ID",
        "Condition": {
            "ArnEquals": {
                "aws:SourceArn": "arn:aws:sns:us-east-1:123456789012:TOPIC-ID"
            }
        }
    }
}
```

La chiave `aws:SourceArn` è presente nel contesto della richiesta solo se una risorsa attiva un servizio per chiamare un altro servizio per conto del proprietario della risorsa. Se un utente IAM prova a eseguire direttamente questa operazione, la condizione restituisce `false` e la richiesta viene rifiutata implicitamente da questa istruzione.

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"ArnEquals": { "aws:SourceArn": "arn:aws:sns:us-west-2:123456789012:TOPIC-ID" }`	`aws:SourceArn: - arn:aws:sns:us-west-2:123456789012:TOPIC-ID`	Match
`"ArnEquals": { "aws:SourceArn": "arn:aws:sns:us-we`	`aws:SourceArn:`	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`st-2:123456789012:TOPIC-ID" }`	`- arn:aws:sns:us-west-2:777788889999:TOPIC-ID`	
`"ArnEquals": { "aws:SourceArn": "arn:aws:sns:us-west-2:123456789012:TOPIC-ID" }`	`aws:SourceArn` non è presente nel contesto della richiesta.	Nessuna corrispondenza

Operatori di condizione ARN multivalore

Se una chiave nella richiesta contiene più valori, gli operatori ARN possono essere qualificati con gli operatori dell'insieme `ForAllValues` e `ForAnyValue`. Per ulteriori informazioni sulla logica di valutazione di più chiavi o valori di contesto, consulta Operatori dell'insieme per le chiavi di contesto multivalore.

Operatore di condizione	Description
`ForAllValues:ArnEquals` `ForAllValues:ArnLike`	Tutti gli ARN nel contesto della richiesta devono corrispondere ad almeno uno dei modelli ARN della policy.
`ForAnyValue:ArnEquals` `ForAnyValue:ArnLike`	Almeno un ARN nel contesto della richiesta deve corrispondere a uno dei modelli ARN della policy.
`ForAllValues:ArnNotEquals` `ForAllValues:ArnNotLike`	Corrispondenza negativa. Nessuno degli ARN nel contesto della richiesta può corrispondere a nessun modello ARN di stringa nella policy.
`ForAnyValue:ArnNotEquals` `ForAnyValue:ArnNotLike`	Corrispondenza negativa.

Operatore di condizione	Description
	Almeno un ARN nel contesto della richiesta NON deve corrispondere a nessuno dei modelli ARN della policy.

Example utilizzo di **`ForAllValues`** con un operatore di condizione ARN

L'esempio seguente viene utilizzato `ForAllValues:ArnLike` per creare o aggiornare una fonte di distribuzione logica per i CloudWatch log di Amazon Logs. Il blocco di condizione include la chiave di condizione `logs:LogGeneratingResourceArns` per filtrare gli ARN delle risorse che generano log passati nella richiesta. Utilizzando questo operatore di condizione, tutti gli ARN nella richiesta devono corrispondere ad almeno un ARN nella policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "logs:PutDeliverySource",
            "Resource": "arn:aws:logs:us-east-1:123456789012:delivery-source:*",
            "Condition": {
                "ForAllValues:ArnLike": {
                    "logs:LogGeneratingResourceArns": [
                        "arn:aws:cloudfront::123456789012:distribution/*",
                        "arn:aws:cloudfront::123456789012:distribution/support*"
                    ]
                }
            }
        }
    ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori delle chiavi di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"ForAllValues:ArnLike": { "logs:LogGeneratingResourceArns": [ "arn:aws::cloudfront:123456789012:distribution/*", "arn:aws::cloudfront:123456789012:distribution/support*" ] }`	`logs:LogGeneratingResourceArns: - arn:aws::cloudfront:123456789012:distribution/costcenter`	Match
`"ForAllValues:ArnLike": { "logs:LogGeneratingResourceArns": [ "arn:aws::cloudfront:123456789012:distribution/*", "arn:aws::cloudfront:123456789012:distribution/support*" ] }`	`logs:LogGeneratingResourceArns: - arn:aws::cloudfront:123456789012:distribution/costcenter - arn:aws::cloudfront:123456789012:distribution/support2025`	Match
`"ForAllValues:ArnLike": { "logs:LogGeneratingResourceArns": [ "arn:aws::cloudfront:123456789012:distribution/*", "arn:aws::cloudfront:123456789012:distribution/support*" ] }`	`logs:LogGeneratingResourceArns: - arn:aws::cloudfront:123456789012:distribution/costcenter - arn:aws::cloudfront:123456789012:distribution/admin`	Match

Condizione della policy	Contesto della richiesta	Risultato
`"ForAllValues:ArnLike": { "logs:LogGeneratingResourceArns": [ "arn:aws::cloudfront:123456789012:distribution/*", "arn:aws::cloudfront:123456789012:distribution/support*" ] }`	`logs:LogGeneratingResourceArns: - arn:aws::cloudfront:777788889999:distribution/costcenter`	Nessuna corrispondenza
`"ForAllValues:ArnLike": { "logs:LogGeneratingResourceArns": [ "arn:aws::cloudfront:123456789012:distribution/*", "arn:aws::cloudfront:123456789012:distribution/support*" ] }`	`logs:LogGeneratingResourceArns` non è presente nel contesto della richiesta.	Match

Il qualificatore `ForAllValues` restituisce True se non ci sono chiavi di contesto nella richiesta o se i valori delle chiavi si riducono a un set di dati Null, ad esempio una stringa vuota. Per evitare che le chiavi di contesto mancanti o con valori vuoti vengano valutate come True, puoi includere l'Operatore di condizione Null nella tua policy con un valore `false` per verificare che la chiave di contesto esista e che il suo valore non sia Null.

... IfExists operatori di condizione

È possibile aggiungere `IfExists` alla fine di qualsiasi nome operatore di condizione ad eccezione della condizione `Null`, ad esempio `StringLikeIfExists`. Questa aggiunta ha lo scopo di dichiarare che "se la chiave di policy è presente nel contesto della richiesta, la chiave deve essere

elaborata come specificato nella policy". Se la chiave non è presente, l'elemento della condizione viene valutato come "true". Altri elementi di condizione nell'istruzione possono comunque risultare in una mancata corrispondenza, ma non una chiave mancante se verificata tramite `...IfExists`. Se stai utilizzando un elemento `"Effect": "Deny"` con un operatore di condizione negato come `StringNotEqualsIfExists`, la richiesta viene comunque negata anche se manca la chiave di condizione.

Esempio di utilizzo di **`IfExists`**

Molte chiavi di condizione descrivono informazioni su un determinato tipo di risorsa e sono presenti solo quando si accede a tale tipo di risorsa. Queste chiavi di condizione non sono presenti in altri tipi di risorse. Questo non causa problemi se l'istruzione della policy si applica a un solo tipo di risorsa. Tuttavia, esistono casi in cui una singola istruzione può essere applicata a più tipi di risorse, ad esempio quando l'istruzione della policy fa riferimento a operazioni di più servizi o quando una determinata operazione all'interno di un servizio accede a diversi tipi di risorse all'interno dello stesso servizio. In questi casi, l'inclusione di una chiave di condizione che si applica solo a una delle risorse nell'istruzione della policy può causare un errore dell'elemento `Condition` nell'istruzione della policy in modo tale che l'elemento `"Effect"` dell'istruzione non si applichi.

Ad esempio, considerare il seguente esempio di policy:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Sid": "THISPOLICYDOESNOTWORK",
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "*",
    "Condition": {"StringLike": {"ec2:InstanceType": [
      "t1.*",
      "t2.*",
      "m3.*"
    ]}}
  }
}
```

Lo scopo della policy precedente è di consentire all'utente di avviare qualsiasi istanza di tipo `t1`, `t2` o `m3`. Tuttavia, l'avvio di un'istanza richiede l'accesso a molte risorse oltre all'istanza stessa; ad esempio, immagini, coppie di chiavi, gruppi di sicurezza e così via. L'intera istruzione viene valutata rispetto a ogni risorsa necessaria per avviare l'istanza. Queste risorse aggiuntive non includono la chiave di condizione `ec2:InstanceType`, pertanto il controllo `StringLike` ha esito negativo e all'utente non è concessa la possibilità di avviare nessun tipo di istanza.

Per risolvere questo problema, utilizzare l'operatore di condizione `StringLikeIfExists`. In questo modo, il test viene effettuato solo se la chiave di condizione esiste. È possibile leggere la policy seguente come: "Se la risorsa da verificare include una chiave di condizione "`ec2:InstanceType`", permettere l'operazione solo se il valore della chiave inizia con `t1.`, `t2.` o `m3.`. Se la risorsa verificata non include quella chiave di condizione, non preoccuparti". L'asterisco (*) nei valori della chiave della condizione, se utilizzato con l'operatore di condizione `StringLikeIfExists`, viene interpretato come un jolly per ottenere corrispondenze parziali tra le stringhe. L'istruzione `DescribeActions` include le azioni necessarie per visualizzare l'istanza nella console.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "RunInstance",
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
        "StringLikeIfExists": {
          "ec2:InstanceType": [
            "t1.*",
            "t2.*",
            "m3.*"
          ]
        }
      }
    },
    {
      "Sid": "DescribeActions",
      "Effect": "Allow",
      "Action": [
```

```
        "ec2:DescribeImages",
        "ec2:DescribeInstances",
        "ec2:DescribeVpcs",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori delle chiavi di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"StringLikeIfExists": { "ec2:InstanceType": [ "t1.*", "t2.*", "m3.*" ] }`	`ec2:InstanceType: - t1.micro`	Match
`"StringLikeIfExists": { "ec2:InstanceType": [ "t1.*", "t2.*", "m3.*" ] }`	`ec2:InstanceType: - m2.micro`	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`"StringLikeIfExists": { "ec2:InstanceType": [ "t1.*", "t2.*", "m3.*" ] }`	`ec2:InstanceType` non è presente nel contesto della richiesta.	Match

Operatore di condizione per verificare la presenza di chiavi di condizione

Utilizza un operatore di condizione `Null` per verificare se una chiave di condizione è presente o meno al momento dell'autorizzazione. Nell'istruzione della policy, utilizza `true` (la chiave non esiste, è null) o `false` (la chiave esiste e il suo valore non è null).

Puoi utilizzare una variabile di policy con l'operatore di condizione `Null`.

Ad esempio, è possibile utilizzare questo operatore di condizione per determinare se un utente sta utilizzando credenziali temporanee oppure le proprie credenziali per effettuare una richiesta. Se l'utente utilizza credenziali temporanee, la chiave `aws:TokenIssueTime` è presente e ha un valore. L'esempio seguente mostra una condizione che indica che l'utente deve utilizzare credenziali temporanee (la chiave non può essere assente) affinché l'utente possa utilizzare l'API Amazon EC2.

JSON

```
{
  "Version":"2012-10-17",
  "Statement":{
      "Action":"ec2:*",
      "Effect":"Allow",
      "Resource":"*",
      "Condition":{"Null":{"aws:TokenIssueTime":"false"}}
  }
}
```

Condizioni con più chiavi di contesto o valori

Puoi utilizzare l'elemento `Condition` di una policy per testare più chiavi di contesto o valori per una singola chiave di contesto in una richiesta. Quando effettui una richiesta a AWS, a livello di codice o tramite Console di gestione AWS, la richiesta include informazioni sul tuo principale, sull'operazione, sui tag e altro ancora. Puoi utilizzare le chiavi di contesto per testare i valori delle chiavi di contesto corrispondenti nella richiesta, con le chiavi di contesto specificate nella condizione della policy. Per ulteriori informazioni e i dati inclusi in una richiesta, consulta Il contesto della richiesta.

Argomenti

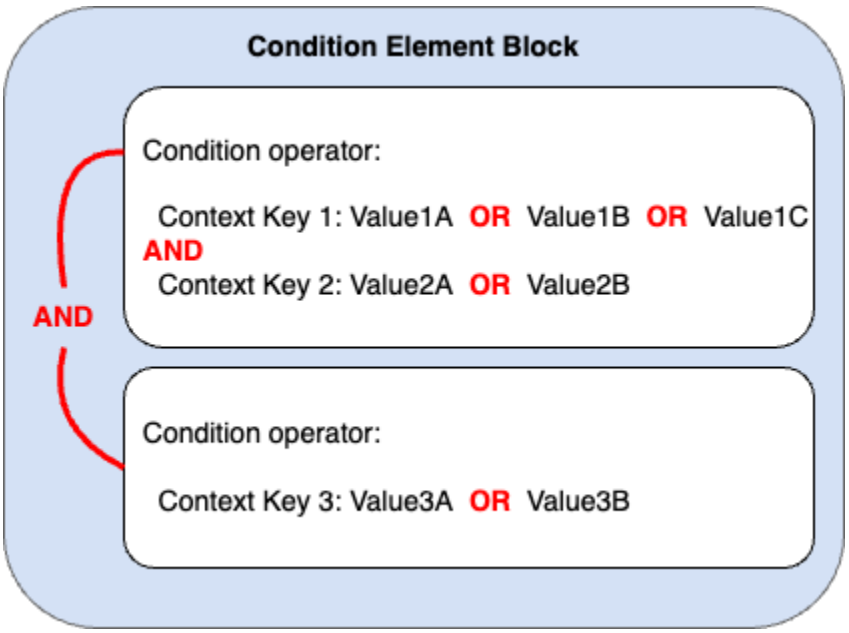
- Logica di valutazione per condizioni con più chiavi di contesto o valori
- Logica di valutazione per gli operatori di negazione della condizione di corrispondenza

Logica di valutazione per condizioni con più chiavi di contesto o valori

Un elemento `Condition` può contenere più operatori di condizione e ciascun operatore di condizione può includere a sua volta più coppie chiave-valore. La maggior parte delle chiavi di contesto supporta l'utilizzo di più valori, se non diversamente specificato.

- Se la tua policy contiene più operatori di condizione, questi vengono valutati utilizzando un `AND` logico.
- Se la policy contiene più chiavi di contesto collegate a un singolo operatore di condizione, le chiavi di contesto vengono valutate utilizzando un `AND` logico.
- Se un singolo operatore di condizione include valori multipli per una chiave di contesto, questi valori sono valutato con un `OR` logico.
- Se un singolo operatore di condizione di corrispondenza negata include valori multipli per una chiave di contesto, questi valori vengono valutati con un `NOR` logico.

Tutte le chiavi di contesto in un blocco di elementi condizionali devono essere risolte in true per richiamare il `Allow` desiderato o l'effetto `Deny`. La figura seguente illustra la logica di valutazione per una condizione con più operatori di condizione e coppie chiave-valore di contesto.



Ad esempio, la seguente policy del bucket S3 illustra come la figura precedente è rappresentata in una policy. Il blocco condizione utilizza operatori di condizione `StringEquals` e `ArnLike` e chiavi di contesto `aws:PrincipalTag` e`aws:PrincipalArn`. Per richiamare l'effetto `Allow` o `Deny` desiderato, tutte le chiavi di contesto nel blocco di condizione devono restituire il valore true. L'utente che effettua la richiesta deve avere entrambe le chiavi tag principali, dipartimento e ruolo, che includono uno dei valori chiave dei tag specificati nella policy. Inoltre, l'ARN principale dell'utente che effettua la richiesta deve corrispondere a uno dei valori `aws:PrincipalArn` specificati nella policy da valutare come true.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222222222222:root"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/department": [
```

```
            "finance",
            "hr",
            "legal"
          ],
          "aws:PrincipalTag/role": [
            "audit",
            "security"
          ]
        },
        "ArnLike": {
          "aws:PrincipalArn": [
            "arn:aws:iam::222222222222:user/Ana",
            "arn:aws:iam::222222222222:user/Mary"
          ]
        }
      }
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"StringEquals": { "aws:PrincipalTag/department": [ "finance", "hr", "legal" ], "aws:PrincipalTag/role": [ "audit", "security" ] }, "ArnLike": { "aws:PrincipalArn": [`	`aws:PrincipalTag/department: legal aws:PrincipalTag/role: audit aws:PrincipalArn: arn:aws:iam::222222222222:user/Mary`	Partita

Condizione della policy	Contesto della richiesta	Risultato

`"arn:aws:iam::222222222222:user/Ana",`
`"arn:aws:iam::222222222222:user/Mary"`
`]`
`}`		

`"StringEquals": {`
`"aws:PrincipalTag/department": [`
`"finance",`
`"hr",`
`"legal"`
`],`
`"aws:PrincipalTag/role": [`
`"audit",`
`"security"`
`]`
`},`
`"ArnLike": {`
`"aws:PrincipalArn": [`
`"arn:aws:iam::222222222222:user/Ana",`
`"arn:aws:iam::222222222222:user/Mary"`
`]`
`}`
	`aws:PrincipalTag/department: hr`
	`aws:PrincipalTag/role: audit`
	`aws:PrincipalArn: arn:aws:iam::222222222222:user/ Nikki`
		Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`"StringEquals": { "aws:PrincipalTag/department": [ "finance", "hr", "legal" ], "aws:PrincipalTag/role": [ "audit", "security" ] }, "ArnLike": { "aws:PrincipalArn": [ "arn:aws:iam::222222222222:user/Ana", "arn:aws:iam::222222222222:user/Mary" ] }`	`aws:PrincipalTag/department: hr aws:PrincipalTag/role: payroll aws:PrincipalArn: arn:aws:iam::22222222222222:user/Mary`	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato

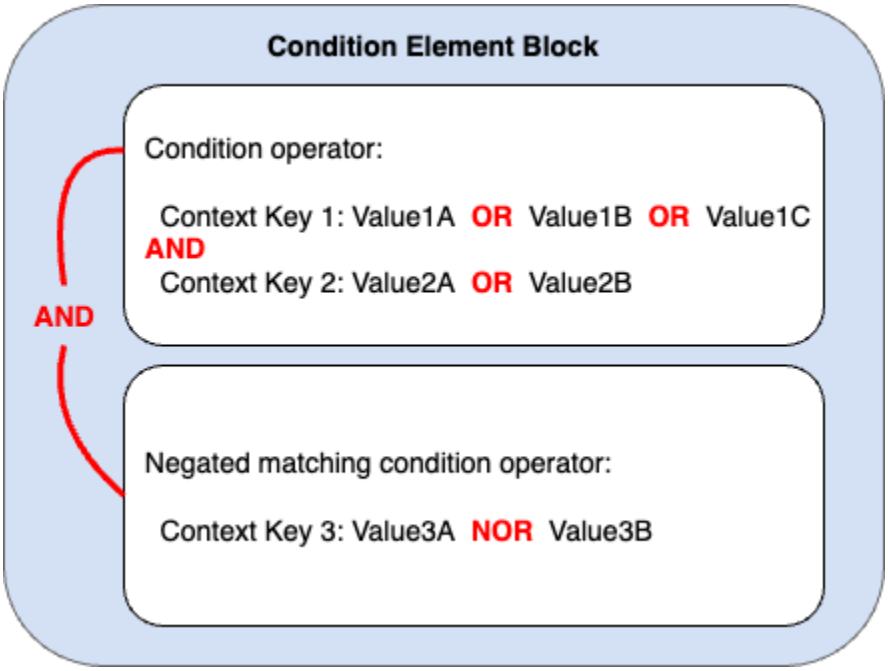
`"StringEquals": { "aws:PrincipalTag/department": [ "finance", "hr", "legal" ], "aws:PrincipalTag/role": [ "audit", "security" ] }, "ArnLike": { "aws:PrincipalArn": [ "arn:aws:iam::222222222222:user/Ana", "arn:aws:iam::222222222222:user/Mary" ] }`	`aws:PrincipalTag/role` non è presente nel contesto della richiesta.
	`aws:PrincipalTag/department: hr`
	`aws:PrincipalArn: arn:aws:iam::222222222222:user/Mary`
		Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`"StringEquals": { "aws:PrincipalTag/department": [ "finance", "hr", "legal" ], "aws:PrincipalTag/role": [ "audit", "security" ] }, "ArnLike": { "aws:PrincipalArn": [ "arn:aws:iam::222222222222:user/Ana", "arn:aws:iam::222222222222:user/Mary" ] }`	`aws:PrincipalTag` non è presente nel contesto della richiesta. `aws:PrincipalArn: arn:aws:iam::22222222222222:user/Mary`	Nessuna corrispondenza

Logica di valutazione per gli operatori di negazione della condizione di corrispondenza

Alcuni operatori di condizione, ad esempio `StringNotEquals` o `ArnNotLike`, usano la corrispondenza negata per confrontare le coppie chiave-valore di contesto nella tua policy con le coppie chiave-valore di contesto in una richiesta. Quando più valori sono elencati in una singola chiave di contesto in una policy con operatori di negazione della condizione di corrispondenza le autorizzazioni efficaci funzionano come un `NOR` logico. Nella corrispondenza negata, un `NOR` o `NOT OR` restituisce true solo se tutti i valori restituiscono false.

La figura seguente illustra la logica di valutazione per una condizione con più operatori di condizione e coppie chiave-valore di contesto. La figura include un operatore di negazione della condizione di corrispondenza per la chiave di contesto 3.



Ad esempio, la seguente policy del bucket S3 illustra come la figura precedente è rappresentata in una policy. Il blocco condizione utilizza operatori di condizione `StringEquals` e `ArnNotLike` e chiavi di contesto `aws:PrincipalTag` e`aws:PrincipalArn`. Per richiamare l'effetto `Allow` o `Deny` desiderato, tutte le chiavi di contesto nel blocco di condizione devono restituire il valore true. L'utente che effettua la richiesta deve avere entrambe le chiavi tag principali, dipartimento e ruolo, che includono uno dei valori chiave dei tag specificati nella policy. Poiché l'operatore di condizione `ArnNotLike` utilizza la corrispondenza negata, l'ARN principale dell'utente che effettua la richiesta deve corrispondere a uno dei valori `aws:PrincipalArn` specificati nella policy da valutare come true.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222222222222:root"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
```

```
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/department": [
            "finance",
            "hr",
            "legal"
          ],
          "aws:PrincipalTag/role": [
            "audit",
            "security"
          ]
        },
        "ArnNotLike": {
          "aws:PrincipalArn": [
            "arn:aws:iam::222222222222:user/Ana",
            "arn:aws:iam::222222222222:user/Mary"
          ]
        }
      }
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"StringEquals": { "aws:PrincipalTag/department": [ "finance", "hr", "legal" ], "aws:PrincipalTag/role": [ "audit", "security" ] },`	`aws:PrincipalTag/department: legal aws:PrincipalTag/role: audit aws:PrincipalArn: arn:aws:iam::222222222222:user/Nikki`	Partita

Condizione della policy	Contesto della richiesta	Risultato
`"ArnNotLike": { "aws:PrincipalArn": [ "arn:aws:iam::222222222222:user/Ana", "arn:aws:iam::222222222222:user/Mary" ] }`		
`"StringEquals": { "aws:PrincipalTag/department": [ "finance", "hr", "legal" ], "aws:PrincipalTag/role": [ "audit", "security" ] }, "ArnNotLike": { "aws:PrincipalArn": [ "arn:aws:iam::222222222222:user/Ana", "arn:aws:iam::222222222222:user/Mary" ] }`	`aws:PrincipalTag/department: hr aws:PrincipalTag/role: audit aws:PrincipalArn: arn:aws:iam::222222222222:user/ Mary`	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
"StringEquals": { "aws:PrincipalTag/department": ["finance", "hr", "legal"], "aws:PrincipalTag/role": ["audit", "security"] }, "ArnNotLike": { "aws:PrincipalArn": ["arn:aws:iam::222222222222:user/Ana", "arn:aws:iam::222222222222:user/Mary"] }	aws:PrincipalTag/department: hr aws:PrincipalTag/role: *payroll* aws:PrincipalArn: arn:aws:iam::222222222222:user/Nikki	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`"StringEquals": { "aws:PrincipalTag/department": [ "finance", "hr", "legal" ], "aws:PrincipalTag/role": [ "audit", "security" ] }, "ArnNotLike": { "aws:PrincipalArn": [ "arn:aws:iam::222222222222:user/Ana", "arn:aws:iam::222222222222:user/Mary" ] }`	> `aws:PrincipalTag/role` non è presente nel contesto della richiesta. `aws:PrincipalTag/department: hr aws:PrincipalArn: arn:aws:iam::22222222222222:user/Nikki`	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`"StringEquals": { "aws:PrincipalTag/department": [ "finance", "hr", "legal" ], "aws:PrincipalTag/role": [ "audit", "security" ] }, "ArnNotLike": { "aws:PrincipalArn": [ "arn:aws:iam::222222222222:user/Ana", "arn:aws:iam::222222222222:user/Mary" ] }`	`aws:PrincipalTag` non è presente nel contesto della richiesta. `aws:PrincipalArn: arn:aws:iam::22222222222222:user/Nikki`	Nessuna corrispondenza

Single-valued rispetto alle chiavi di contesto multivalore

La differenza tra le chiavi di contesto a valore singolo e multivalore dipende dal numero di valori nel contesto della richiesta e non dal numero di valori nella condizione della policy.

- Single-valuedle chiavi del contesto di condizione hanno al massimo un valore nel contesto della richiesta. Ad esempio, quando si etichettano le risorse in AWS, ogni tag di risorsa viene archiviato come coppia chiave-valore. Poiché una chiave di tag di risorsa può avere un solo valore di tag, the section called "ResourceTag" è una chiave di contesto a valore singolo. Non utilizzare operatori con una chiave di contesto a valore singolo.
- Le chiavi di contesto con condizione multivalore possono avere più di un valore nel contesto della richiesta. Ad esempio, quando tagghi le risorse in AWS, puoi includere più coppie di tag

chiave-valore in un'unica richiesta. Pertanto, the section called "TagKeys" è una chiave di contesto multivalore. Le chiavi di contesto multivalore richiedono un operatore di condizione.

Ad esempio, una richiesta può provenire al massimo da un endpoint VPC, quindi the section called "SourceVpce" è una chiave di contesto a valore singolo. Poiché un servizio può avere più di un nome del principale di servizio appartenente al servizio, the section called "PrincipalServiceNamesList" è una chiave di contesto multivalore.

> **Important**
>
> La differenza tra le chiavi di contesto della condizione a valore singolo e multivalore dipende dal numero di valori nel contesto della richiesta e non dal numero di valori nella condizione della policy.

Per esempi di policy che dimostrano condizioni con più chiavi e valori di contesto, consulta. Esempi di policy delle condizioni

Punti chiave

- Le classificazioni Single-valued e multivalore sono incluse nella descrizione di ciascuna chiave di contesto della condizione come Tipo di valore nell'argomento. AWS chiavi di contesto delle condizioni globali
- Le chiavi di contesto multivalore nella documentazione di riferimento per l'autorizzazione del servizio utilizzano un prefisso `ArrayOf` seguito dal tipo di categoria dell'operatore di condizione, come `ArrayOfString` o`ArrayOfARN`, che indica che la richiesta può includere più valori per una chiave di contesto della condizione.
- Puoi utilizzare qualsiasi chiave di contesto a valore singolo disponibile come variabile di policy, ma non è possibile utilizzare una chiave di contesto multivalore come variabile di policy. Per ulteriori informazioni sulle variabili di policy, consultare Elementi delle policy IAM: variabili e tag.
- Quando si utilizzano chiavi di contesto che includono coppie chiave-valore, è importante notare che anche se possono esserci più valori tag-chiave, ogni `tag-key` può avere un solo valore.
  - the section called "PrincipalTag", the section called "RequestTag" e the section called "ResourceTag" sono chiavi di contesto a valore singolo.
  - the section called "TagKeys" definisce quali tag-chiavi sono consentiti in una richiesta ma non include i valori tag-chiave. Poiché è possibile includere più coppie chiave-valore dei tag in una richiesta, `aws:TagKeys` è una chiave di contesto multivalore.

- Le chiavi di contesto multivalore richiedono un operatore di condizione. Non utilizzare operatori di condizione `ForAllValues` o `ForAnyValue` chiavi di contesto a valore singolo. L'utilizzo di operatori di set di condizioni con chiavi di contesto a valore singolo può portare a policy eccessivamente permissive.

Operatori dell'insieme per le chiavi di contesto multivalore

Per confrontare la chiave di contesto della condizione con una chiave di contesto di richiesta con più valori chiave, devi utilizzare gli operatori di insiemi `ForAllValues` o `ForAnyValue`. Questi operatori di insieme sono usati per paragonare due insiemi di valori, ad esempio il set di tag in una richiesta e il set di tag in una condizione della policy.

I qualificatori `ForAllValues` e `ForAnyValue` aggiungono funzionalità di operazione di insieme all'operatore di condizione, in modo che tu possa testare chiavi di contesto della richiesta multivalore con più chiavi di contesto multiple in una condizione della policy. Inoltre, se includi una chiave di contesto di stringa multivalore nella policy con un carattere jolly o una variabile, devi utilizzare anche l'operatore di condizione `StringLike`. I valori multipli delle chiavi di condizione devono essere racchiusi tra parentesi quadre come in un array, ad esempio `"Key2":["Value2A", "Value2B"]`.

ForAllValues

Il qualificatore `ForAllValues` verifica se il valore di ogni membro del contesto della richiesta è corrispondente all'operatore di condizione che segue il qualificatore. La condizione restituisce `true` se ogni valore della chiave di contesto nella richiesta corrisponde a un valore della chiave di contesto nella policy. Restituisce `true` anche se non ci sono chiavi di contesto nella richiesta.

> **Important**
>
> Fai attenzione se usi `ForAllValues` con un effetto `Allow` perché ciò può essere eccessivamente permissivo se la presenza di chiavi di contesto mancanti nel contesto della richiesta è imprevista. Dovresti sempre includere l'operatore di condizione `Null` nella tua policy con un valore `false` per verificare se la chiave di contesto esiste e il suo valore non è Null. Per vedere un esempio, consulta Controllo dell'accesso in base alle chiavi di tag.

Esempio di operatore di set ForAllValues

Nel seguente esempio, ForAllValues viene utilizzato con aws: TagKeys per consentire agli utenti di eliminare tag specifici assegnati a un'istanza EC2. Questa policy consente agli utenti di eliminare

solo i tag `environment` e `cost-center`. Puoi eliminarli separatamente o insieme. I tag-chiavi nella richiesta devono corrispondere esattamente alle chiavi specificate nella policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteTags",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "environment",
                        "cost-center"
                    ]
                },
                "Null": {
                    "aws:TagKeys": "false"
                }
            }
        }
    ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"ForAllValues:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }, "Null": {`	`aws:TagKeys: - environment`	Corrisponde

Condizione della policy	Contesto della richiesta	Risultato
`"aws:TagKeys": "false" }`		
`"ForAllValues:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }, "Null": { "aws:TagKeys": "false" }`	`aws:TagKeys: - cost-center`	Incontro
`"ForAllValues:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }, "Null": { "aws:TagKeys": "false" }`	`aws:TagKeys: - environment - cost-center`	Incontro

Condizione della policy	Contesto della richiesta	Risultato
`"ForAllValues:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }, "Null": { "aws:TagKeys": "false" }`	`aws:TagKeys: - environment - dept`	Nessuna corrispondenza
`"ForAllValues:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }, "Null": { "aws:TagKeys": "false" }`	`aws:TagKeys` non è presente nel contesto della richiesta.	Nessuna corrispondenza

Nota che nell'ultimo esempio, il risultato è "No Match" perché il controllo della condizione Null impedisce la corrispondenza quando manca la chiave di contesto. Questa è una best practice per evitare policy eccessivamente permissive.

ForAnyValue

Il qualificatore `ForAnyValue` verifica se almeno un membro del set di valori di richiesta è corrispondente ad almeno un membro del set di valori delle chiavi di condizione della policy. La condizione restituisce `true` se uno qualsiasi dei valori della chiave di contesto nella richiesta corrisponde a un valore qualsiasi della chiave di contesto nella policy. Se non esiste alcuna chiave di contesto corrispondente o se la chiave non esiste, la condizione restituisce `false`.

> **Important**
>
> Quando si usa `ForAnyValue` con un effetto `Deny`, se la chiave di contesto non è presente nella richiesta, la policy viene valutata come Nessuna corrispondenza. Per un comportamento coerente, aggiungi un controllo esplicito `Null` delle condizioni nella tua policy per verificare se la chiave di contesto esiste. Per informazioni dettagliate, vedi Operatore di condizione per verificare la presenza di chiavi di condizione.

Esempio ForAnyValue di operatore di set

Nel seguente esempio, ForAnyValue viene utilizzato con aws: TagKeys per consentire agli utenti di eliminare tag specifici assegnati a un'istanza EC2. Questa policy consente agli utenti di eliminare i tag di un'istanza se le chiavi di tag specificate nella richiesta includono `environment` o `cost-center`. La richiesta può includere chiavi di tag aggiuntive oltre a quelle specificate nella policy, ma deve includere almeno una delle chiavi specificate per soddisfare la condizione.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteTags",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": [
                        "environment",
                        "cost-center"
                    ]
                }
            }
        }
    ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"ForAnyValue:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }`	`aws:TagKeys: - environment`	Corrisponde
`"ForAnyValue:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }`	`aws:TagKeys: - cost-center`	Incontro
`"ForAnyValue:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }`	`aws:TagKeys: - environment - cost-center`	Incontro
`"ForAnyValue:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }`	`aws:TagKeys: - environment - dept`	Incontro
`"ForAnyValue:StringEquals": {`	`aws:TagKeys:`	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato
`"aws:TagKeys": [ "environment", "cost-center" ] }`	`- dept`	
`"ForAnyValue:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }`	`aws:TagKeys` non è presente nel contesto della richiesta.	Nessuna corrispondenza

Esempi di policy delle condizioni

Nelle policy IAM, puoi specificare più valori per chiavi di contesto sia a valore singolo che multivalore per il confronto con il contesto della richiesta. La seguente serie di esempi di policy mostra le condizioni delle policy con più chiavi e valori di contesto.

> **Note**
>
> Per inviare una policy e includerla in questa guida di riferimento, utilizza il pulsante Feedback in fondo a questa pagina. Per esempi di policy basate su identità IAM, consulta Esempi di policy basate su identità IAM.

Esempi di policy sulle condizioni: chiavi di Single-valued contesto

- Blocchi di condizioni multipli con chiavi di contesto a valore singolo
- Un blocco di condizioni con più chiavi e valori di contesto a valore singolo

Esempi di policy relativi alle condizioni: chiavi di contesto multivalore

- Politica di negazione con operatore di set di condizioni ForAllValues
- Politica di rifiuto con operatore di set di condizioni ForAnyValue

Esempi chiave di contesto multivalore

La seguente serie di esempi di policy mostra come creare condizioni politicy con chiavi di contesto multivalore.

Esempio: politica di rifiuto con operatore di set di condizioni ForAllValues

Gli esempi seguenti mostrano come utilizzare una policy basata sull'identità per negare l'uso di operazioni di tagging IAM quando nella richiesta sono inclusi prefissi specifici della chiave di tag. I valori per `aws:TagKeys` includono un carattere jolly (*) per la corrispondenza parziale delle stringhe. La policy include l'operatore dell'insieme `ForAllValues` con la chiave di contesto `aws:TagKeys` perché la chiave di contesto della richiesta può includere più valori. Affinché la chiave di contesto `aws:TagKeys` corrisponda, ogni valore nel contesto della richiesta deve corrispondere ad almeno un valore presente nella policy.

L'operatore dell'insieme `ForAllValues` restituisce True anche se non ci sono chiavi di contesto nella richiesta.

Puoi evitare che le chiavi di contesto mancanti o le chiavi di contesto con valori vuoti vengano valutate come True includendo un operatore di condizione `Null` nella tua policy con un valore `false` per verificare se la chiave di contesto nella richiesta esiste e il suo valore non è Null. Per ulteriori informazioni, consulta Operatore di condizione per verificare la presenza di chiavi di condizione.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

Example Negazione di un singolo valore di condizione della policy per una chiave di contesto multivalore

Nel seguente esempio, la policy nega le richieste in cui i valori di `aws:TagKeys` nella richiesta non includono il prefisso key1. Il contesto della richiesta può avere più valori, ma a causa dell'operatore dell'insieme di condizioni `ForAllValues`, tutti i valori chiave del tag nel contesto della richiesta devono iniziare con il prefisso key1.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyRestrictedTags",
      "Effect": "Deny",
      "Action": [
        "iam:Tag*",
        "iam:UnTag*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "ForAllValues:StringNotLike": {
          "aws:TagKeys": "key1*"
        }
      }
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta. Per un'istruzione Nega, una Corrispondenza è Negata e una Mancata corrispondenza è Non negata, quindi può essere consentita da un'altra istruzione.

Condizione della policy	Contesto della richiesta	Risultato
`"ForAllValues:StringNotLike": { "aws:TagKeys": "key1*" }`	`aws:TagKeys: - key1:legal`	Nessuna corrispondenza Può essere consentito da un'altra istruzione.
`"ForAllValues:StringNotLike": { "aws:TagKeys": "key1*" }`	`aws:TagKeys: - key1:hr - key1:personnel`	Nessuna corrispondenza Può essere consentito da un'altra istruzione.

Condizione della policy	Contesto della richiesta	Risultato
`"ForAllValues:StringNotLike": { "aws:TagKeys": "key1*" }`	`aws:TagKeys: - key2:audit`	Partita
`"ForAllValues:StringNotLike": { "aws:TagKeys": "key1*" }`	`aws:TagKeys` non è presente nel contesto della richiesta.	Incontro

Example Negazione di più valori di condizione della policy per una chiave di contesto multivalore

Nell'esempio seguente, la policy nega le richieste in cui i valori `aws:TagKeys` della richiesta non includono il prefisso key1 o key2 Il contesto della richiesta può avere più valori, ma a causa dell'operatore dell'insieme di condizioni `ForAllValues`, tutti i valori chiave del tag nel contesto della richiesta devono iniziare con il prefisso key1 o key2.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyRestrictedTags",
      "Effect": "Deny",
      "Action": [
        "iam:Tag*",
        "iam:UnTag*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "ForAllValues:StringNotLike": {
          "aws:TagKeys": [
```

```
            "key1*",
            "key2*"
          ]
        }
      }
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta. Per un'istruzione Nega, una Corrispondenza è Negata e una Mancata corrispondenza è Non negata, quindi può essere consentita da un'altra istruzione.

Condizione della policy	Contesto della richiesta	Risultato
`"ForAllValues:StringNotLike": { "aws:TagKeys": [ "key1*", "key2*" ] }`	`aws:TagKeys: - key1:legal`	Nessuna corrispondenza Può essere consentito da un'altra istruzione.
`"ForAllValues:StringNotLike": { "aws:TagKeys": [ "key1*", "key2*" ] }`	`aws:TagKeys: - key1:hr - key1:personnel`	Nessuna corrispondenza Può essere consentito da un'altra istruzione.
`"ForAllValues:StringNotLike": { "aws:TagKeys": [ "key1*", "key2*" ] }`	`aws:TagKeys: - key1:hr - key2:audit`	Nessuna corrispondenza Può essere consentito da un'altra istruzione.

Condizione della policy	Contesto della richiesta	Risultato
`"ForAllValues:StringNotLike": { "aws:TagKeys": [ "key1*", "key2*" ] }`	`aws:TagKeys: - key3:legal`	Partita
`"ForAllValues:StringNotLike": { "aws:TagKeys": [ "key1*", "key2*" ] }`	`aws:TagKeys` non è presente nel contesto della richiesta.	Incontro

Esempio: politica di rifiuto con operatore di set di condizioni ForAnyValue

Il seguente esempio di policy basata sull'identità nega la creazione di istantanee dei volumi di istanza EC2 se alcune istantanee sono contrassegnate con una delle chiavi di tag specificate nella policy,`environmentowebserver`. La policy include l'operatore dell'insieme `ForAnyValue` con la chiave di contesto `aws:TagKeys` perché la chiave di contesto della richiesta può includere più valori. Se la richiesta di etichettatura include uno dei valori chiave dei tag specificati nella policy, la `aws:TagKeys` chiave di contesto restituisce true richiamando l'effetto della policy di negazione.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ec2:CreateSnapshot",
        "ec2:CreateSnapshots"
      ],
      "Resource": "arn:aws:ec2:us-west-2::snapshot/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": "webserver"
        }
      }
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta. Per un'istruzione Nega, una Corrispondenza è Negata e una Mancata corrispondenza è Non negata, quindi può essere consentita da un'altra istruzione.

Condizione della policy	Contesto della richiesta	Risultato
`"ForAnyValue:StringEquals": { "aws:TagKeys": "webserver" }`	`aws:TagKeys: - webserver`	Partita
`"ForAnyValue:StringEquals": { "aws:TagKeys": "webserver" }`	`aws:TagKeys: - environment - webserver - test`	Incontro
`"ForAnyValue:StringEquals": { "aws:TagKeys": "webserver"`	`aws:TagKeys: - environment - test`	Nessuna corrispondenza Può essere consentito da un'altra istruzione.

Condizione della policy	Contesto della richiesta	Risultato
`}`		
`"ForAnyValue:StringEquals": { "aws:TagKeys": "webserver" }`	`aws:TagKeys` non è presente nel contesto della richiesta.	Nessuna corrispondenza Può essere consentito da un'altra istruzione.

Single-valued esempi di politiche chiave di contesto

La seguente serie di esempi di policy mostra come creare condizioni nella policy con chiavi di contesto a valore singolo.

Esempio: più blocchi di condizioni con chiavi di contesto a valore singolo

Quando un blocco di condizioni contiene più condizioni, ognuna con una singola chiave di contesto, tutte le chiavi di contesto devono risolversi in true per l'effetto `Allow` o `Deny` che si desidera richiamare. Quando si utilizzano operatori per la condizione di corrispondenza negata, la logica di valutazione del valore della condizione viene invertita.

Il seguente esempio consente agli utenti di creare volumi EC2 e applicare tag a tali volumi durante la creazione del volume. Il contesto della richiesta deve includere un valore per la chiave di contesto `aws:RequestTag/project` e il valore della chiave di contesto `aws:ResourceTag/environment` può essere qualsiasi cosa tranne la produzione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVolume",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
```

```
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
      "Condition": {
        "StringLike": {
          "aws:RequestTag/project": "*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:*/*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceTag/environment": "production"
        }
      }
    }
  ]
}
```

Il contesto della richiesta deve includere un tag-valore del progetto e non può essere creato affinché una risorsa di produzione richiami l'effetto `Allow`. Il seguente volume EC2 è stato creato correttamente perché il nome del progetto è `Feature3` con un tag della risorsa `QA`.

```
aws ec2 create-volume \
    --availability-zone us-east-1a \
    --volume-type gp2 \
    --size 80 \
    --tag-specifications 'ResourceType=volume,Tags=[{Key=project,Value=Feature3},
{Key=environment,Value=QA}]'
```

Esempio: un blocco di condizioni con più chiavi di contesto a valore singolo

Quando un blocco di condizioni contiene più chiavi di contesto e ogni chiave di contesto ha valori multipli, ogni chiave di contesto deve risolversi in true in almeno un valore chiave per l'effetto `Allow` o `Deny` che si desidera richiamare. Quando si utilizzano operatori per la condizione di corrispondenza negata, la logica di valutazione del valore della chiave di contesto viene invertita.

L'esempio seguente consente agli utenti di avviare ed eseguire attività sui cluster Amazon Elastic Container Service.

- Il contesto della richiesta deve includereproduction O prod-backup per la aws:RequestTag/environment chiave di contesto E.
- La chiave di contesto ecs:cluster assicura che le attività vengano eseguite su entrambi i cluster default1 O default2 ARN ECS.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:RunTask",
        "ecs:StartTask"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": [
            "production",
            "prod-backup"
          ]
        },
        "ArnEquals": {
          "ecs:cluster": [
            "arn:aws:ecs:us-east-1:111122223333:cluster/default1",
            "arn:aws:ecs:us-east-1:111122223333:cluster/default2"
          ]
        }
      }
    }
  ]
}
```

La tabella seguente mostra come AWS valuta questa politica in base ai valori della chiave di condizione nella richiesta.

Condizione della policy	Contesto della richiesta	Risultato
`"StringEquals": { "aws:RequestTag/environment": [ "production", "prod-backup" ] }, "ArnEquals": { "ecs:cluster": [ "arn:aws:ecs:us-east-1:111122223333:cluster/default1", "arn:aws:ecs:us-east-1:111122223333:cluster/default2" ] }`	`aws:RequestTag: environment:production ecs:cluster: arn:aws:ecs:us-east-1:111122223333:cluster/default1`	Match
`"StringEquals": { "aws:RequestTag/environment": [ "production", "prod-backup" ] }, "ArnEquals": { "ecs:cluster": [ "arn:aws:ecs:us-east-1:111122223333:cluster/default1", "arn:aws:ecs:us-east-1:111122223333:cluster/default2" ] }`	`aws:RequestTag: environment:prod-backup ecs:cluster: arn:aws:ecs:us-east-1:111122223333:cluster/default2`	Match
`"StringEquals": { "aws:RequestTag/environment": [`	`aws:RequestTag: webserver:production ecs:cluster:`	Nessuna corrispondenza

Condizione della policy	Contesto della richiesta	Risultato

`"production",`
`"prod-backup"`
`]`
`},`
`"ArnEquals": {`
`"ecs:cluster": [`
`"arn:aws:ecs:us-east-1:111122223333:cluster/default1",`
`"arn:aws:ecs:us-east-1:111122223333:cluster/default2"`
`]`
`}`
	`arn:aws:ecs:us-east-1:111122223333:cluster/default2`	

`"StringEquals": {`
`"aws:RequestTag/environment": [`
`"production",`
`"prod-backup"`
`]`
`},`
`"ArnEquals": {`
`"ecs:cluster": [`
`"arn:aws:ecs:us-east-1:111122223333:cluster/default1",`
`"arn:aws:ecs:us-east-1:111122223333:cluster/default2"`
`]`
`}`
	`aws:RequestTag` non è presente nel contesto della richiesta.
	`ecs:cluster`
	`arn:aws:ecs:us-east-1:111122223333:cluster/default2`
		Nessuna corrispondenza

Elementi delle policy IAM: variabili e tag

Utilizza le variabili di policy AWS Identity and Access Management (IAM) come segnaposto quando non conosci il valore esatto di una risorsa o di una chiave di condizione quando scrivi la policy.

> **Note**
>
> Se AWS non è possibile risolvere una variabile, l'intera istruzione potrebbe non essere valida. Ad esempio, se utilizzi la variabile `aws:TokenIssueTime`, questa viene risolta in un valore solo quando il richiedente è stato autenticato tramite le credenziali temporanee (un ruolo IAM). Per evitare che le variabili causino dichiarazioni non valide, usa... IfExists operatore di condizione.

Argomenti

- Introduzione
- Utilizzo delle variabili nelle policy
- Tag come variabili di policy
- Casi in cui è possibile utilizzare le variabili di policy
- Variabili di policy senza valore
- Richiesta di informazioni utilizzabili per le variabili di policy
- Specifica dei valori di default
- Ulteriori informazioni

Introduzione

Nelle policy IAM, numerose operazioni consentono di assegnare un nome a risorse specifiche per le quali si desidera controllare l'accesso. Ad esempio, la policy di seguito consente all'utente di elencare, leggere e scrivere gli oggetti nel bucket S3 `amzn-s3-demo-bucket` per i progetti `marketing`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket"],
```

```
      "Condition": {"StringLike": {"s3:prefix": ["marketing/*"]}}
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/marketing/*"]
    }
  ]
}
```

In alcuni casi, potresti non conoscere il nome esatto della risorsa al momento di scrivere la policy. Puoi generalizzare la policy per renderla utilizzabile da molti utenti senza bisogno di effettuare una copia univoca per ciascun utente. Invece di creare una policy separata per ciascun utente, consigliamo di creare una singola policy di gruppo che funzioni per tutti gli utenti che appartengono a tale gruppo.

Utilizzo delle variabili nelle policy

È possibile definire valori dinamici all'interno delle policy utilizzando variabili di policy che impostano i segnaposti in una policy.

Le variabili sono contrassegnate utilizzando un prefisso **$**seguito da una coppia di parentesi graffe (**{ }**) che includono il nome della variabile del valore della richiesta.

Quando la policy viene valutata, le variabili di policy vengono sostituite con valori provenienti dalle chiavi di contesto condizionali inoltrate nella richiesta. Le variabili possono essere utilizzate nelle politiche basate sull'identità, nelle politiche delle risorse, nelle politiche di controllo dei servizi, nelle politiche di sessione e nelle politiche degli endpoint VPC. Identity-based le politiche utilizzate come limiti delle autorizzazioni supportano anche le variabili di policy.

Le chiavi di contesto delle condizioni globali possono essere utilizzate come variabili nelle richieste tra AWS i servizi. Anche le chiavi di condizione specifiche del servizio possono essere utilizzate come variabili quando interagiscono con le risorse AWS , ma sono disponibili soltanto quando le richieste vengono effettuate su risorse che le supportano. Per un elenco delle chiavi di contesto disponibili per ogni AWS servizio e risorsa, consulta il Service Authorization Reference . In determinate circostanze, non è possibile inserire un valore nelle chiavi di contesto delle condizioni globali. Per ulteriori informazioni su ciascuna chiave, consulta AWS global condition context keys.

Important

- I nomi delle chiavi non fanno distinzione tra maiuscole e minuscole. Ad esempio, `aws:CurrentTime` è uguale a `AWS:currenttime`.
- È possibile utilizzare qualsiasi chiave di condizione a valore singolo come variabile. Non è possibile utilizzare una chiave della condizione multi-valore come variabile.

L'esempio seguente mostra una policy per un utente o un ruolo IAM che sostituisce il nome di una risorsa specifica con una variabile di policy. Puoi riutilizzare questa policy sfruttando i vantaggi della chiave di condizione `aws:PrincipalTag`. Quando questa policy viene valutata, `${aws:PrincipalTag/team}` consente le operazioni solo se il nome del bucket termina con il nome del team dal tag del principale `team`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket"],
      "Condition": {"StringLike": {"s3:prefix": ["${aws:PrincipalTag/team}/*"]}}
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/${aws:PrincipalTag/team}/*"]
    }
  ]
}
```

La variabile viene contrassegnata utilizzando un prefisso $ seguito da una coppia di parentesi graffe ({ }). All'interno dei caratteri ${ } è possibile includere il nome del valore ricavato dalla richiesta da utilizzare nella policy. I valori che possono essere utilizzati sono descritti più avanti in questa pagina.

Per maggiori dettagli su questa chiave di condizione globale, consulta aws:PrincipalTag/tag-key nell'elenco di chiavi di condizioni globali.

> **Note**
>
> Per usare le variabili di policy, è necessario che l'elemento `Version` sia incluso in una dichiarazione. Inoltre, la versione deve essere impostata su una versione che supporti le variabili di policy. Le variabili sono state introdotte a partire dalla versione `2012-10-17`. Le versioni precedenti del linguaggio di policy non supportano le variabili. Se l'elemento `Version` non viene incluso e impostato su una data appropriata, alcune variabili, come ad esempio `${aws:username}`, saranno trattate come stringhe letterali nella policy.
> Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Diversamente, viene creata una versione della policy quando si modifica una policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per ulteriori informazioni sull'elemento di policy `Version`, consultare the section called "Version". Per ulteriori informazioni sulle versioni di policy, consultare the section called "Controllo delle versioni delle policy IAM".

Una policy che consente a un principale di ottenere oggetti dal percorso /David di un bucket S3 è simile alla seguente:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
```

```
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/David/*"
      ]
    }
  ]
}
```

Se questa policy è collegata all'utente `David`, tale utente ottiene gli oggetti dal proprio bucket S3, ma potrebbe essere necessario creare una policy separata per ogni utente che include il nome dell'utente. Ciascuna policy dovrà quindi essere collegata ai singoli utenti.

Utilizzando una variabile di policy, è possibile creare policy che possono essere riutilizzate. La seguente policy consente a un utente di ottenere oggetti da un bucket Amazon S3 se il valore tag-chiave per `aws:PrincipalTag` corrisponde al valore del `owner` tag-chiave inviato nella richiesta.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Sid": "AllowUnlessOwnedBySomeoneElse",
    "Effect": "Allow",
    "Action": ["s3:GetObject"],
    "Resource": ["*"],
    "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/owner": "${aws:PrincipalTag/owner}"
        }
      }
    }
  ]
}
```

Se utilizzi una variabile di policy al posto di un utente di questo tipo, non è necessaria una policy separata per ogni singolo utente. Nell'esempio seguente, la policy è collegata a un ruolo IAM assunto dai Product Manager tramite credenziali di sicurezza temporanee. Quando un utente richiede di aggiungere un oggetto Amazon S3, IAM sostituisce il valore del tag `dept` della richiesta corrente per la variabile `${aws:PrincipalTag}` e valuta la policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOnlyDeptS3Prefix",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/${aws:PrincipalTag/dept}/*"
            ]
        }
    ]
}
```

Tag come variabili di policy

In alcuni AWS servizi è possibile associare attributi personalizzati alle risorse create da tali servizi. Ad esempio, puoi applicare tag a bucket Amazon S3 o a utenti IAM. Questi tag sono coppie chiave-valore. È possibile definire il nome della chiave di tag e il valore associato al nome della chiave. Ad esempio, puoi creare un tag con una chiave **`department`** e un valore **`Human Resources`**. Per ulteriori informazioni sul tagging delle entità IAM, consulta Tag per AWS Identity and Access Management risorse. Per informazioni sul tagging delle risorse create da altri servizi AWS , consulta la documentazione di tali servizi. Per ulteriori informazioni sull'utilizzo dell'editor di tag, consulta l'articolo relativo all'utilizzo dell'editor di tag nella Guida per l'utente della Console di gestione AWS .

Puoi applicare tag alle risorse IAM per semplificare il rilevamento, l'organizzazione e il monitoraggio delle risorse IAM. Puoi inoltre applicare tag alle identità IAM per controllare l'accesso alle risorse o al tagging. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Casi in cui è possibile utilizzare le variabili di policy

Le variabili di policy possono essere utilizzate nell'elemento `Resource` e per confrontare stringhe nell'elemento `Condition`.

Elemento risorsa

È possibile utilizzare una variabile criterio nell'elemento `Resource`, ma solo nella parte risorsa dell'ARN. Questa parte dell'ARN appare dopo i quinti due punti (:). Non è possibile utilizzare una variabile per sostituire parti dell'ARN prima dei quinti due punti, ad esempio il servizio o l'account. Per ulteriori informazioni sul formato ARN, consulta ARN IAM.

Per sostituire una parte di un ARN con un valore di tag, inserisci il prefisso e il nome della chiave in `${ }`. Ad esempio, il seguente elemento Risorsa si riferisce solo a un bucket con lo stesso nome del valore nel tag department dell'utente richiedente.

```
"Resource": ["arn:aws::s3:::amzn-s3-demo-bucket/
${aws:PrincipalTag/department}"]
```

Molte AWS risorse utilizzano ARN che contengono un nome creato dall'utente. La seguente policy IAM garantisce che solo gli utenti voluti con i valori di tag access-project, access-application e access-environment corrispondenti possono modificare le relative risorse. Inoltre, utilizzando le corrispondenze con carattere jolly *, sono in grado di consentire suffissi personalizzati per i nomi delle risorse.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessBasedOnArnMatching",
      "Effect": "Allow",
      "Action": [
        "sns:CreateTopic",
        "sns:DeleteTopic"],
      "Resource": ["arn:aws:sns:*:*:${aws:PrincipalTag/access-project}-
${aws:PrincipalTag/access-application}-${aws:PrincipalTag/access-environment}-*"
      ]
    }
  ]
}
```

Elemento condizione

È possibile utilizzare una variabile di criterio per i valori `Condition` in qualsiasi condizione che coinvolga gli operatori stringa o gli operatori ARN. Gli operatori stringa includono `StringEquals`, `StringLike` e `StringNotLike`. Gli operatori ARN includono `ArnEquals` e `ArnLike`. Non è possibile utilizzare una variabile di criterio con altri operatori, ad esempio `Numeric`, `Date`, `Boolean`, `Binary`, `IP Address` o `Null`. Per ulteriori informazioni sugli operatori delle condizioni, vedere Elementi della policy JSON IAM: operatori di condizione.

Quando fai riferimento a un tag in un'espressione dell'elemento `Condition`, utilizza il prefisso e il nome della chiave pertinenti come chiave di condizione. Quindi utilizza il valore che desideri testare nel valore della condizione.

Ad esempio, la seguente policy di esempio consente l'accesso completo agli utenti; ma solo se il tag `costCenter` è collegato all'utente. Il tag deve inotlre avere un valore pari a `12345` o `67890`. Se il tag non ha nessun valore o ha qualsiasi altro valore, la richiesta ha esito negativo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iam:*user*"
       ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:ResourceTag/costCenter": [ "12345", "67890" ]
        }
      }
    }
  ]
}
```

Variabili di policy senza valore

Quando le variabili di policy fanno riferimento a una chiave di contesto delle condizioni che non ha valore o non è presente nel contesto di autorizzazione per una richiesta, il valore è effettivamente nullo. Non esiste un valore uguale o simile. Le chiavi di contesto delle condizioni potrebbero non essere presenti nel contesto di autorizzazione quando:

- Si utilizzano le chiavi di contesto delle condizioni specifiche del servizio nelle richieste inviate a risorse che non supportano tale chiave di condizione.
- I tag sulle sessioni, sulle risorse, sulle richieste o sui principali IAM non sono presenti.
- Altre circostanze, come specificate per ogni chiave di contesto delle condizioni globali alla pagina AWS chiavi di contesto delle condizioni globali.

Quando si utilizza una variabile senza valore nell'elemento della condizione di una policy IAM, gli Elementi della policy JSON IAM: operatori di condizione come `StringEquals` o `StringLike` non corrispondono e l'istruzione della policy non ha effetto.

Gli operatori di condizione invertiti come `StringNotEquals` o `StringNotLike` corrispondono a un valore nullo, poiché il valore della chiave di condizione che stanno verificando non corrisponde o non è uguale al valore effettivamente nullo.

Nel seguente esempio, per consentire l'accesso `aws:principaltag/Team` deve essere uguale a `s3:ExistingObjectTag/Team`. L'accesso è esplicitamente negato quando `aws:principaltag/Team` non è impostato. Se una variabile che non ha valore nel contesto di autorizzazione viene utilizzata come parte dell'elemento `Resource` o `NotResource` di una policy, la risorsa che include una variabile di policy senza valore non corrisponderà ad alcuna risorsa.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
   {
    "Effect": "Deny",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
      "StringNotEquals": {
```

```
          "s3:ExistingObjectTag/Team": "${aws:PrincipalTag/Team}"
        }
      }
    }
  ]
}
```

Richiesta di informazioni utilizzabili per le variabili di policy

È possibile utilizzare l'elemento `Condition` di una policy JSON per confrontare le chiavi nel contesto della richiesta con i valori chiave specificati nella policy. Quando si utilizza una variabile di policy, AWS sostituisce un valore della chiave del contesto della richiesta al posto della variabile nella policy.

Valori della chiave dell'entità principale

I valori per `aws:username`, `aws:userid` e `aws:PrincipalType` dipendono dal tipo di principale che ha avviato la richiesta. Ad esempio, la richiesta può essere effettuata utilizzando le credenziali di un utente IAM, di un ruolo IAM o dell' Utente root dell'account AWS. La seguente lista mostra i valori di tali chiavi per i diversi tipi di principale.

- Utente root dell'account AWS
  - `aws:username`: (non presente)
  - `aws:userid`: ID Account AWS
  - `aws:PrincipalType`: `Account`
- Utente IAM
  - `aws:username`: *`IAM-user-name`*
  - `aws:userid`: ID univoco
  - `aws:PrincipalType`: `User`
- AWS STS utente principale federato
  - `aws:username`: (non presente)
  - `aws:userid`: *`account`*:*`caller-specified-name`*
  - `aws:PrincipalType`: `FederatedUser`
- Principali federati OIDC e SAML

> **Note**
>
> Per informazioni sulle chiavi di policy disponibili quando si usa la federazione OIDC, consulta Federazione OIDC.

  - `aws:username`: (non presente)
  - `aws:userid`: (non presente)
  - `aws:PrincipalType`: `AssumedRole`
- Ruolo presunto
  - `aws:username`: (non presente)
  - `aws:userid`: *`role-id`*:*`caller-specified-role-name`*
  - `aws:PrincipalType`: `Assumed role`
- Ruolo assegnato a un'istanza Amazon EC2
  - `aws:username`: (non presente)
  - `aws:userid`: *`role-id`*:*`ec2-instance-id`*
  - `aws:PrincipalType`: `Assumed role`
- Chiamante anonimo (solo Amazon SQS, Amazon SNS e Amazon S3)
  - `aws:username`: (non presente)
  - `aws:userid`: (non presente)
  - `aws:PrincipalType`: `Anonymous`

Per gli elementi di questo elenco, nota quanto segue:

- non presente significa che il valore non è riportato nelle informazioni della richiesta corrente e qualsiasi tentativo di trovare una corrispondenza avrà esito negativo e l'istruzione viene considerata non valida.
- *`role-id`*è un identificatore univoco assegnato a ciascun ruolo al momento della creazione. È possibile visualizzare l'ID del ruolo con il AWS CLI comando: `aws iam get-role --role-name` *`rolename`*

- *caller-specified-name*e *caller-specified-role-name* sono nomi che vengono passati dal processo chiamante (ad esempio un'applicazione o un servizio) quando effettua una chiamata per ottenere credenziali temporanee.
- *ec2-instance-id*è un valore assegnato all'istanza quando viene avviata e viene visualizzato nella pagina Istanze della console Amazon EC2. Puoi anche visualizzare l'ID dell'istanza eseguendo il AWS CLI comando: `aws ec2 describe-instances`

Informazioni disponibili nelle richieste per i principali federati

I principali federati vengono autenticati utilizzando un sistema diverso da IAM. Ad esempio, un'azienda potrebbe disporre di un'applicazione da utilizzare internamente che effettua chiamate verso AWS. Fornire un'identità IAM a ogni utente dell'azienda che utilizza l'applicazione potrebbe risultare poco pratico. Al contrario, l'azienda può utilizzare un'applicazione proxy (livello intermedio) con una singola identità IAM, oppure un provider di identità (IdP) SAML. L'applicazione proxy o l'IdP SAML autentica i singoli utenti individuali tramite la rete aziendale. Un'applicazione proxy può quindi utilizzare la propria identità di IAM per ottenere credenziali di sicurezza provvisorie per i singoli utenti. Un IdP SAML può in effetti scambiare informazioni sull'identità con credenziali di sicurezza AWS temporanee. Le credenziali temporanee possono quindi essere utilizzate per accedere alle risorse. AWS

Allo stesso modo, potresti creare un'applicazione per dispositivi mobili che richiede l'accesso alle risorse AWS . In questo caso, puoi ricorrere alla federazione OIDC, per fare in modo che l'app autentichi l'utente utilizzando un provider di identità conosciuto, come Login with Amazon, Amazon Cognito, Facebook o Google. A questo punto, l'applicazione potrà utilizzare le informazioni di autenticazione dell'utente fornite da tali provider per ottenere le credenziali di sicurezza temporanee per l'accesso alle risorse AWS .

Il modo consigliato per utilizzare la federazione OIDC è sfruttare Amazon Cognito e gli SDK mobili. AWS Per ulteriori informazioni, consulta gli argomenti seguenti:

- Guida per l'utente di Amazon Cognito
- Scenari comuni per le credenziali temporanee

Caratteri speciali

Alcune variabili di policy speciali e predefinite hanno valori fissi che consentono di rappresentare caratteri che altrimenti avrebbero un significato diverso. Se tali caratteri speciali fanno parte della

stringa per cui stai cercando una corrispondenza e vengono inseriti letteralmente, il loro significato sarebbe frainteso. Ad esempio, se nella stringa inserisci un asterisco (*), questo non sarà interpretato come un semplice asterisco, ma come un carattere jolly corrispondente a tutti i caratteri. In tali casi, puoi utilizzare le seguenti variabili di policy predefinite:

- ${*} - da usare quando devi inserire un asterisco (*).
- ${?} - da usare quando devi inserire un punto interrogativo (?).
- ${$} - da usare quando devi inserire il simbolo del dollaro ($).

Queste variabili di policy predefinite possono essere inserite in qualsiasi stringa in cui è possibile utilizzare le normali variabili di policy.

Specifica dei valori di default

Per aggiungere un valore di default a una variabile, racchiudi il valore di default tra virgolette singole (`' '`) e separa il testo della variabile e il valore di default con una virgola e uno spazio (`, `).

Ad esempio, se a un principale è applicato un tag`team=yellow`, è possibile accedere al bucket Amazon S3 `ExampleCorp's` denominato `amzn-s3-demo-bucket-yellow`. Una policy con questa risorsa consente ai membri del team di accedere al bucket del team, ma non a quelli di altri team. Per gli utenti senza tag team, la policy imposta un valore di default di `company-wide` per il nome del bucket. Questi utenti possono accedere solo al bucket `amzn-s3-demo-bucket-company-wide` dove possono visualizzare informazioni generali, come le istruzioni per entrare a far parte di un team.

```
"Resource":"arn:aws:s3:::amzn-s3-demo-bucket-${aws:PrincipalTag/team, 'company-wide'}"
```

Ulteriori informazioni

Per ulteriori informazioni sulle policy, consultare:

- Politiche e autorizzazioni in AWS Identity and Access Management
- Esempi di policy basate su identità IAM
- Documentazione di riferimento degli elementi delle policy JSON IAM
- Logica di valutazione delle policy
- Federazione OIDC

Elementi della policy JSON IAM: tipi di dati supportati

Questa sezione elenca i tipi di dati che sono supportati quando si specificano valori in policy JSON. Il linguaggio della policy non supporta tutti i tipi per ciascun elemento della policy; per informazioni su ciascun elemento, consultare le sezioni precedenti.

- Stringhe
- Numeri (interi e valori a virgola mobile)
- Booleano
- Null
- Elenchi
- Mappe
- Strutture (sono solo mappe nidificate)

La tabella seguente associa ogni tipo di dati alla serializzazione. Tieni presente che tutte le politiche devono essere attive UTF-8. Per informazioni sui tipi di dati JSON, vedere RFC 4627.

Tipo	JSON
Stringa	Stringa
Numero intero	Numero
Float	Numero
Booleano	true false
Null	null
Data	Stringa in linea con il profilo W3C di ISO 8601
IpAddress	Stringa in linea con RFC 4632
List	Array
Oggetto	Oggetto

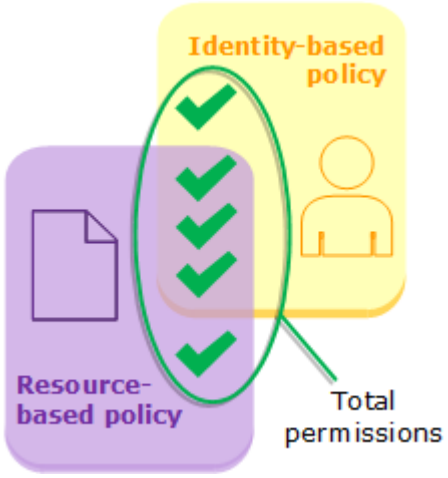
Logica di valutazione delle policy

Quando un principale tenta di utilizzare l' Console di gestione AWS, l' AWS API o il AWS CLI, quel principale invia una richiesta a. AWS Quando un AWS servizio riceve la richiesta, AWS completa diversi passaggi per determinare se consentire o rifiutare la richiesta.

1. Autenticazione: autentica AWS innanzitutto il principale che effettua la richiesta, se necessario. Questo passaggio non è necessario per alcuni servizi, ad esempio Amazon S3, che consentono alcune richieste da parte di utenti anonimi.
2. Elaborazione del contesto della richiesta— AWS elabora le informazioni raccolte nella richiesta per determinare quali politiche si applicano alla richiesta.
3. In che modo AWS la logica del codice di applicazione valuta le richieste per consentire o negare l'accesso— AWS valuta tutti i tipi di policy e l'ordine delle policy influisce sul modo in cui vengono valutate. AWS quindi elabora le politiche in base al contesto della richiesta per determinare se la richiesta è consentita o rifiutata.

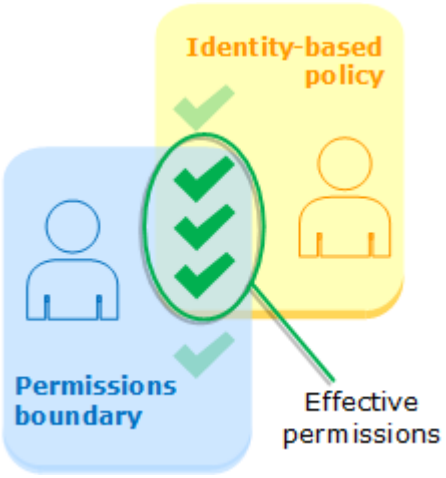
Valutazione delle policy basate su identità con policy basate su risorse

Identity-based le politiche e le politiche basate sulle risorse concedono le autorizzazioni alle identità o alle risorse a cui sono collegate. Quando un'entità IAM (utente o ruolo) richiede l'accesso a una risorsa all'interno dello stesso account, AWS valuta tutte le autorizzazioni concesse dalle politiche basate sull'identità e sulle risorse. Le autorizzazioni risultanti sono le autorizzazioni totali dei due tipi. Se un'azione è consentita da una policy basata sull'identità, una policy basata sulle risorse o entrambe, allora consente l'azione. AWS Un rifiuto esplicito in una di queste policy sostituisce l'autorizzazione.



Valutazione delle policy basate su identità con i limiti delle autorizzazioni

Quando si AWS valutano le politiche basate sull'identità e i limiti delle autorizzazioni per un utente, le autorizzazioni risultanti sono l'intersezione delle due categorie. Ciò significa che quando aggiungi un limite delle autorizzazioni a un utente con policy basate su identità esistenti, potresti ridurre il numero di operazioni che l'utente può eseguire. Di contro, quando rimuovi un limite delle autorizzazioni da un utente, potresti aumentare il numero di operazioni che può eseguire. Un rifiuto esplicito in una di queste policy sostituisce l'autorizzazione. Per informazioni su come altri tipi di policy vengono valutati con i limiti delle autorizzazioni, consulta Valutazione delle autorizzazioni valide con i limiti.



Valutazione delle politiche basate sull'identità con AWS Organizations SCP o RCP

Quando un utente appartiene a un account membro di un'organizzazione e accede a una risorsa per la quale non è configurata una policy basata sulle risorse, le autorizzazioni risultanti sono l'intersezione delle policy dell'utente, delle policy di controllo dei servizi (SCP) e delle policy di controllo delle risorse (RCP). Ciò significa che un'azione deve essere consentita da tutti e tre i tipi di policy. Un rifiuto esplicito nella policy basata sull'identità, una SCP o una RCP ha la precedenza sull'autorizzazione.



Puoi scoprire se il tuo account è un membro di un'organizzazione in AWS Organizations. I membri dell'organizzazione potrebbero essere influenzati da una SCP o una RCP. Per visualizzare questi dati utilizzando il AWS CLI comando o l'operazione AWS API, devi disporre delle autorizzazioni per l'azione per la tua entità. `organizations:DescribeOrganization` AWS Organizations È necessario disporre di autorizzazioni aggiuntive per eseguire l'operazione nella AWS Organizations console. Per sapere se un SCP o RCP sta negando l'accesso a una richiesta specifica o per modificare le autorizzazioni effettive, contatta l'amministratore. AWS Organizations

Elaborazione del contesto della richiesta

Quando AWS valuta e autorizza una richiesta, assembla le informazioni sulla richiesta in un contesto di richiesta. Il contesto della richiesta contiene tutte le informazioni che possono essere utilizzate nella valutazione delle policy.

- Principale: l'utente, il ruolo o il principale utente federato che ha inviato la richiesta. Le informazioni sull'entità principale includono le policy associate a tale entità principale.
- Azioni: una o più azioni che il principale desidera eseguire.
- Risorse: uno o più oggetti AWS risorsa su cui vengono eseguite le azioni o le operazioni.
- Dati sulla risorsa – I dati correlati alla risorsa che viene richiesta. Possono essere incluse informazioni quali un nome di tabella di DynamoDB o un tag su un'istanza Amazon EC2.
- Dati di ambiente: informazioni sull'indirizzo IP, l'agente utente, lo stato SSL abilitato o l'ora del giorno.

Queste informazioni vengono confrontate con le policy applicabili per determinare se accettare o rifiutare la richiesta. È possibile organizzare queste informazioni sulle proprietà utilizzando il modello

Principal, Action, Resource and Condition (PARC) per comprendere meglio come vengono valutate AWS le politiche.

Comprensione del modello PARC

Il modello PARC rappresenta il contesto della richiesta basato sui quattro elementi JSON nel linguaggio della policy:

- Principal: l'entità che effettua la richiesta. Un principale rappresenta un utente umano o un carico di lavoro programmatico che può essere autenticato e quindi autorizzato a eseguire azioni negli Account AWS.
- Action: l'operazione in corso. Spesso l'operazione verrà mappata a un'operazione API.
- Resource: la risorsa AWS su cui viene eseguita l'operazione.
- Condition: vincoli aggiuntivi che devono essere soddisfatti affinché la richiesta sia consentita.

Di seguito è illustrato un esempio di come il modello PARC potrebbe rappresentare un contesto della richiesta:

```
Principal: AIDA123456789EXAMPLE
Action: s3:CreateBucket
Resource: arn:aws:s3:::amzn-s3-demo-bucket1
Context:
- aws:UserId=AIDA123456789EXAMPLE:BobsSession
- aws:PrincipalAccount=123456789012
- aws:PrincipalOrgId=o-example
- aws:PrincipalARN=arn:aws:iam::AIDA123456789EXAMPLE:role/HR
- aws:MultiFactorAuthPresent=true
- aws:CurrentTime=...
- aws:EpochTime=...
- aws:SourceIp=...
- aws:PrincipalTag/dept=123
- aws:PrincipalTag/project=blue
- aws:RequestTag/dept=123
```

Importanza del contesto della richiesta

Comprendere il contesto della richiesta e il modo in cui interagisce con la valutazione delle policy è fondamentale per le seguenti finalità:

- Risoluzione dei problemi di accesso

- Progettazione di policy efficaci e sicure
- Comprensione dell'ambito completo delle autorizzazioni concesse da una policy
- Previsione dell'esito delle valutazioni delle policy in vari scenari

Visualizzando il contesto della richiesta utilizzando il modello PARC, è possibile comprendere più facilmente come AWS prende le decisioni di autorizzazione e progetta le politiche in modo più efficace.

Come AWS utilizza il contesto della richiesta

Durante la valutazione delle politiche, AWS confronta le informazioni nel contesto della richiesta con le informazioni specificate in tutte le politiche applicabili. Ciò include politiche basate sull'identità, politiche basate sulle risorse, limiti delle autorizzazioni IAM, Organizations, Organizations e policy di sessione. SCPs RCPs

Per ogni tipo di policy, utilizza il contesto della richiesta per verificare AWS :

- Se la policy si applica alla richiesta in base al principale.
- Se l'operazione richiesta è consentita sulla risorsa specificata.
- Se le condizioni specificate nella policy sono soddisfatte dal contesto della richiesta.

Il modo in cui AWS valuta le politiche dipende dai tipi di politiche che si applicano al contesto della richiesta. Questi tipi di policy possono essere utilizzati in un singolo Account AWS. Per ulteriori informazioni su questi tipi di policy, consulta Politiche e autorizzazioni in AWS Identity and Access Management. Per informazioni su come AWS valuta le politiche per l'accesso tra account diversi, consulta. Cross-account logica di valutazione delle politiche

- AWS Organizations politiche di controllo delle risorse (RCPs): AWS Organizations RCPs specificano le autorizzazioni massime disponibili per le risorse all'interno degli account di un'organizzazione o di un'unità organizzativa (OU). RCPs si applicano alle risorse degli account dei membri e influiscono sulle autorizzazioni effettive per i responsabili, incluse le Utente root dell'account AWS, indipendentemente dal fatto che i responsabili appartengano all'organizzazione. RCPs non si applicano alle risorse dell'account di gestione dell'organizzazione e alle chiamate effettuate da ruoli collegati al servizio. Se una RCP è presente, le autorizzazioni concesse da policy basate su identità e policy basate su risorse alle risorse negli account membri sono effettive solo se l'RCP consente l'operazione.

- AWS Organizations policy di controllo del servizio (SCPs): AWS Organizations SCPs specificano le autorizzazioni massime disponibili per i responsabili all'interno degli account di un'organizzazione o di un'unità organizzativa (OU). SCPs si applicano ai responsabili degli account dei membri, compresi ciascuno. Utente root dell'account AWS Se una SCP è presente, le autorizzazioni concesse da policy basate su identità e policy basate su risorse ai principali negli account membri sono effettive solo se l'SCP consente l'operazione. Le uniche eccezioni sono i principali dell'account di gestione dell'organizzazione e i ruoli collegati ai servizi.
- Policy basate sulle risorse: le policy basate sulle risorse concedono le autorizzazioni per i principali specificati nella policy. Le autorizzazioni definiscono ciò che l'entità principale può fare con la risorsa a cui è collegata la policy.
- Limiti delle autorizzazioni: i limiti delle autorizzazioni sono una funzionalità avanzata che imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo). Quando si imposta un limite delle autorizzazioni per un'entità, l'entità può eseguire solo le operazioni consentite dalle sue policy basate su identità e dai suoi limiti delle autorizzazioni. In alcuni casi, un rifiuto implicito in un limite delle autorizzazioni può limitare le autorizzazioni concesse da una policy basata sulle risorse. Per ulteriori informazioni, consulta In che modo AWS la logica del codice di applicazione valuta le richieste per consentire o negare l'accesso.
- Policy basate su identità: le policy basate su identità sono collegate a un'identità IAM (utente, gruppo di utenti o ruolo) e concede le autorizzazioni per entità IAM (utenti e ruoli). Se a una richiesta si applicano solo politiche basate sull'identità, AWS verifica che tutte queste politiche ne accertino almeno una. `Allow`
- Policy di sessione: le policy di sessione sono policy che si inviano come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o una sessione utente federato. Per creare una sessione del ruolo in modo programmatico, è possibile utilizzare una delle operazioni API `AssumeRole*`. Quando esegui questa operazione e passi le policy di sessione, le autorizzazioni della sessione risultante sono l'intersezione della policy basata su identità dell'utente dell'entità IAM e delle policy di sessione. Per creare una sessione per l'utente federato, si utilizzano le chiavi di accesso dell'utente IAM per chiamare in modo programmatico l'operazione API `GetFederationToken`. Per ulteriori informazioni, consulta Policy di sessione.

Occorre ricordare che un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione.

> **Note**
>
> AWS Organizations le politiche dichiarative consentono di dichiarare e applicare in modo centralizzato la configurazione desiderata per un determinato Servizio AWS aspetto su

larga scala all'interno dell'organizzazione. Poiché le policy dichiarative vengono applicate direttamente a livello di servizio, non influiscono direttamente sulle richieste di valutazione delle policy e non sono incluse nel contesto della richiesta. Per ulteriori informazioni, consulta Politiche dichiarative nella Guida per l'utente. AWS Organizations

Esempio di valutazione delle policy utilizzando il modello PARC

Per illustrare come il contesto della richiesta interagisce con la valutazione delle policy, prendiamo in considerazione un esempio di policy:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:CreateBucket",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/dept": "123"
        }
      }
    }
  ]
}
```

In questo esempio, la policy consentirebbe l'operazione `CreateBucket` solo quando il contesto della richiesta include un valore `aws:PrincipalTag/dept` di "123" e la risorsa corrisponde al nome del bucket `amzn-s3-demo-bucket1`. La tabella seguente mostra come AWS utilizza il contesto della richiesta per valutare questa policy e prendere decisioni di autorizzazione.

Policy	Contesto della richiesta	Risultato della valutazione
`"StringEquals": { "aws:PrincipalTag/dept": "123"`	`Principal: AIDA12345 6789EXAMPLE`	Partita

Policy	Contesto della richiesta	Risultato della valutazione
} 	Action: s3:Create Bucket Resource: arn:aws:s 3:::amzn-s3-demo-b ucket1 Context: - aws:PrincipalTag/ dept=123	

"StringEquals": {
"aws:PrincipalTag/
dept": "123"
}
	Principal: AIDA12345
	6789EXAMPLE
	Action: s3: *DeleteBuc*
	ket
	Resource: arn:aws:s
	3:::amzn-s3-demo-b
	ucket1
	Context:
	- aws:PrincipalTag/
	dept=123
		Nessuna corrispondenza

"StringEquals": {
"aws:PrincipalTag/
dept": "123"
}
	Principal: AIDA12345
	6789EXAMPLE
	Action: s3:Create
	Bucket
	Resource: arn:aws:s
	3:::amzn-s3-demo-b
	ucket1
	Context:
	- aws:PrincipalTag/
	dept= *321*
		Nessuna corrispondenza

Policy	Contesto della richiesta	Risultato della valutazione

`"StringEquals": { "aws:PrincipalTag/dept": "123" }`	`Principal: AIDA123456789EXAMPLE`
	`Action: s3:CreateBucket`
	`Resource: arn:aws:s3:::amzn-s3-demo-bucket1`
	`Context:`
	Nessun `aws:PrincipalTag` nella richiesta.
		Nessuna corrispondenza

In che modo AWS la logica del codice di applicazione valuta le richieste per consentire o negare l'accesso

Il codice di AWS applicazione decide se una richiesta inviata a AWS deve essere accolta o rifiutata. AWS valuta tutte le politiche applicabili al contesto della richiesta. Di seguito è riportato un riepilogo della logica di valutazione delle AWS politiche.

- Per impostazione predefinita, tutte le richieste vengono negate implicitamente con l'eccezione dell' Utente root dell'account AWS, che ha accesso completo.
- Per essere consentite, le richieste devono essere esplicitamente consentite da una policy o da un insieme di policy che seguono la logica di valutazione riportata di seguito.
- Un rifiuto esplicito sovrascrive un consenso esplicito.

La valutazione della policy può variare a seconda che la richiesta riguardi un singolo account oppure più account. Per i dettagli su come viene presa una decisione di valutazione della policy per un ruolo o un utente IAM all'interno di un singolo account, consulta Valutazione delle policy per le richieste all'interno di un singolo account. Per i dettagli su come viene presa una decisione di valutazione della policy per le richieste tra più account, consulta Cross-account logica di valutazione delle politiche.

- Rifiuta valutazione: per impostazione predefinita, tutte le richieste vengono rifiutate. Si tratta del cosiddetto rifiuto implicito. Il codice di AWS applicazione valuta tutte le politiche all'interno dell'account che si applicano alla richiesta. Queste includono AWS Organizations SCP e RCP, policy basate sulle risorse, policy basate sull'identità, limiti delle autorizzazioni IAM e policy di

sessione. In tutte le policy, il codice di attuazione cerca un'istruzione `Deny` applicabile alla richiesta. Questa azione si chiama rifiuto esplicito. Se il codice di applicazione trova anche un solo rifiuto esplicito applicabile, restituisce Rifiuta come decisione finale. Se non c'è un rifiuto esplicito, la valutazione del codice di attuazione continua.

- AWS Organizations RCP: il codice di applicazione valuta AWS Organizations le politiche di controllo delle risorse (RCP) applicabili alla richiesta. Le RCP si applicano alle risorse dell'account a cui sono collegate le RCP. Se il codice di applicazione non trova alcune istruzione `Allow` applicabile nelle RCP, restituisce Rifiuta come decisione finale. Tieni presente che una policy gestita da AWS chiamata `RCPFullAWSAccess` viene creata e allegata automaticamente a ogni entità dell'organizzazione, inclusa la root, ogni unità organizzativa e Account AWS quando sono abilitate le RCP. `RCPFullAWSAccess` non può essere scollegato, quindi ci sarà sempre una istruzione `Allow`. Se non c'è alcuna RCP oppure se l'RCP consente l'operazione richiesta, la valutazione del codice di applicazione continua.
- AWS Organizations SCP: il codice di applicazione valuta le politiche di controllo del AWS Organizations servizio (SCP) applicabili alla richiesta. Le SCP si applicano alle entità dell'account in cui sono collegate le SCP. Se il codice di applicazione non trova alcune istruzione `Allow` applicabile nelle SCP, restituisce Rifiuta come decisione finale. Se non c'è alcuna SCP oppure se l'SCP consente l'operazione richiesta, la valutazione del codice di attuazione continua.
- Resource-based politiche: all'interno dello stesso account, le politiche basate sulle risorse influiscono sulla valutazione delle politiche in modo diverso a seconda del tipo di principale che accede alla risorsa e del principale consentito nella politica basata sulle risorse. A seconda del tipo di principale, un `Allow` in una policy basata sulle risorse può comportare una decisione definitiva di `Allow`, anche se è presente un rifiuto implicito in una policy basata su identità, un limite delle autorizzazioni o una policy di sessione.

  Per la maggior parte delle risorse, è necessario solo un'autorizzazione `Allow` esplicita per il principale in una policy basata sulle identità o una policy basata sulle risorse per concedere l'accesso. Le politiche di fiducia del ruolo IAM e le politiche chiave del KMS sono eccezioni a questa logica, perché devono consentire esplicitamente l'accesso ai committenti. ??? Resource-based politiche per servizi diversi da IAM e AWS KMS possono anche richiedere una `Allow` dichiarazione esplicita all'interno dello stesso account per concedere l'accesso. Per ulteriori informazioni, consulta la documentazione per il servizio specifico che stai utilizzando.

  Per le richieste di valutazione delle policy per un singolo account, la policy basata sulle risorse differisce dagli altri tipi di policy se il principale specificato è un utente IAM, un ruolo IAM o un principale di sessione. I principali della sessione includono sessioni di ruolo IAM o principali utente

federato AWS STS. Se una policy basata sulle risorse concede l'autorizzazione direttamente all'utente IAM o al principale di sessione che sta effettuando la richiesta, un rifiuto implicito in una policy basata sull'identità, un limite di autorizzazioni o una policy di sessione non influiscono sulla decisione finale.

- Ruolo IAM: Resource-based le politiche che concedono autorizzazioni a un ruolo IAM ARN sono limitate da una negazione implicita in un limite di autorizzazioni o in una policy di sessione. È possibile specificare l'ARN del ruolo nell'elemento Principal o nella chiave di condizione `aws:PrincipalArn`. In entrambi i casi, il principale che effettua la richiesta è la sessione del ruolo IAM.

  I limiti delle autorizzazioni e le policy di sessione non limitano le autorizzazioni concesse tramite la chiave di condizione `aws:PrincipalArn` con un carattere jolly (*) nell'elemento Principal, a meno che le policy basate sulle identità non contengano un rifiuto esplicito. Per ulteriori informazioni, consulta Principali ruolo IAM.

  Esempio di ARN di ruolo

  ```
  arn:aws:iam::111122223333:role/examplerole
  ```

- Sessione come ruolo IAM: all'interno dello stesso account, le policy basate sulle risorse che concedono le autorizzazioni all'ARN della sessione come ruolo IAM concedono le autorizzazioni direttamente alla sessione come ruolo assunto. Le autorizzazioni concesse direttamente a una sessione non sono limitate da un rifiuto implicito in una policy basata su identità, da un limite delle autorizzazioni o da una policy di sessione. Quando si assume un ruolo e si effettua una richiesta, il principale che effettua la richiesta è l'ARN della sessione come ruolo IAM e non l'ARN del ruolo stesso. Per ulteriori informazioni, consulta Principali della sessione come ruolo.

  Esempio di ARN della sessione come ruolo

  ```
  arn:aws:sts::111122223333:assumed-role/examplerole/examplerolesessionname
  ```

- Utente IAM: all'interno dello stesso account, le politiche basate sulle risorse che concedono autorizzazioni all'ARN di un utente IAM (ovvero, non una sessione come utente federato) non sono limitate da un rifiuto implicito in una policy basata su identità o in un limite delle autorizzazioni.

  Esempio di ARN dell'utente IAM

```
arn:aws:iam::111122223333:user/exampleuser
```

- AWS STS utente principale federato: una sessione utente federata è una sessione creata chiamando. `GetFederationToken` Quando un utente federato effettua una richiesta, il principale che effettua la richiesta è l'ARN dell'utente federato e non l'ARN dell'utente IAM che ha eseguito la federazione. All'interno dello stesso account, le policy basate sulle risorse che concedono le autorizzazioni all'ARN dell'utente federato concedono le autorizzazioni direttamente alla sessione. Le autorizzazioni concesse direttamente a una sessione non sono limitate da un rifiuto implicito in una policy basata su identità, da un limite delle autorizzazioni o da una policy di sessione.

  Tuttavia, se una policy basata sulle risorse concede l'autorizzazione all'ARN dell'utente IAM che ha eseguito la federazione, le richieste fatte dall'utente federato durante la sessione sono limitate da un rifiuto implicito in un limite di autorizzazione o in una policy di sessione.

  Esempio di ARN di sessione utente federato

  ```
  arn:aws:sts::111122223333:federated-user/exampleuser
  ```

- Identity-based politiche: il codice di applicazione controlla le politiche basate sull'identità per il principale. Per un utente IAM, queste includono le policy utente e le policy dei gruppi a cui appartiene l'utente. Se non ci sono policy basate su identità o istruzioni nelle policy basata su identità che consentono l'operazione richiesta, la richiesta viene rifiutata implicitamente e il codice di applicazione restituisce Rifiuta come decisione finale. Se un'istruzione in qualsiasi policy basata su identità applicabile consente l'operazione richiesta, la valutazione del codice continua.
- Limiti delle autorizzazioni IAM: il codice di applicazione controlla se l'entità IAM utilizzata dal principale ha un limite delle autorizzazioni. Se la policy utilizzata per impostare il limite delle autorizzazioni non consente l'operazione richiesta, la richiesta viene rifiutata implicitamente. Il codice di attuazione restituisce Deny (Rifiuta) come decisione finale. Se non c'è alcun limite delle autorizzazioni oppure se il limite delle autorizzazioni consente l'operazione richiesta, la valutazione del codice continua.
- Policy di sessione: il codice di applicazione verifica se il principale è un principale di sessione. I titolari delle sessioni includono una sessione di ruolo IAM o una sessione utente federata AWS STS . Se il principale non è un principale di sessione, il codice di attuazione restituisce Allow (Consenti) come decisione finale.

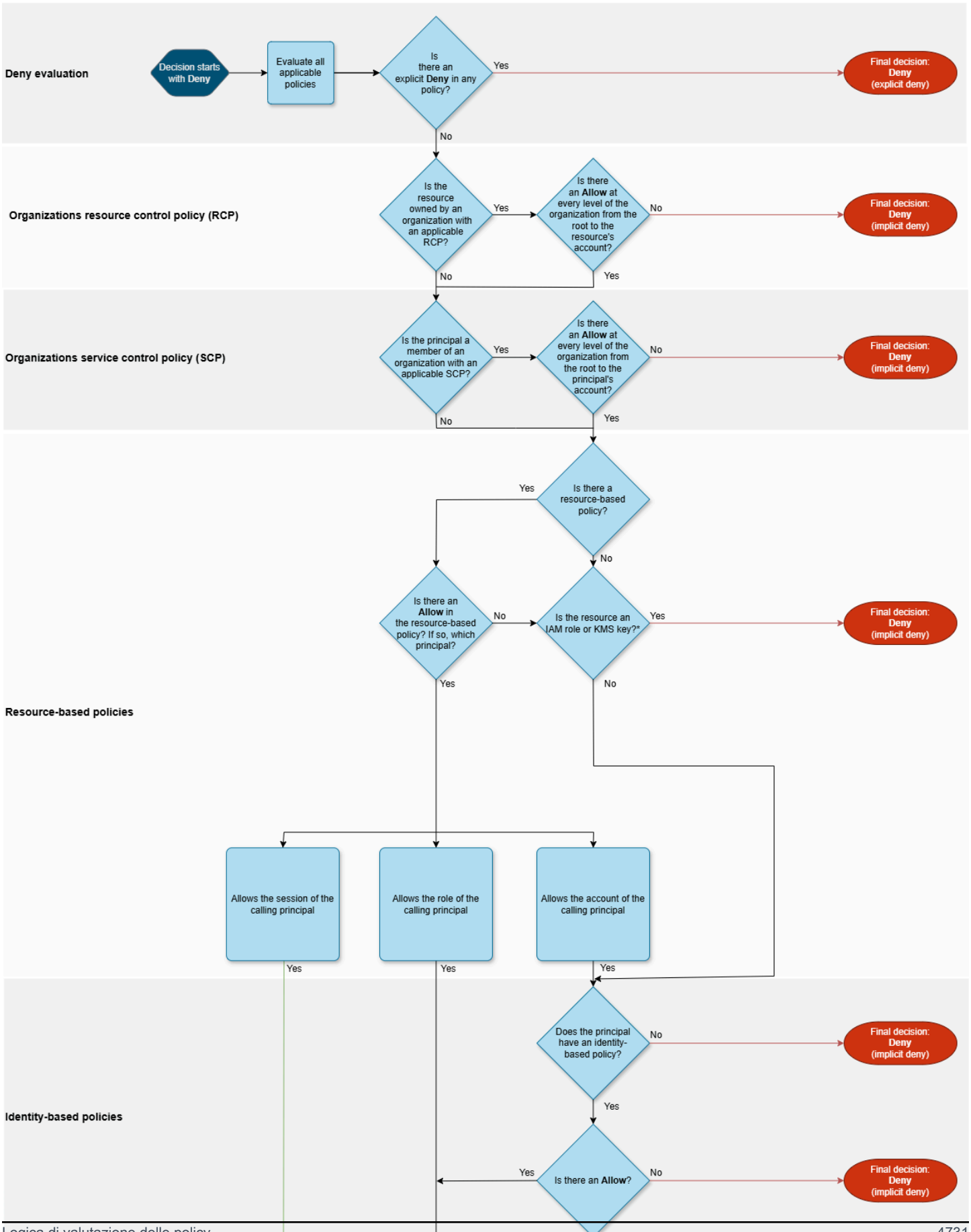
Per i principali della sessione, il codice di applicazione verifica se una policy di sessione è passata nella richiesta. Puoi passare una policy di sessione mentre usi l' AWS API AWS CLI or per ottenere credenziali temporanee per un ruolo o un utente principale AWS STS federato. Se non hai approvato una policy di sessione, viene creata una policy di sessione predefinita e il codice di applicazione restituisce Consenti come decisione finale.

- Se una policy di sessione è presente e non consente l'operazione richiesta, la richiesta viene rifiutata implicitamente. Il codice di attuazione restituisce Deny (Rifiuta) come decisione finale.
- Il codice di applicazione verifica se il principale è una sessione del ruolo. Se il principale è una sessione di ruolo, la richiesta è Consentita. In caso contrario, la richiesta viene negata implicitamente e il codice di applicazione restituisce Rifiuta come decisione finale.
- Se una policy di sessione è presente e consente l'operazione richiesta, il codice di attuazione restituisce Consenti come decisione finale.

Valutazione delle policy per le richieste all'interno di un singolo account

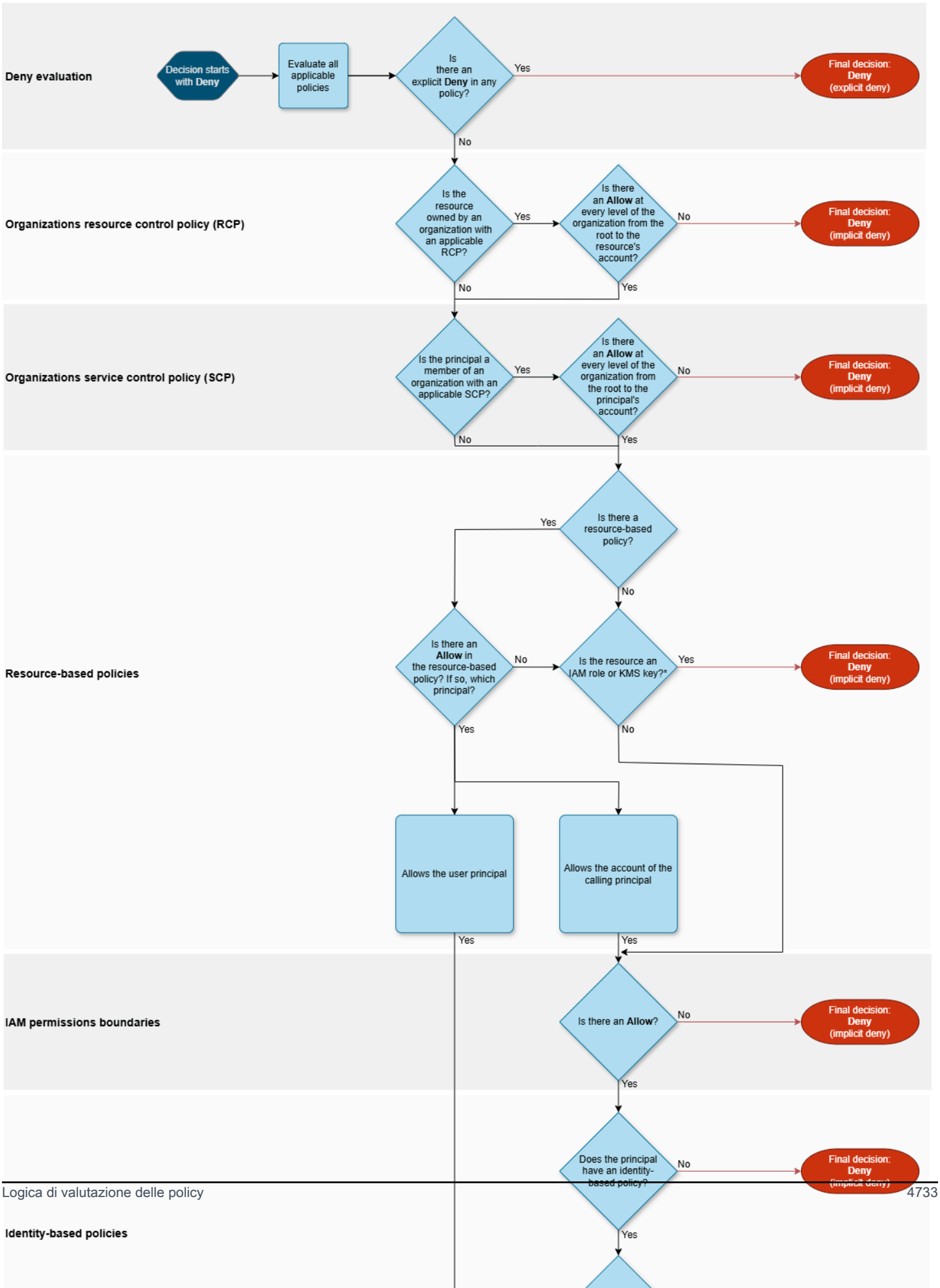
Valutazione della policy per un ruolo IAM

Il seguente diagramma di flusso fornisce dettagli su come viene presa una decisione di valutazione della policy per un ruolo IAM all’interno di un singolo account.


Deny evaluation
Decision starts with Deny
Evaluate all applicable policies
Is there an explicit Deny in any policy?
Yes
Final decision: Deny (explicit deny)
No
Organizations resource control policy (RCP)
Is the resource owned by an organization with an applicable RCP?
Yes
Is there an Allow at every level of the organization from the root to the resource's account?
No
Final decision: Deny (implicit deny)
No
Yes
Organizations service control policy (SCP)
Is the principal a member of an organization with an applicable SCP?
Yes
Is there an Allow at every level of the organization from the root to the principal's account?
No
Final decision: Deny (implicit deny)
No
Yes
Resource-based policies
Yes
Is there a resource-based policy?
No
Is there an Allow in the resource-based policy? If so, which principal?
No
Is the resource an IAM role or KMS key?*
Yes
Final decision: Deny (implicit deny)
Yes
No
Allows the session of the calling principal
Allows the role of the calling principal
Allows the account of the calling principal
Yes
Yes
Yes
Identity-based policies
Does the principal have an identity-based policy?
No
Final decision: Deny (implicit deny)
Yes
Yes
Is there an Allow?
No
Final decision: Deny (implicit deny)
IAM permissions boundaries
Does the principal have a permissions
Yes
Is there an Allow?
No
Final decision: Deny

Valutazione della policy per un utente IAM

Il seguente diagramma di flusso fornisce dettagli su come viene presa una decisione di valutazione della policy per un utente IAM all'interno di un singolo account.


Deny evaluation
Decision starts with Deny
Evaluate all applicable policies
Is there an explicit Deny in any policy?
Yes
Final decision: Deny (explicit deny)
No
Organizations resource control policy (RCP)
Is the resource owned by an organization with an applicable RCP?
Yes
Is there an Allow at every level of the organization from the root to the resource's account?
No
Final decision: Deny (implicit deny)
No
Yes
Organizations service control policy (SCP)
Is the principal a member of an organization with an applicable SCP?
Yes
Is there an Allow at every level of the organization from the root to the principal's account?
No
Final decision: Deny (implicit deny)
No
Yes
Resource-based policies
Is there a resource-based policy?
Yes
No
Is there an Allow in the resource-based policy? If so, which principal?
No
Is the resource an IAM role or KMS key?*
Yes
Final decision: Deny (implicit deny)
Yes
No
Allows the user principal
Allows the account of the calling principal
Yes
Yes
IAM permissions boundaries
Is there an Allow?
No
Final decision: Deny (implicit deny)
Yes
Identity-based policies
Does the principal have an identity-based policy?
No
Final decision: Deny (implicit deny)
Yes

Esempio di valutazione delle policy basate su identità e delle policy basate su risorse

I tipi di policy più comuni sono quelle basate su identità e quelle basate su risorse. Quando viene richiesto l'accesso a una risorsa, AWS valuta tutte le autorizzazioni concesse dalle policy per almeno un Allow all'interno dello stesso account. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione.

> **Important**
>
> Se la policy basata sull'identità o la policy basata sulle risorse all'interno dello stesso account consente la richiesta e l'altra no, la richiesta è comunque consentita.

Supponiamo che Carlos, con il nome utente `carlossalazar`, voglia salvare un file nel bucket `amzn-s3-demo-bucket-carlossalazar-logs` di Amazon S3.

Supponi inoltre che la policy seguente sia collegata all'utente IAM `carlossalazar`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3ListRead",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowS3Self",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket-carlossalazar/*",
                "arn:aws:s3:::amzn-s3-demo-bucket-carlossalazar"
```

```
            ]
        },
        {
            "Sid": "DenyS3Logs",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::*log*"
        }
    ]
}
```

L'istruzione `AllowS3ListRead` in questa policy consente a Carlos di visualizzare un elenco di tutti i bucket nell'account. L'istruzione `AllowS3Self` consente a Carlos l'accesso completo al bucket con lo stesso nome usato per il nome utente. L'istruzione `DenyS3Logs` nega a Carlos l'accesso a qualsiasi bucket di S3 il cui nome includa `log`.

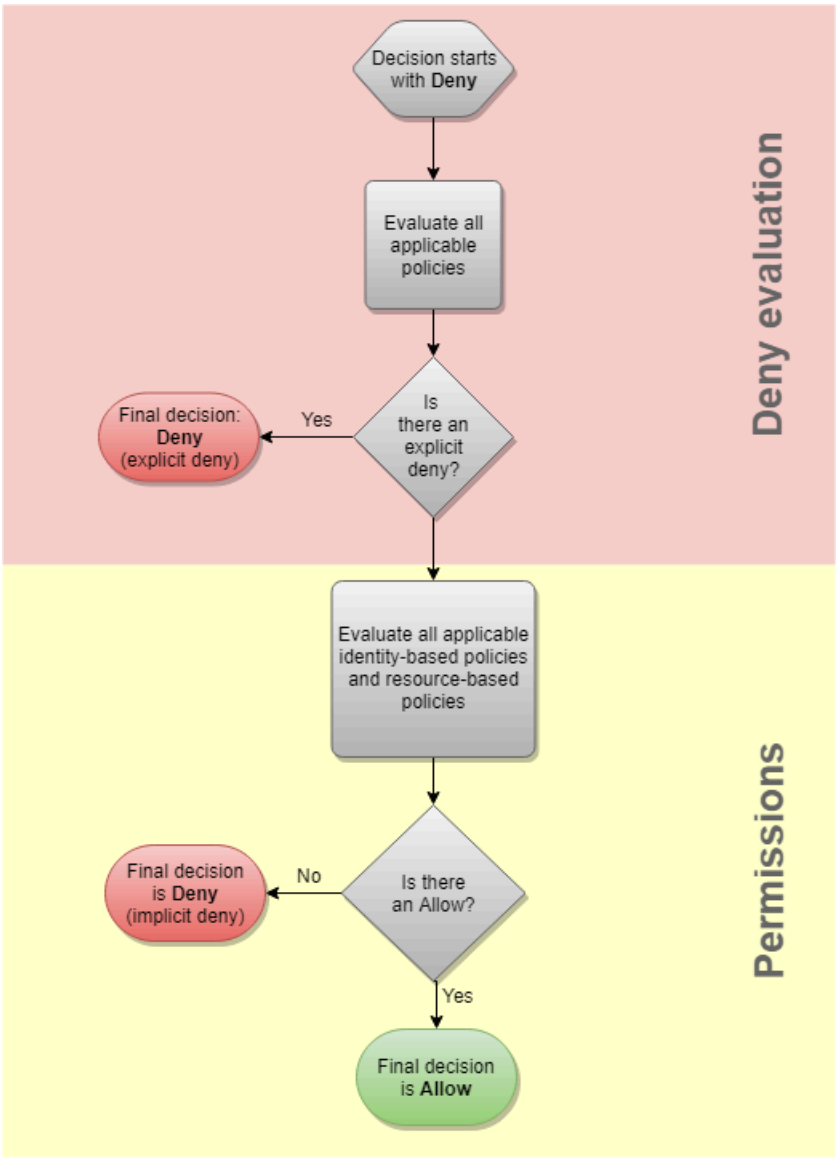
Inoltre, la seguente policy basata su risorse (detta policy del bucket) viene collegata al bucket `amzn-s3-demo-bucket-carlossalazar`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/carlossalazar"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket-carlossalazar/*",
                "arn:aws:s3:::amzn-s3-demo-bucket-carlossalazar"
            ]
        }
    ]
}
```

Questa policy specifica che solo l'utente `carlossalazar` può accedere al bucket `amzn-s3-demo-bucket-carlossalazar`.

Quando Carlos richiede di salvare un file nel `amzn-s3-demo-bucket-carlossalazar-logs` bucket, AWS determina quali politiche si applicano alla richiesta. In questo caso, sono applicabili solo la policy basata su identità e la policy basata su risorse. Entrambe sono policy di autorizzazione. Poiché non ci sono limiti di autorizzazione applicabili, la logica di valutazione viene ridotta a quanto segue.



AWS verifica innanzitutto la presenza di una `Deny` dichiarazione che si applichi al contesto della richiesta. Ne trova una, perché la policy basata su identità rifiuta esplicitamente a Carlos l'accesso a qualsiasi bucket di S3 utilizzato per la creazione di log. A Carlos viene negato l'accesso.

Supponiamo che poi si renda conto del suo errore e cerchi di salvare il file nel `amzn-s3-demo-bucket-carlossalazar` bucket. AWS verifica la presenza di una `Deny` dichiarazione e non la trova. Verifica quindi le policy di autorizzazione. Sia la policy basata sull'identità che la policy basata sulle risorse consentono la richiesta. Pertanto, AWS consente la richiesta. Se uno dei due avesse rifiutato esplicitamente l'istruzione, la richiesta sarebbe stata negata. Se uno dei tipi di policy consente la richiesta e l'altro no, la richiesta è comunque consentita.

Cross-account logica di valutazione delle politiche

Puoi consentire a un principale in un account di accedere alle risorse in un secondo account. Questo è chiamato accesso tra account. Quando consenti l'accesso tra account, l'account in cui si trova il principale viene denominato l'account attendibile . L'account in cui si trova la risorsa è l'account che concede fiducia .

Per consentire l'accesso tra account, collega una policy basata sulle risorse alla risorsa che desideri condividere. Devi inoltre collegare una policy basata sull'identità all'identità che agisce come il principale nella richiesta. La policy basata su risorse nell'account che concede fiducia deve specificare il principale dell'account attendibile che avrà accesso alla risorsa. Puoi specificare l'intero account o i relativi utenti IAM, i AWS STS federated user principal, i ruoli IAM o le sessioni con ruoli presunti. Puoi anche specificare un AWS servizio come principale. Per ulteriori informazioni, consulta Come specificare un principale.

La policy basata su identità del principale deve consentire l'accesso richiesto alla risorsa nel servizio che concede fiducia. A questo scopo, specifica l'ARN della risorsa.

In IAM, puoi collegare una policy basata sulle risorse a un ruolo IAM per consentire ai principali in altri account di assumere tale ruolo. La policy basata sulle risorse del ruolo è denominata policy di attendibilità del ruolo. Dopo aver assunto tale ruolo, i principali consentiti possono utilizzare le credenziali temporanee risultanti per accedere a più risorse nell'account. Questo accesso è definito nella policy di autorizzazioni basata su identità del ruolo. Per informazioni sul perché consentire l'accesso tra account utilizzando i ruoli è diverso dal consentire l'accesso tra account utilizzando altre policy basate sulle risorse, consulta Accesso alle risorse multi-account in IAM.

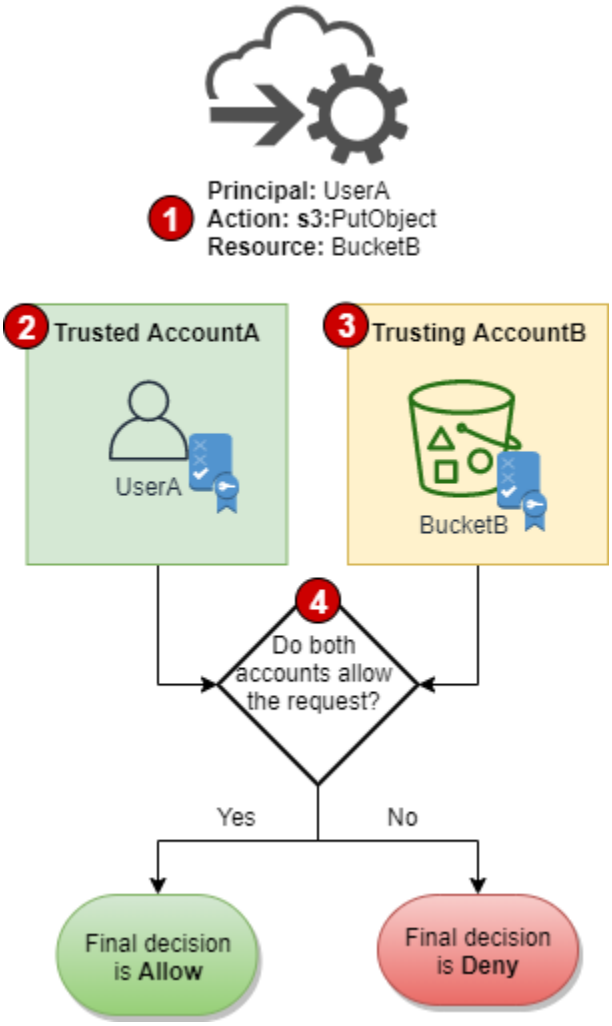
> **Important**
>
> Altri servizi possono influire sulla logica di valutazione dei criteri. Ad esempio, AWS Organizations supporta le politiche di controllo dei servizi e le politiche di controllo delle risorse che possono essere applicate ai principali e alle risorse di uno o più account. AWS

Resource Access Manager supporta frammenti di policy che controllano le azioni che i responsabili sono autorizzati a eseguire sulle risorse condivise con loro.

Determinare se una richiesta tra account è consentita

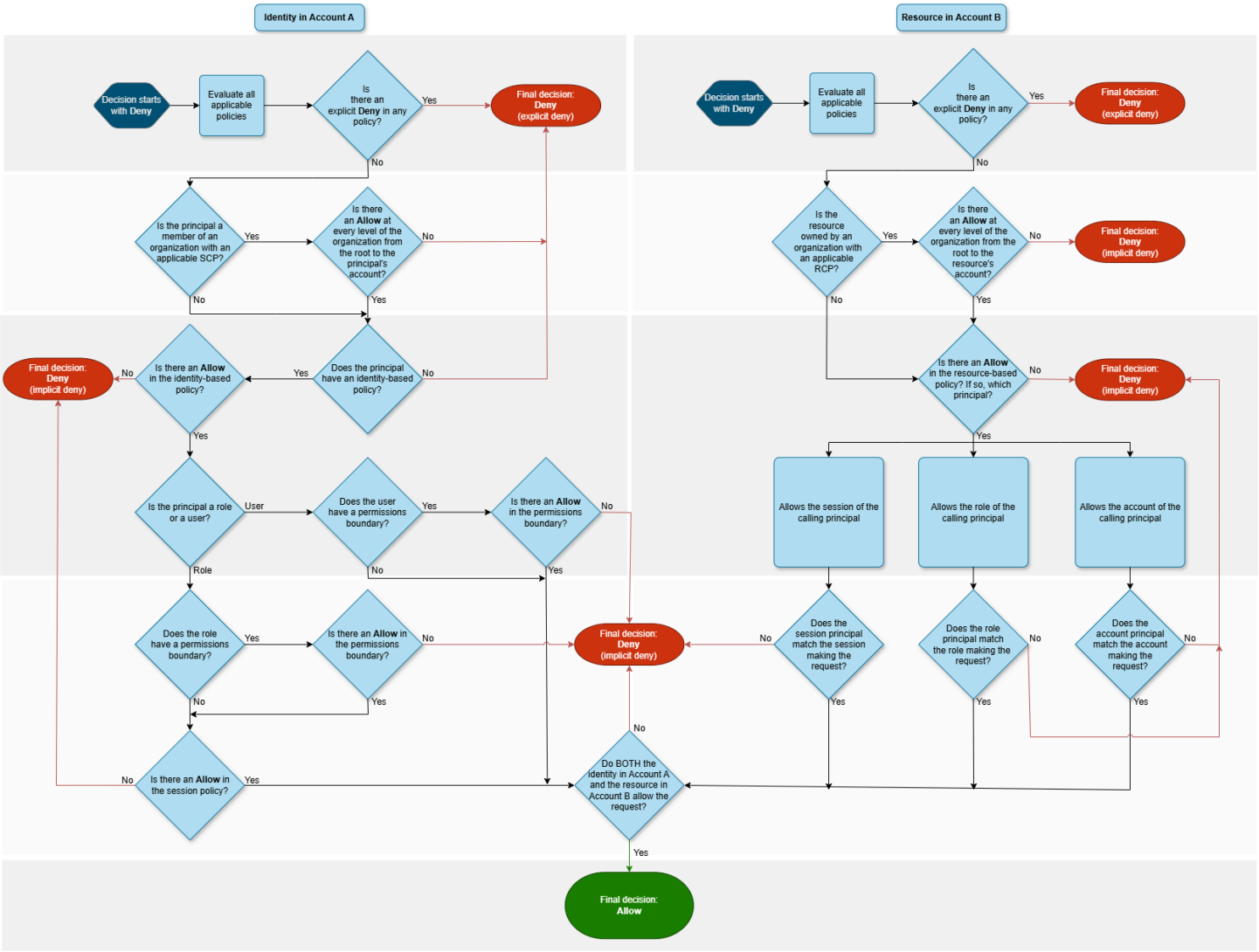
Per le richieste tra account, il richiedente nell'`AccountA` attendibile deve disporre di una policy basata su identità. Tale policy deve consentire di effettuare una richiesta alla risorsa nell' che concede fiducia `AccountB`. Inoltre, la policy basata sulle risorse nell'`AccountB` deve consentire al richiedente nell'`AccountA` di accedere alla risorsa.

Quando effettui una richiesta tra più account, AWS esegue due valutazioni. AWS valuta la richiesta nell'account fiduciario e nell'account fidato. Per ulteriori informazioni su come una richiesta viene valutata all'interno di un singolo account, consulta In che modo AWS la logica del codice di applicazione valuta le richieste per consentire o negare l'accesso. La richiesta è consentita solo se entrambe le valutazioni restituiscono come decisione `Allow`.



1. Quando un principale in un account effettua una richiesta per accedere a una risorsa in un altro account, questa è una richiesta tra account.
2. Il principale che esegue la richiesta esiste nell'account attendibile (`AccountA`). Quando AWS valuta questo account, controlla la policy basata su identità e le eventuali policy che possono limitare una policy basata su identità. Per ulteriori informazioni, consulta Valutazione delle policy basate su identità con i limiti delle autorizzazioni.
3. La risorsa richiesta esiste nell'account che concede fiducia (`AccountB`). Quando AWS valuta questo account, controlla la policy basata sulle risorse collegata alla risorsa richiesta e le eventuali policy che possono limitare una policy basata sulle risorse. Per ulteriori informazioni, consulta Valutazione delle policy basate su identità con policy basate su risorse.
4. AWS consente la richiesta solo se entrambe le valutazioni delle politiche dell'account consentono la richiesta.

Il seguente diagramma di flusso fornisce un'illustrazione più dettagliata di come viene presa una decisione di valutazione delle policy per una richiesta multi-account. Ancora una volta, AWS consente la richiesta solo se entrambe le valutazioni delle politiche dell'account consentono la richiesta.



Esempio di valutazione della policy multiaccount

Nell'esempio seguente viene illustrato uno scenario in cui a un utente in un account vengono concesse autorizzazioni da una policy basata sulle risorse in un secondo account.

Supponiamo che Carlos sia uno sviluppatore con un ruolo IAM denominato `Demo` nell'account 111111111111. Vuole salvare un file nel bucket `amzn-s3-demo-bucket-production-logs` di Amazon S3 nell'account 222222222222.

Supponiamo inoltre che la policy seguente sia collegata al ruolo IAM `Demo`.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3ListRead",
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Sid": "AllowS3ProductionObjectActions",
            "Effect": "Allow",
            "Action": "s3:*Object*",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-production/*"
        },
        {
            "Sid": "DenyS3Logs",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::*log*",
                "arn:aws:s3:::*log*/*"
            ]
        }
    ]
}
```

L'istruzione `AllowS3ListRead` in questa policy consente a Carlos di visualizzare un elenco di tutti i bucket in Amazon S3. L'istruzione `AllowS3ProductionObjectActions` consente a Carlos l'accesso completo al bucket `amzn-s3-demo-bucket-production`.

Inoltre, la seguente policy basata sulle risorse (denominata policy del bucket) è collegata al bucket `amzn-s3-demo-bucket-production` nell'account 222222222222.

JSON

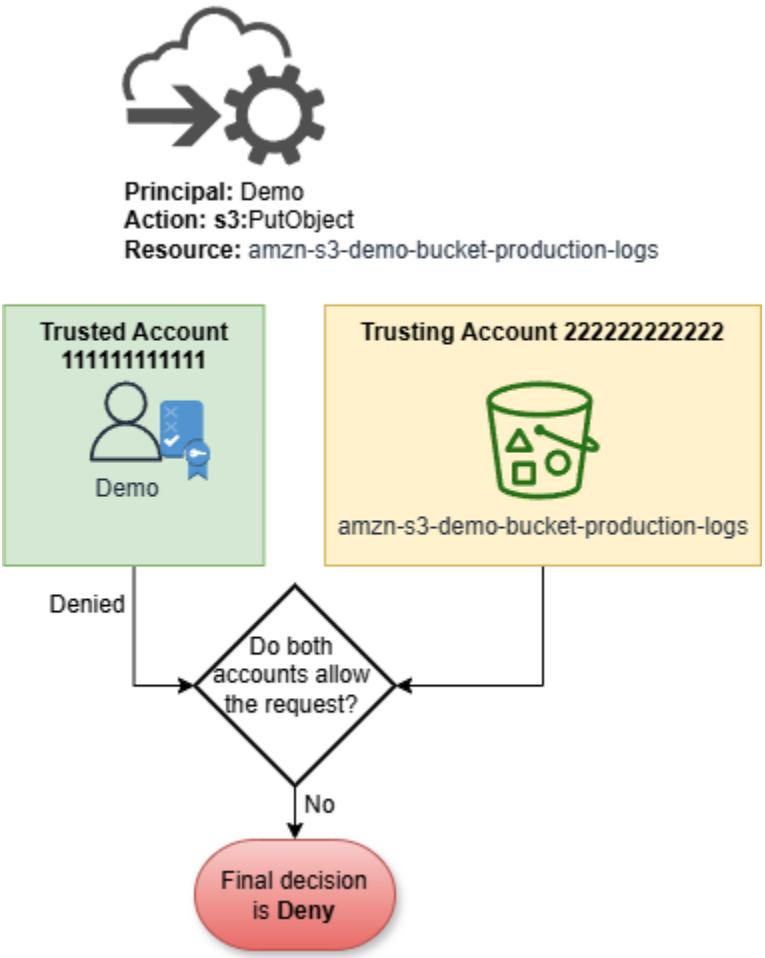
```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetObject*",
                "s3:PutObject*",
                "s3:ReplicateObject",
                "s3:RestoreObject"
            ],
            "Principal": { "AWS": "arn:aws:iam::111111111111:role/Demo" },
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-production/*"
        }
    ]
}
```

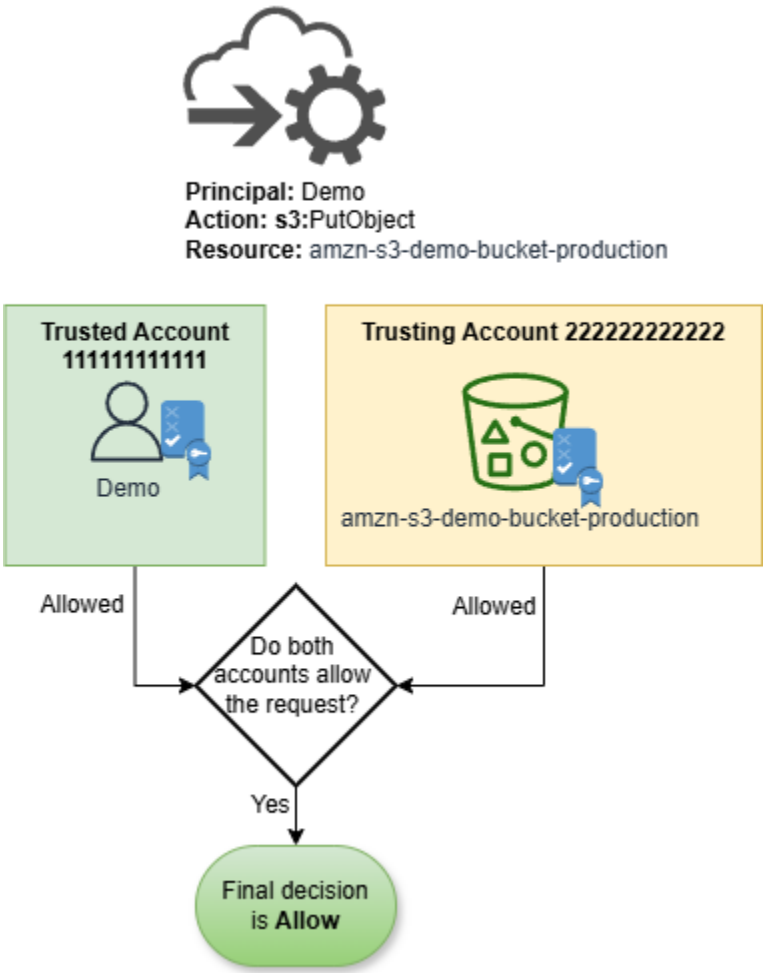
Questa policy consente al ruolo `Demo` di accedere agli oggetti nel bucket `amzn-s3-demo-bucket-production`. Il ruolo può creare e modificare, ma non eliminare gli oggetti nel bucket. Il ruolo non riesce a gestire il bucket da solo.

Quando Carlos richiede di salvare un file nel `amzn-s3-demo-bucket-production-logs` bucket, AWS determina quali politiche si applicano alla richiesta. In questo caso, la policy basata su identità collegata al ruolo `Demo` è la sola policy valida nell'account `111111111111`. Nell'account `222222222222`, non esiste una policy basata sulle risorse collegata al bucket `amzn-s3-demo-bucket-production-logs`. Quando AWS valuta l'account`111111111111`, restituisce una decisione di. `Deny` Questo perché l'istruzione `DenyS3Logs` nella policy basata su identità nega esplicitamente l'accesso a qualsiasi bucket di log. Per ulteriori informazioni su come una richiesta viene valutata all'interno di un singolo account, consulta In che modo AWS la logica del codice di applicazione valuta le richieste per consentire o negare l'accesso.

Poiché la richiesta viene negata esplicitamente all'interno di uno degli account, la decisione finale è di negare la richiesta.



Supponiamo che Carlos si accorga allora del suo errore e cerchi di salvare il file nel bucket. `Production` AWS controlla innanzitutto l'account `111111111111` per determinare se la richiesta è consentita. Si applica solo la politica basata sull'identità e consente la richiesta. AWS quindi controlla l'account. `222222222222` Vale solo la policy basata sulle risorse collegata al bucket `Production` e consente la richiesta. Poiché entrambi gli account consentono la richiesta, la decisione finale è di consentire la richiesta.



Differenza tra rifiuto esplicito e implicito

Una richiesta genera un rifiuto esplicito se policy applicabile include un'istruzione `Deny`. Se le policy applicabili a una richiesta includono un'istruzione `Allow` e un'istruzione `Deny`, l'istruzione `Deny` prevale sull'istruzione `Allow`. La richiesta viene rifiutata esplicitamente.

Una rifiuto implicito si verifica quando non c'è un'istruzione `Deny` applicabile ma non c'è neanche un'istruzione `Allow` applicabile. Poiché a un principale IAM viene rifiutato l'accesso per impostazione predefinita, questo deve essere autorizzato esplicitamente a eseguire un'operazione. In caso contrario, l'accesso viene negato implicitamente.

Quando progetti una strategia di autorizzazione, devi creare policy con istruzioni `Allow` per consentire alle entità principali di eseguire richieste. Tuttavia, puoi scegliere qualsiasi combinazione di rifiuti espliciti e impliciti.

Ad esempio, è possibile creare la seguente policy che include operazioni consentite, operazioni rifiutate implicitamente e operazioni rifiutate esplicitamente. La dichiarazione `AllowGetList`

permette l'accesso in sola lettura alle operazioni IAM che iniziano con i prefissi `Get` e `List`. Tutte le altre azioni in IAM, come `iam:CreatePolicy`, sono rifiutate implicitamente. La dichiarazione `DenyReports` impedisce esplicitamente l'accesso ai report IAM impedendo l'accesso alle operazioni che includono il suffisso `Report`, come `iam:GetOrganizationsAccessReport`. Se qualcuno aggiunge un'altra policy a questo principale per concedere l'accesso ai report IAM, come `iam:GenerateCredentialReport`, le richieste relative ai report vengono ancora rifiutate a causa di questo rifiuto esplicito.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowGetList",
            "Effect": "Allow",
            "Action": [
                "iam:Get*",
                "iam:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DenyReports",
            "Effect": "Deny",
            "Action": "iam:*Report",
            "Resource": "*"
        }
    ]
}
```

Sintassi del linguaggio della policy JSON IAM

Questa pagina riporta una sintassi formale per il linguaggio utilizzato per creare le policy JSON in IAM. Presentiamo questa grammatica in modo che sia possibile comprendere come costruire e convalidare le policy.

Per esempi di policy, consultare i seguenti argomenti:

- Politiche e autorizzazioni in AWS Identity and Access Management

- Esempi di policy basate su identità IAM
- Politiche di esempio per lavorare nella console Amazon EC2 ed esempi di politiche per l'utilizzo della AWS CLI, della CLI di Amazon EC2 o di un AWS SDK nella Amazon EC2 User Guide.
- Esempi di policy del bucket e Esempi di policy utente nella Guida per l'utente di Amazon Simple Storage Service.

Per esempi di policy utilizzate in altri AWS servizi, consulta la documentazione relativa a tali servizi.

Argomenti

- Il linguaggio di policy e JSON
- Convenzioni utilizzate in questa sintassi
- Grammatica
- Note sulla sintassi delle policy

Il linguaggio di policy e JSON

Le policy sono espresse in JSON. Quando crei o modifichi una policy JSON, IAM può eseguire la convalida delle policy per facilitare la creazione di una policy efficace. IAM identificherà gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente le policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

In questo documento, non forniamo una descrizione completa di ciò che costituisce un JSON valido. Tuttavia, alcune regole JSON di base:

- È consentito spazio vuoto tra singole entità.
- I valori sono racchiusi tra virgolette. Le virgolette sono facoltative per valori numerici e booleani.
- Molti elementi (ad esempio `action_string_list` e `resource_string_list`) possono richiedere un array JSON come valore. Gli array possono richiedere uno o più valori. Se più di un valore è incluso, l'array è tra parentesi quadre (`[` e `]`) e delimitato da virgole, come nell'esempio seguente:

  ```
  "Action" : ["ec2:Describe*","ec2:List*"]
  ```

- I tipi di dati JSON di base (booleano, numero e stringa) sono definiti in RFC 7159.

Convenzioni utilizzate in questa sintassi

Le convenzioni seguenti vengono utilizzate in questa grammatica:

- I seguenti caratteri sono token JSON e sono inclusi nelle policy:

  ```
  { } [ ] " , :
  ```

- I seguenti caratteri sono caratteri speciali nella grammatica e non sono inclusi nelle policy:

  ```
  = < > ( ) |
  ```

- Se un elemento permette più valori, è indicato utilizzando valori ripetuti, un delimitatore di virgole e puntini di sospensione (`...`). Esempi:

  ```
  [<action_string>, <action_string>, ...]
  ```

  ```
  <principal_map> = { <principal_map_entry>, <principal_map_entry>, ... }
  ```

  Se più valori sono consentiti, è anche valido per includere un solo valore. Per un solo valore, la virgola finale deve essere omessa. Se l'elemento richiede un array (contrassegnato con [e]), ma solo un valore è incluso, le parentesi sono facoltative. Esempi:

  ```
  "Action": [<action_string>]
  ```

  ```
  "Action": <action_string>
  ```

- Un punto di domanda (?) in seguito a un elemento indica che l'elemento è facoltativo. Esempio:

  ```
  <version_block?>
  ```

  Tuttavia, assicurarsi di fare riferimento alle note che seguono l'inserzione sulla grammatica sugli elementi opzionali.

- Una linea verticale (|) tra elementi indica alternative. Nella grammatica, le parentesi definiscono la portata delle alternative. Esempio:

  ```
  ("Principal" | "NotPrincipal")
  ```

- Gli elementi che devono essere stringhe letterali vengono racchiusi tra virgolette ('"'). Esempio:

  ```
  <version_block> = "Version" : ("2008-10-17" | "2012-10-17" )
  ```

Per ulteriori note, consultare Note sulla sintassi delle policy in seguito all'inserzione sulla grammatica.

Grammatica

La seguente inserzione descrive il linguaggio grammaticale della policy. Per le convenzioni utilizzate nell'inserzione, consultare la sezione precedente. Per ulteriori informazioni, consultare le seguenti note:

> **Note**
>
> Questa grammatica descrive le policy contrassegnate con una versione di `2008-10-17` e `2012-10-17` . Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Diversamente, una versione della policy viene creata quando si apportano modifiche alla policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version. Per ulteriori informazioni sulle versioni di policy, consultare the section called "Controllo delle versioni delle policy IAM".

```
policy  = {
     <version_block?>,
     <id_block?>,
     <statement_block>
}

<version_block> = "Version" : ("2008-10-17"        | "2012-10-17"       )

<id_block> = "Id" : <policy_id_string>

<statement_block> = "Statement" : [ <statement>, <statement>, ... ]

<statement> = {
    <sid_block?>,
    <principal_block?>,
    <effect_block>,
    <action_block>,
    <resource_block>,
    <condition_block?>
}

<sid_block> = "Sid" : <sid_string>
```

```
<effect_block> = "Effect" : ("Allow" | "Deny")

<principal_block> = ("Principal" | "NotPrincipal") : ("*" | <principal_map>)

<principal_map> = { <principal_map_entry>, <principal_map_entry>, ... }

<principal_map_entry> = ("AWS" | "Federated" | "Service" | "CanonicalUser") :
    [<principal_id_string>, <principal_id_string>, ...]

<action_block> = ("Action" | "NotAction") :
    ("*" | <action_string> | [<action_string>, <action_string>, ...])

<resource_block> = ("Resource" | "NotResource") :
    : ("*" | <resource_string> | [<resource_string>, <resource_string>, ...])

<condition_block> = "Condition" : { <condition_map> }
<condition_map> = {
  <condition_type_string> : { <condition_key_string> : <condition_value_list> },
  <condition_type_string> : { <condition_key_string> : <condition_value_list> }, ...
}
<condition_value_list> = [<condition_value>, <condition_value>, ...]
<condition_value> = (<condition_value_string> | <condition_value_string> |
 <condition_value_string>)
```

Note sulla sintassi delle policy

- Una singola policy può contenere una gamma di istruzioni.
- Le policy hanno una dimensione massima tra 2048 e 10.240 caratteri, in base a quale entità la policy è collegata. Per ulteriori informazioni, consulta IAM e AWS STS quote. I calcoli delle dimensioni della policy non includono spazi vuoti.
- I singoli elementi non devono contenere più istanze della stessa chiave. Ad esempio, non è possibile includere il blocco `Effect` due volte nella stessa istruzione.
- I blocchi possono essere visualizzati in qualsiasi ordine. Ad esempio, `version_block` può seguire `id_block` in una policy. Analogamente, `effect_block`, `principal_block`, `action_block` può comparire in qualsiasi ordine all'interno di un'istruzione.
- `id_block` è facoltativo nelle policy basate su risorse. Non deve essere incluso nelle policy basate sulle identità.

- L'elemento `principal_block` è obbligatorio nelle policy basate su risorse (ad esempio, nelle policy del bucket Amazon S3) e nelle policy di attendibilità per i ruoli IAM. Non deve essere incluso nelle policy basate sulle identità.
- L'elemento `principal_map` nelle policy di bucket Amazon S3 può includere l'ID `CanonicalUser`. La maggior parte delle policy basate su risorse non supporta questa mappatura. Per ulteriori informazioni sull'utilizzo dell'ID utente canonico in una policy di bucket, consulta Specifica di un principale in una policy nella Guida per l'utente di Amazon Simple Storage Service.
- Ogni valore di stringa (`policy_id_string`, `sid_string`, `principal_id_string`, `action_string`, `resource_string`, `condition_type_string`, `condition_key_string` e la versione di stringa di `condition_value`) può avere propri valori consentiti specifici, restrizioni di lunghezza minima e massima e un formato interno necessario.

Note sui valori di stringa

Questa sezione fornisce ulteriori informazioni sui valori di stringa utilizzati in diversi elementi in una policy.

`action_string`

Consiste in un namespace di un servizio, due punti e il nome di un'azione. I nomi delle operazioni possono includere caratteri jolly. Esempi:

```
"Action":"ec2:StartInstances"

"Action":[
  "ec2:StartInstances",
  "ec2:StopInstances"
]

"Action":"cloudformation:*"

"Action":"*"

"Action":[
  "s3:Get*",
  "s3:List*"
]
```

`policy_id_string`

Fornisce un modo per includere informazioni sulla policy complessiva. Alcuni servizi, ad esempio Amazon SQS e Amazon SNS, utilizzano l'elemento `Id` in modi riservati. Salvo diversamente limitato da un singolo servizio, policy_id_string può includere spazi. Alcuni servizi richiedono che questo valore sia univoco in un account AWS .

> **Note**
>
> L'endpoint `id_block` è consentito nelle policy basate su risorse, ma non nelle policy basate sulle identità.

Non esiste alcun limite alla lunghezza, anche se questa stringa contribuisce alla lunghezza complessiva della policy, che è limitata.

```
"Id":"Admin_Policy"

"Id":"cd3ad3d9-2776-4ef1-a904-4c229d1642ee"
```

`sid_string`

Fornisce un modo per includere informazioni su una istruzione individuale. Per le policy IAM, i caratteri alfanumerici di base (A-Z, a-z,0-9) sono gli unici caratteri consentiti nel valore. `Sid` Altri servizi AWS che supportano policy basate su risorse possono avere altri requisiti per il valore `Sid`. Ad esempio, alcuni servizi richiedono che questo valore sia univoco all'interno di un Account AWS e alcuni servizi consentono caratteri aggiuntivi come spazi nel valore. `Sid`

```
"Sid":"1"

"Sid": "ThisStatementProvidesPermissionsForConsoleAccess"
```

`principal_id_string`

Fornisce un modo per specificare un principale utilizzando l'Amazon Resource Name (ARN) dell'utente IAM Account AWS, del ruolo IAM, dell'utente federato o dell'utente con ruolo presunto. Per un Account AWS, puoi anche utilizzare il modulo breve `AWS:accountnumber` anziché l'ARN completo. Per tutte le opzioni, tra cui ruoli assunti, servizi AWS e così via, consulta Come specificare un principale.

Nota che puoi usare * solo per specificare "everyone/anonymous.» Non è possibile utilizzarlo per specificare parte di un nome o di un ARN.

`resource_string`

Nella maggior parte dei casi, è composto da un Amazon Resource Name (ARN). È possibile utilizzare i caratteri jolly (* e ?) nella parte relativa alle risorse dell'ARN. Per ulteriori informazioni sull'utilizzo dei caratteri jolly negli ARN, consulta Utilizzo di caratteri jolly nei percorsi.

> **Note**
>
> Quando si specifica un ARN incompleto (uno con meno dei sei campi standard) in una policy basata sull'identità, completa AWS automaticamente l'ARN aggiungendo caratteri jolly (*) a tutti i campi mancanti. Ad esempio, specificare equivale `aarn:aws:sqs:*:*:*`, che consente `arn:aws:sqs` l'accesso a tutte le risorse Amazon Amazon SQS in tutte le regioni e gli account. Tuttavia, le policy di sessione trasmesse e AWS STS AssumeRole AssumeRoleWithWebIdentity le AssumeRoleWithSAML richieste non supportano gli ARN incompleti. L'utilizzo di un ARN incompleto in una policy di sessione genererà un `MalformedPolicyDocumentException` errore.

```
"Resource":"arn:aws:iam::123456789012:user/Bob"

"Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*"
```

`condition_type_string`

Identifica il tipo di condizione da testare, ad esempio `StringEquals`, `StringLike`, `NumericLessThan`, `DateGreaterThanEquals`, `Bool`, `BinaryEquals`, `IpAddress`, `ArnEquals` ecc. Per l'elenco completo dei tipi di condizione, consultare Elementi della policy JSON IAM: operatori di condizione.

```
"Condition": {
  "NumericLessThanEquals": {
    "s3:max-keys": "10"
  }
}

"Condition": {
  "Bool": {
```

```
    "aws:SecureTransport": "true"
  }
}

"Condition": {
  "StringEquals": {
      "s3:x-amz-server-side-encryption": "AES256"
   }
}
```

`condition_key_string`

Identifica la chiave della condizione il cui valore verrà testato per determinare se la condizione è soddisfatta. AWS definisce un set di chiavi di condizione disponibili in tutti i AWS servizi, inclusi `aws:PrincipalTypeaws:SecureTransport`, e`aws:userid`.

Per un elenco delle chiavi di AWS condizione, vedereAWS chiavi di contesto delle condizioni globali. Per le chiavi di condizione specifiche per un servizio, consultare la documentazione per quel servizio, tra cui quanto segue:

- Specifica delle condizioni in una policy nella Guida per l'utente di Amazon Simple Storage Service
- Policy di IAM per Amazon EC2 nella Guida per l'utente di Amazon EC2.

```
"Condition":{
  "Bool": {
      "aws:SecureTransport": "true"
   }
}

"Condition": {
  "StringNotEquals": {
      "s3:x-amz-server-side-encryption": "AES256"
   }
}

"Condition": {
  "StringEquals": {
    "aws:ResourceTag/purpose": "test"
  }
}
```

`condition_value_string`

Identifica il valore di condition_key_string che determina se la condizione è soddisfatta. Per un elenco completo di valori validi per un tipo di condizione, consulta Elementi della policy JSON IAM: operatori di condizione.

```
"Condition":{
  "ForAnyValue:StringEquals": {
  "dynamodb:Attributes": [
   "ID",
   "PostDateTime"
         ]
  }
}
```

AWS politiche gestite per le funzioni lavorative

Consigliamo di utilizzare le policy che concedono il privilegio minimo o che concedono solo le autorizzazioni richieste per eseguire un processo. Il modo più sicuro per concedere il privilegio minimo consiste nello scrivere una policy personalizzata con solo le autorizzazioni necessarie al team. È necessario creare un processo per consentire al team di richiedere ulteriori autorizzazioni quando necessario. La creazione di policy gestite dai clienti IAM che forniscono al team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza.

Per iniziare ad aggiungere autorizzazioni alle tue identità IAM (utenti, gruppi di utenti e ruoli), puoi usare. AWS policy gestite AWS le policy gestite coprono casi d'uso comuni e sono disponibili nel tuo. Account AWS AWS le politiche gestite non concedono autorizzazioni con privilegi minimi. Considera il rischio per la sicurezza di concedere ai principali più autorizzazioni di quelle necessarie per svolgere il proprio lavoro.

Puoi allegare policy AWS gestite, incluse le funzioni lavorative, a qualsiasi identità IAM. Per passare alle autorizzazioni con privilegi minimi, puoi eseguire AWS Identity and Access Management Access Analyzer per monitorare i principali con policy gestite. AWS Dopo aver appreso quali autorizzazioni stanno utilizzando, puoi scrivere una policy personalizzata o generare una policy con solo le autorizzazioni richieste per il team. È meno sicuro, ma offre maggiore flessibilità man mano che impari come il tuo team sta utilizzando. AWS

AWS le politiche gestite per le funzioni lavorative sono progettate per allinearsi strettamente alle funzioni lavorative più comuni nel settore IT. È possibile utilizzare queste policy per concedere le

autorizzazioni necessarie per eseguire le attività che ci si aspetta da qualcuno in una determinata funzione lavorativa. Queste policy consolidano le autorizzazioni per molti servizi in un'unica policy con la quale è più semplice collaborare rispetto ad avere le autorizzazioni disperse in molte policy.

Utilizzare i ruoli per combinare i servizi

Alcune policy utilizzano i ruoli dei servizi IAM per aiutarti a sfruttare le funzionalità presenti in altri AWS servizi. Queste policy concedono l'accesso `aiam:passrole`, il che consente a un utente con la policy di passare un ruolo a un AWS servizio. Questo ruolo delega le autorizzazioni IAM al AWS servizio per eseguire azioni per tuo conto.

È necessario creare ruoli in base alle proprie esigenze. Ad esempio, la policy Network Administrator consente a un utente con la policy di passare un ruolo denominato «flow-logs-vpc» al servizio Amazon. CloudWatch CloudWatch utilizza quel ruolo per registrare e acquisire il traffico IP per i VPC creati dall'utente.

Per seguire le best practice di sicurezza, le policy per le funzioni lavorative includono filtri che limitano i nomi dei ruoli validi che possono essere passati. In questo modo è possibile evitare di concedere autorizzazioni non necessarie. Se gli utenti richiedono i ruoli di servizio opzionali, è necessario creare un ruolo che segue la convenzione di denominazione specificata nella policy. È possibile concedere le autorizzazioni al ruolo. Una volta completata questa operazione, l'utente può configurare il servizio per utilizzare il ruolo, concedendogli tutte le autorizzazioni che il ruolo fornisce.

Nelle seguenti sezioni, ogni nome di policy è un collegamento alla pagina dei dettagli della policy nella Console di gestione AWS. Qui è possibile visualizzare il documento della policy e riconsultare le autorizzazioni che concede.

Funzione processo dell'amministratore

AWS nome della policy gestita: AdministratorAccess

Caso d'uso: questo utente ha accesso completo e può delegare le autorizzazioni per ogni servizio e risorsa in AWS.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede tutte le azioni per tutti i AWS servizi e per tutte le risorse dell'account. Per ulteriori informazioni sulla policy gestita, consulta AdministratorAccess la AWS Managed Policy Reference Guide.

> **Note**
>
> Prima che un utente o un ruolo IAM possa accedere alla Gestione costi e fatturazione AWS console con le autorizzazioni di questa policy, devi prima attivare l'accesso agli utenti e ai ruoli IAM. A tale scopo, seguire le istruzioni riportate in Concedere l'accesso alla console di fatturazione per delegare l'accesso alla console di fatturazione.

Funzione di processo di fatturazione

AWS nome della policy gestita: Fatturazione

Caso d'uso: questo utente deve visualizzare i dati di fatturazione, impostare i pagamenti e autorizzarli. L'utente può monitorare i costi accumulati per l'intero AWS servizio.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione policy: questa policy concede le autorizzazioni complete per gestire fatturazione, costi, metodi di pagamento, budget e report. Per ulteriori esempi di policy di gestione dei costi, consulta gli esempi di policy AWS Billing nella Guida per l'utente di Gestione costi e fatturazione AWS Per ulteriori informazioni sulla policy gestita, consulta Billing nella Guida di riferimento alle policy gestite da AWS .

> **Note**
>
> Prima che un utente o un ruolo IAM possa accedere alla Gestione costi e fatturazione AWS console con le autorizzazioni di questa policy, devi prima attivare l'accesso agli utenti e ai ruoli IAM. A tale scopo, seguire le istruzioni riportate in Concedere l'accesso alla console di fatturazione per delegare l'accesso alla console di fatturazione.

Funzione di processo dell'amministratore di database

AWS nome della policy gestita: DatabaseAdministrator

Caso d'uso: questo utente imposta, configura e gestisce i database nel AWS cloud.

Aggiornamenti delle policy: AWS mantiene e aggiorna questa policy. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione policy: questa policy concede le autorizzazioni per creare, configurare e gestire i database. Include l'accesso a servizi di AWS database, come Amazon DynamoDB, Amazon Relational Database Service (RDS) e Amazon Redshift. Visualizza la policy per l'elenco completo di servizi di database supportati dalla policy. Per ulteriori informazioni sulla policy gestita, consulta la Managed Policy Reference DatabaseAdministrator Guide AWS .

Questa politica relativa alle funzioni lavorative supporta la possibilità di trasferire ruoli ai AWS servizi. La policy consente l'operazione `iam:PassRole` solo per i ruoli denominati nella tabella seguente. Per ulteriori informazioni, consulta Creazione dei ruoli e collegamento delle policy (console) più avanti in questo argomento.

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	Seleziona questa politica AWS gestita
Consentire all'utente di monitorare i database RDS	rds-monitoring-role	Ruolo Amazon RDS per il monitoraggio avanzato	AmazonRDS EnhancedMonitoring Role
Consenti AWS Lambda di monitorare il tuo database e accedere a database esterni	rdbms-lambda-access	Amazon EC2	AWSLambda_FullAccess
Consentire a Lambda di caricare file su Amazon	lambda_exec_role	AWS Lambda	Creare una nuova policy gestita secondo quanto

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	Seleziona questa politica AWS gestita
S3 e su cluster Amazon Redshift con DynamoDB			definito in AWS Big Data Blog
Consentire alle funzioni Lambda di agire come trigger per le tabelle DynamoDB	lambda-dynamodb-*	AWS Lambda	AWSLambda DynamoDBExecutionRole
Consentire alle funzioni Lambda di accedere ad Amazon RDS in un VPC	lambda-vpc-execution-role	Creazione di un ruolo con una policy di attendibilità secondo quanto definito nella Guida per gli sviluppatori di AWS Lambda	AWSLambda VPCAccessExecutionRole
Permetti AWS Data Pipeline di accedere alle tue AWS risorse	DataPipelineDefaultRole	Creazione di un ruolo con una policy di attendibilità secondo quanto definito nella Guida per gli sviluppatori di AWS Data Pipeline	La AWS Data Pipeline documentazione elenca le autorizzazioni richieste per questo caso d'uso. Vedi i ruoli IAM per AWS Data Pipeline

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	Seleziona questa politica AWS gestita
Consentire alle applicazioni in esecuzione su istanze Amazon EC2 di accedere alle risorse AWS	DataPipelineDefaultResourceRole	Creazione di un ruolo con una policy di attendibilità secondo quanto definito nella Guida per gli sviluppatori di AWS Data Pipeline	AmazonEC2RoleforDataPipelineRole

Funzione di processo per data scientist

AWS nome della policy gestita: DataScientist

Caso d'uso: questo utente esegue attività e query Hadoop. L'utente, inoltre accede e analizza le informazioni per l'analisi dei dati e di business intelligence.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della policy: questa policy concede le autorizzazioni per creare, gestire ed eseguire query su un cluster Amazon EMR ed eseguire analisi dei dati con strumenti come Amazon. QuickSight La policy include l'accesso a servizi di data scientist aggiuntivi, come Amazon EC2 AWS Data Pipeline, Amazon Kinesis, Amazon Machine Learning e AI. SageMaker Visualizza la policy per l'elenco completo dei servizi scientifici dei dati supportati dalla policy. Per ulteriori informazioni sulla policy gestita, consulta DataScientist la Managed Policy Reference Guide. AWS

Questa politica relativa alle funzioni lavorative supporta la possibilità di trasferire ruoli ai AWS servizi. Una dichiarazione consente di passare qualsiasi ruolo all' SageMaker IA. Un'altra istruzione consente l'operazione `iam:PassRole` solo per i ruoli denominati nella tabella seguente. Per ulteriori informazioni, consulta Creazione dei ruoli e collegamento delle policy (console) più avanti in questo argomento.

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da seleziona re
Consentire alle istanze Amazon EC2 l'accesso ai servizi e alle risorse idonei per i cluster	EMR-EC2_D efaultRole	Amazon EMR per EC2	AmazonEla sticMapRe duceforEC2Role
Consentire ad Amazon EMR di accedere al servizio e alle risorse Amazon EC2 per i cluster	EMR_ DefaultRole	Amazon EMR	AmazonEMR ServicePolicy_v2
Consenti a Kinesis Managed Service per Apache Flink di accedere alle origini dati in streaming	kinesis-*	Creare un ruolo con una policy di affidabilità secondo quanto definito in AWS Big Data Blog.	Consultare AWS Big Data Blog, che delinea quattro possibili opzioni, a seconda del caso d'uso
Consenti l'accesso AWS Data Pipeline alle tue risorse AWS	DataPipelineDefaul tRole	Creazione di un ruolo con una policy di attendibi lità secondo quanto definito nella Guida per gli sviluppat ori di AWS Data Pipeline	La AWS Data Pipeline documenta zione elenca le autorizzazioni richieste per questo caso d'uso. Vedi i ruoli IAM per AWS Data Pipeline
Consentire alle applicazioni in esecuzione su istanze Amazon EC2 di accedere alle risorse AWS	DataPipelineDefaul tResourceRole	Creazione di un ruolo con una policy di attendibi lità secondo quanto definito nella Guida per gli sviluppat	AmazonEC2 RoleforDataPipelin eRole

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da seleziona re
		ori di AWS Data Pipeline	

Funzione di processo per l'utente avanzato sviluppatore

AWS nome della policy gestita: PowerUserAccess

Caso d'uso: questo utente esegue attività di sviluppo di applicazioni e può creare e configurare risorse e servizi che supportano lo sviluppo di applicazioni AWS consapevoli.

Aggiornamenti delle policy: AWS mantiene e aggiorna questa policy. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della politica: la prima dichiarazione di questa politica utilizza l'`NotAction`elemento per consentire tutte le azioni per tutti i AWS servizi e per tutte le risorse tranne AWS Identity and Access Management, AWS Organizations, e Gestione dell'account AWS. La seconda istruzione concede le autorizzazioni IAM per creare un ruolo collegato ai servizi. Questo è obbligatorio per alcuni servizi che devono accedere alle risorse di un altro servizio, ad esempio un bucket Amazon S3.

Concede inoltre AWS Organizations le autorizzazioni per visualizzare le informazioni sull'organizzazione dell'utente, tra cui l'account di gestione, l'email e le limitazioni dell'organizzazione. Sebbene questa policy limiti IAM AWS Organizations, consente all'utente di eseguire tutte le azioni di IAM Identity Center se IAM Identity Center è abilitato. Concede inoltre le autorizzazioni di gestione dell'account per visualizzare quali AWS regioni sono abilitate o disabilitate per l'account.

Funzione di processo per l'amministratore di rete

AWS nome della politica gestita: NetworkAdministrator

Caso d'uso: questo utente ha il compito di configurare e gestire le risorse AWS di rete.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni

di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della policy: questa policy concede le autorizzazioni per creare e mantenere risorse di rete in Auto Scaling, Amazon EC2, Route 53 AWS Direct Connect, Amazon, Elastic Load Balancing CloudFront, Amazon SNS AWS Elastic Beanstalk, CloudWatch Logs, Amazon S3 CloudWatch, IAM e Amazon Virtual Private Cloud. Per ulteriori informazioni sulla policy gestita, consulta la Managed Policy Reference Guide. NetworkAdministrator AWS

Questa funzione lavorativa richiede la capacità di trasferire ruoli ai AWS servizi. La policy concede `iam:GetRole` e `iam:PassRole` solo per quei ruoli denominati nella tabella seguente. Per ulteriori informazioni, consulta Creazione dei ruoli e collegamento delle policy (console) più avanti in questo argomento.

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da seleziona re
Consente ad Amazon VPC di creare e gestire i CloudWatc h log in Logs per conto dell'utente per monitorare il traffico IP in entrata e in uscita dal tuo VPC	flow-logs-*	Creazione di un ruolo con una policy di attendibi lità secondo quanto definito nella Guida per l'utente di Amazon VPC	Questo caso d'uso non prevede una policy AWS gestita esistente, ma la documentazione elenca le autorizza zioni richieste. Consulta la Guida per l'utente di Amazon VPC.

Read-only accesso

AWS nome della politica gestita: ReadOnlyAccess

Caso d'uso: questo utente richiede l'accesso in sola lettura a tutte le risorse in un Account AWS.

> **Important**
>
> Questo utente avrà anche accesso ai dati di lettura in servizi di archiviazione come i bucket Amazon S3 e le tabelle Amazon DynamoDB.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della policy: questa policy concede le autorizzazioni per elencare, ottenere, descrivere e visualizzare in altro modo le risorse e i relativi attributi. Non include funzioni mutanti come create o delete. Questa politica include l'accesso in sola lettura ai AWS servizi relativi alla sicurezza, come e. AWS Identity and Access Management Gestione costi e fatturazione AWS

Visualizza la policy per l'elenco completo di servizi e operazioni supportati dalla policy. Per ulteriori informazioni sulla policy gestita, consulta la Managed Policy Reference ReadOnlyAccess Guide AWS . Se hai bisogno di una policy simile che non conceda l'accesso ai dati di lettura nei servizi di archiviazione, consulta View-only funzione di lavoro dell'utente.

Accesso completo alle azioni del servizio MCP

AWS nome della politica gestita: AWSMcpServiceActionsFullAccess

Caso d'uso: questo utente richiede l'accesso ai AWS servizi tramite server AWS MCP. Questa politica non consente l'accesso alle azioni intraprese da un servizio MCP ad altri AWS servizi.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede le autorizzazioni per avviare qualsiasi azione del servizio AWS MCP. È possibile utilizzarla quando non è necessario specificare le autorizzazioni per servizio MCP. AWS Non concede autorizzazioni per le azioni intraprese dal servizio MCP ad altri AWS servizi, tali autorizzazioni devono sempre essere concesse separatamente e in aggiunta alle azioni del servizio MCP. Per ulteriori informazioni sulla policy gestita, vedere la Managed Policy Reference AWSMcpServiceActionsFullAccess Guide AWS .

Funzione di processo del revisore sicurezza

AWS nome della policy gestita: SecurityAudit

Caso d'uso: questo utente monitora gli account per la conformità ai requisiti di sicurezza. Questo utente può accedere ai log e agli eventi per analizzare potenziali violazioni alla sicurezza o potenziale attività non autorizzata.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede le autorizzazioni per visualizzare i dati di configurazione per molti AWS servizi e per esaminarne i registri. Per ulteriori informazioni sulla policy gestita, consulta SecurityAudit la Managed Policy Reference Guide. AWS

Funzione di processo dell'utente di Support

AWS nome della policy gestita: AWSSupportAccess

Caso d'uso: questo utente contatta l' AWS assistenza, crea casi di supporto e visualizza lo stato dei casi esistenti.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede le autorizzazioni per creare e aggiornare Supporto i casi. Per ulteriori informazioni sulla policy gestita, consulta AWSSupportAccess la Managed Policy Reference Guide. AWS

Funzione di processo dell'amministratore di sistema

AWS nome della policy gestita: SystemAdministrator

Caso d'uso: questo utente imposta e gestisce le risorse per le operazioni di sviluppo.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni

di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede le autorizzazioni per creare e mantenere risorse per un'ampia varietà di AWS servizi, tra cui Amazon AWS CloudTrail,, CloudWatch,, AWS CodeCommit, AWS CodeDeploy AWS Config, Amazon EC2 Servizio di directory AWS,,, AWS Identity and Access Management, Amazon RDS AWS Key Management Service AWS Lambda, Route 53, Amazon S3, Amazon SES, Amazon SQS e Amazon VPC. AWS Trusted Advisor Per ulteriori informazioni sulla policy gestita, consulta la Managed Policy Reference Guide. SystemAdministrator AWS

Questa funzione lavorativa richiede la capacità di trasferire ruoli ai AWS servizi. La policy concede `iam:GetRole` e `iam:PassRole` solo per quei ruoli denominati nella tabella seguente. Per ulteriori informazioni, consulta Creazione dei ruoli e collegamento delle policy (console) più avanti in questo argomento. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da seleziona re
Consentire alle applicazioni in esecuzione in istanze di EC2 in un cluster Amazon ECS di accedere ad Amazon ECS	ecr-sysadmin-*	Ruolo Amazon EC2 per EC2 Container Service	AmazonEC2 ContainerServicefo rEC2Role
Consentire a un utente di monitorare i database	rds-monitoring-role	Ruolo Amazon RDS per il monitoraggio avanzato	AmazonRDS EnhancedM onitoringRole
Consenti alle app in esecuzione nelle istanze EC2 di accedere AWS alle risorse.	ec2-sysadmin-*	Amazon EC2	Policy di esempio per il ruolo che concede l'accesso a un bucket S3 come illustrato nella Guida per l'utente di Amazon EC2;

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da selezionare
			personalizzare in base alle esigenze
Consenti a Lambda di leggere i flussi DynamoDB e scrivere nei log CloudWatch	lambda-sysadmin-*	AWS Lambda	AWSLambdaDynamoDBExecutionRole

View-only funzione di lavoro dell'utente

AWS nome della politica gestita: ViewOnlyAccess

Caso d'uso: questo utente può visualizzare un elenco di AWS risorse e metadati di base nell'account per tutti i servizi. L'utente non può leggere i contenuti o i metadati delle risorse che superano la quota ed elencare informazioni per le risorse.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti a AWS politiche gestite per le funzioni lavorative.

Descrizione della politica: questa politica garantisce`List*`, `Describe*` `Get*View*`, e `Lookup*` l'accesso alle risorse per AWS i servizi. Per vedere quali azioni include questa politica per ciascun servizio, consulta ViewOnlyAccess. Per ulteriori informazioni sulla policy gestita, consulta ViewOnlyAccess la AWS Managed Policy Reference Guide.

Aggiornamenti a AWS politiche gestite per le funzioni lavorative

Queste politiche sono tutte mantenute AWS e aggiornate per includere il supporto per nuovi servizi e nuove funzionalità man mano che vengono aggiunte dai AWS servizi. Queste policy non possono essere modificate dai clienti. È possibile creare una copia della policy e quindi modificarla, ma tale copia non viene aggiornata automaticamente poiché AWS introduce nuovi servizi e operazioni API.

Per una policy di funzione del processo, è possibile visualizzare la cronologia delle versioni e l'ora e la data di ogni aggiornamento nella console IAM. A tale scopo, utilizza i collegamenti presenti in

questa pagina per visualizzare i dettagli delle policy. Quindi scegli la scheda Versioni di policy per visualizzare le versioni. Questa pagina mostra le ultime 25 versioni di una policy. Per visualizzare tutte le versioni di una policy, chiama il AWS CLI comando get-policy-version o l'operazione API. GetPolicyVersion

> **Note**
>
> Puoi avere fino a cinque versioni di una policy gestita dal cliente, ma AWS conserva la cronologia completa delle versioni delle policy gestite. AWS

Creazione dei ruoli e collegamento delle policy (console)

Molte delle politiche elencate in precedenza consentono di configurare AWS i servizi con ruoli che consentono a tali servizi di eseguire operazioni per conto dell'utente. Le policy della funzione lavorativa specificano i nomi di ruolo esatti che è necessario utilizzare o almeno includono un prefisso che specifica la prima parte del nome che può essere utilizzato. Per creare uno di questi ruoli, eseguire le operazioni descritte nella procedura seguente.

Per creare un ruolo per un Servizio AWS (console IAM)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Tipo di entità attendibile, seleziona Servizio AWS.
4. Per Servizio o caso d'uso, scegli un servizio, quindi scegli il caso d'uso. I casi d'uso sono definiti dal servizio per includere la politica di fiducia richiesta dal servizio.
5. Scegli Next (Successivo).
6. Per Policy di autorizzazione, le opzioni dipendono dal caso d'uso selezionato:
   - Se il servizio definisce le autorizzazioni per il ruolo, le policy di autorizzazioni non possono essere selezionate.
   - Seleziona una policy da un set limitato di policy di autorizzazione.
   - Seleziona una policy tra tutte le policy di autorizzazione.
   - Non selezionare policy di autorizzazioni, crea le policy dopo la creazione del ruolo e quindi collegale al ruolo.

7. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una funzionalità avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.

   a. Apri la sezione Imposta limite delle autorizzazioni e seleziona Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo.

      IAM include un elenco delle politiche AWS gestite e gestite dal cliente nel tuo account.

   b. Selezionare la policy da utilizzare per il limite delle autorizzazioni.

8. Scegli Next (Successivo).

9. Per Nome del ruolo, le opzioni dipendono dal servizio:

   - Se il servizio definisce il nome del ruolo, non puoi modificarlo.
   - Se il servizio definisce un prefisso per il nome del ruolo, puoi inserire un suffisso facoltativo.
   - Se il servizio non definisce il nome del ruolo, puoi assegnare un nome al ruolo.

   > **Important**
   >
   > Quando assegni un nome a un ruolo, tieni presente quanto segue:
   >
   > - I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS e non possono essere resi univoci in base alle maiuscole e minuscole.
   >
   >   Ad esempio, non creare ruoli denominati **`PRODROLE`** e **`prodrole`**. Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole, tuttavia quando un nome di ruolo viene visualizzato ai clienti nella console, ad esempio durante il processo di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole.
   > - Non è possibile modificare il nome del ruolo dopo averlo creato, in quanto altre entità possono fare riferimento al ruolo.

10. (Facoltativo) In Descrizione, inserisci una descrizione per il ruolo.

11. (Facoltativo) Per modificare i casi d'uso e le autorizzazioni per il ruolo, in Fase 1: seleziona le entità attendibili o Fase 2: aggiungi autorizzazioni seleziona Modifica.

12. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, aggiungi i tag come coppie chiave-valore. Per ulteriori informazioni sull'uso dei tag in IAM, consulta Tag for AWS Identity and Access Management resources nella IAM User Guide.

13. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

Esempio 1: configurazione di un utente come amministratore di database (console)

Questo esempio illustra i passaggi necessari per configurare Alice, un utente IAM, come Amministratore del database. Utilizza le informazioni nella prima riga della tabella nella sezione e consenti all'utente di abilitare il monitoraggio Amazon RDS. Alleghi la DatabaseAdministrator policy all'utente IAM di Alice in modo che possa gestire i servizi di database Amazon. Questa policy, inoltre, consente ad Alice di passare un ruolo denominato `rds-monitoring-role` al servizio Amazon RDS, che consente al servizio di monitorare i database Amazon RDS per suo conto.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Policy e inserisci **`database`** nella casella di ricerca, quindi premi Invio.
3. Seleziona il pulsante di opzione relativo alla DatabaseAdministrator policy, scegli Azioni, quindi scegli Allega.
4. Nell'elenco di entità, seleziona Alice, quindi scegli Collega policy. Alice ora può AWS amministrare i database. Tuttavia, per consentire ad Alice di monitorare tali database, è necessario configurare il ruolo di servizio.
5. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
6. Seleziona il tipo di ruolo Servizio AWS , quindi scegli Amazon RDS.
7. Seleziona il caso d'uso Ruolo Amazon RDS per il monitoraggio avanzato.
8. Amazon RDS definisce le autorizzazioni per il ruolo. Selezionare Next: Review (Successivo: esamina) per continuare.
9. Il nome del ruolo deve essere uno di quelli specificati dalla DatabaseAdministrator policy che Alice ha ora. Uno di questi è **`rds-monitoring-role`**. Inseriscilo in Role name (Nome ruolo).
10. (Facoltativo) In Descrizione ruolo, immettere una descrizione per il nuovo ruolo.
11. Dopo aver revisionato i dettagli, selezionare Create role (Crea ruolo).
12. Alice ora può abilitare Monitoraggio avanzato RDS nella sezione Monitoraggio della console Amazon RDS. Ad esempio, può eseguire questa operazione quando crea un'istanza database, crea una replica di lettura o modifica un'istanza di database. Deve inserire il nome di ruolo che ha creato (rds-monitoring-role) nella casella Monitoring Role (Ruolo di monitoraggio) quando imposta Enable Enhanced Monitoring (Abilita monitoraggio avanzato) su Yes (Sì).

Esempio 2: configurazione di un utente come amministratore di rete (console)

Questo esempio illustra i passaggi necessari per configurare Jorge, un utente IAM, come Amministratore di rete. Utilizza le informazioni nella tabella in tale sezione per consentire a Jorge di monitorare il traffico IP in uscita e in entrata da VPC. Consente inoltre a Jorge di acquisire tali informazioni nei CloudWatch log di Logs. Si allega la NetworkAdministrator policy all'utente IAM di Jorge in modo che possa configurare le risorse di AWS rete. Inoltre, tale policy consente a Jorge di passare un ruolo il cui nome inizia con `flow-logs*` ad Amazon EC2 al momento della creazione del log di flusso. In questo scenario, diversamente dall'esempio 1, non è disponibile un tipo di ruolo di servizio predefinito, è necessario eseguire pochi passaggi in maniera diversa.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy e inserisci **`network`** nella casella di ricerca, quindi premi Invio.
3. Seleziona il pulsante di opzione accanto alla NetworkAdministrator policy, scegli Azioni, quindi scegli Allega.
4. Nell'elenco degli utenti, seleziona la casella di controllo accanto a Jorge e scegli Collega policy. Jorge può ora amministrare le risorse di AWS rete. Tuttavia, per consentire il monitoraggio di traffico IP nel VPC, è necessario configurare il ruolo del servizio.
5. Poiché il ruolo del servizio che bisogna creare non dispone di una policy gestita predefinita, è necessario prima crearlo. Nel riquadro di navigazione, selezionare Policies (Policy) e Create Policy (Crea policy).
6. Nella sezione Editor di policy, seleziona l'opzione JSON e copia il testo dal seguente documento della policy JSON. Incolla il testo nella casella di testo JSON.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
```

```
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

7. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

8. Nella pagina Verifica e crea, digita **`vpc-flow-logs-policy-for-service-role`** come nome della policy. Rivedi il campo Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy, quindi seleziona Crea policy per salvare il lavoro.

   La nuova policy appare nell'elenco delle policy gestite ed è pronta a collegare.

9. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
10. Seleziona il tipo di ruolo Servizio AWS , quindi scegli Amazon EC2.
11. Seleziona il caso d'uso Amazon EC2.
12. Nella pagina Attach permissions policies (Collega policy delle autorizzazioni), selezionare la policy creata in precedenza, vpc-flow-logs-policy-for-service-role, e selezionare Next: Review (Successivo: esamina).
13. Il nome del ruolo deve essere consentito dalla NetworkAdministrator politica che Jorge ha ora. Qualsiasi nome che inizia con `flow-logs-` è consentito. Per questo esempio, inserisci **`flow-logs-for-jorge`** come Role name (Nome del ruolo).
14. (Facoltativo) In Descrizione ruolo, immettere una descrizione per il nuovo ruolo.
15. Dopo aver revisionato i dettagli, selezionare Create role (Crea ruolo).
16. Ora è possibile configurare la policy attendibilità necessaria per questo scenario. Nella pagina Ruoli, scegli il ruolo flow-logs-for-jorge (il nome, non la casella di controllo). Nella pagina dei dettagli per il nuovo ruolo, selezionare la scheda Trust relationships (Relazioni di trust) e selezionare Edit trust relationship (Modifica relazione di trust).

17. Modificare la riga "Servizio" come segue, sostituendo la voce per `ec2.amazonaws.com`:

```
        "Service": "vpc-flow-logs.amazonaws.com"
```

18. Jorge può ora creare log di flusso da un VPC o una sottorete nella console Amazon EC2. Quando crei il log di flusso, specifica il ruolo flow-logs-for-jorge. Quel ruolo dispone delle autorizzazioni per creare il log e scriverci dati.

AWS chiavi di contesto delle condizioni globali

Quando un principale fa una richiesta a AWS, AWS raccoglie le informazioni sulla richiesta in un contesto di richiesta. ??? È possibile utilizzare l'elemento `Condition` di una policy JSON per confrontare le chiavi della richiesta con i valori chiave specificati nella policy. Le informazioni sulla richiesta vengono fornite da diverse origini, tra cui il principale che effettua la richiesta, la risorsa su cui viene effettuata la richiesta e i metadati relativi alla richiesta stessa.

Le chiavi di condizione globali possono essere utilizzate in tutti i servizi AWS . Sebbene queste chiavi di condizione possano essere utilizzate in tutte le policy, la chiave non è disponibile in tutti i contesti di richiesta. Ad esempio, la chiave di condizione `aws:SourceAccount` è disponibile solo quando la chiamata alla risorsa viene effettuata direttamente da un principale del servizio AWS. Per ulteriori informazioni sulle circostanze in cui una chiave globale è inclusa nel contesto della richiesta, consultare le informazioni sulla disponibilità di ogni chiave.

Alcuni singoli servizi creano le proprie chiavi di condizione che sono disponibili nel contesto della richiesta per altri servizi. Cross-service le chiavi di condizione sono un tipo di chiave di condizione globale che include un prefisso corrispondente al nome del servizio, ad esempio `ec2:` or`lambda:`, ma sono disponibili in altri servizi.

Service-specific le chiavi di condizione sono definite per l'uso con un singolo AWS servizio. Ad esempio, Amazon S3 consente di scrivere una policy con la chiave di condizione `s3:VersionId` per limitare l'accesso a una versione specifica di un oggetto Amazon S3. Questa chiave di condizione è unica per il servizio, il che significa che funziona solo con le richieste al servizio Amazon S3. Per le chiavi di condizione specifiche del servizio, consulta Azioni, risorse e Chiavi di condizione per AWS i servizi e scegli il servizio di cui desideri visualizzare le chiavi.

> **Note**
>
> Se si utilizzano chiavi di condizione disponibili solo in alcune circostanze, è possibile utilizzare IfExists le versioni degli operatori di condizione. Se le chiavi di condizione sono assenti da

un contesto di richiesta, la policy può non riuscire a effettuare la valutazione. Ad esempio, utilizzare il seguente blocco condizionale con gli operatori `...IfExists` per verificare se una richiesta proviene da uno specifico intervallo IP o da un determinato VPC. Se una o entrambe le chiavi non sono incluse nel contesto della richiesta, la condizione restituisce comunque `true`. I valori vengono controllati solo se la chiave specificata è inclusa nel contesto della richiesta. Per ulteriori informazioni su come viene valutata una policy quando una chiave non è presente per altri operatori, consulta Operatori di condizione.

```
"Condition": {
    "IpAddressIfExists": {"aws:SourceIp" : ["xxx"] },
    "StringEqualsIfExists" : {"aws:SourceVpc" : ["yyy"]}
}
```

Important

Per confrontare la condizione con un contesto di richiesta con più valori chiave, devi utilizzare gli operatori su set `ForAllValues` o `ForAnyValue`. Utilizza gli operatori di insieme solo con le chiavi della condizione multivalore. Non utilizzare operatori con chiavi di condizione a valore singolo. Per ulteriori informazioni, consulta Operatori dell'insieme per le chiavi di contesto multivalore.

Proprietà del principale	Proprietà di una sessione di ruolo	Proprietà della rete	Proprietà della risorsa	Proprietà della richiesta
era: PrincipalArn era: PrincipalAccount era: PrincipalOrgPaths	era: AssumedRoot era: FederatedProvider era: TokenIssueTime	era: SourceIp era: SourceVpc era: SourceVpcArn	era: ResourceAccount aws: ResourceOrgID era: ResourceOrgPaths	ha detto: CalledVia aws: CalledViaFirst era: CalledViaLast

Proprietà del principale	Proprietà di una sessione di ruolo	Proprietà della rete	Proprietà della risorsa	Proprietà della richiesta
aws: PrincipalOrg ID aws:PrincipalTag/ *tag-key* era: PrincipalIsAWSService era: PrincipalServiceName era: PrincipalServiceNamesList era: PrincipalType aws:userid aws:username	era: SignInSessionArn era: MultiFactorAuthAge era: MultiFactorAuthPresent era: ChatbotSourceArn era: Ec2InstanceSourceVpc era: Ec2InstanceSourcePrivateIPv4 era: SourceIdentity era: ViaCustomerDomain	era: SourceVpce era: VpceAccount aws: VpceOrgID era: VpceOrgPaths leggi: VpcSourceIp	era:ResourceTag/ *tag-key*	aws: AWSMCPCalledVia aws:ViaAWSService AWS: tramite il servizio AWS MCP era: CurrentTime era: EpochTime aws:Referer era: RequestedRegion era:RequestTag/ *tag-key* ha detto: TagKeys era: SecureTransport era: SourceAccount

Proprietà del principale	Proprietà di una sessione di ruolo	Proprietà della rete	Proprietà della risorsa	Proprietà della richiesta
	ec2: RoleDelivery ec2: SourceInstanceArn colla: RoleAssumedBy colla: CredentialIssuingService codebuild: BuildArn codebuild: ProjectArn lambda: SourceFunctionArn sms: SourceInstanceArn archivio di identità: UserId			leggi: SourceArn aws: ID SourceOrg leggi: SourceOrgPaths era: UserAgent era: IsMcpServiceAction

Chiavi di condizione sensibili

Le seguenti chiavi di condizione sono considerate sensibili. L'uso di caratteri jolly in queste chiavi di condizione non ha alcun caso d'uso valido, anche con una sottostringa del valore della chiave con un carattere jolly. Questo perché la jolly potrebbe far corrispondere la chiave di condizione a qualsiasi valore, il che potrebbe rappresentare un rischio per la sicurezza.

- era: PrincipalAccount
- aws: PrincipalOrg ID
- era: ResourceAccount
- aws: ResourceOrg ID
- era: SourceAccount
- aws: ID SourceOrg
- era: SourceVpc
- era: SourceVpce
- era: VpceAccount
- aws: VpceOrg ID

Proprietà del principale

Utilizza le chiavi di condizione seguenti per confrontare i dettagli dell'entità che effettua la richiesta con le proprietà del principale specificate nella policy. Per un elenco dei principali che possono effettuare richieste, consulta Come specificare un principale.

era: PrincipalArn

Utilizzare questa chiave per confrontare il nome della risorsa Amazon (ARN) del principale che ha effettuato la richiesta con l'ARN specificato nella policy. Per i ruoli IAM, il contesto della richiesta restituisce l'ARN del ruolo, non l'ARN dell'utente che ha assunto il ruolo.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta per tutte le richieste firmate. Le richieste anonime non includono questa chiave. È possibile specificare i seguenti tipi di principali in questa chiave di condizione:
  - Ruolo IAM
  - Utente IAM
  - AWS STS utente principale federato

  - Account AWS utente root
- Tipo di dati: ARN

  AWS consiglia di utilizzare gli operatori ARN anziché gli operatori di stringa quando si confrontano gli ARN.
- Tipo di valore: Single-valued
- Valori di esempio L'elenco seguente mostra il valore del contesto della richiesta restituito per diversi tipi di principali che è possibile specificare nella chiave di condizione `aws:PrincipalArn`:
  - Ruolo IAM- Il contesto della richiesta contiene il seguente valore per la chiave di condizione `aws:PrincipalArn`. Non specificare l'ARN della sessione del ruolo assunto come valore per questa chiave di condizione. Per ulteriori informazioni sul principale della sessione del ruolo assunto, consulta Principali della sessione come ruolo.

    ```
    arn:aws:iam::123456789012:role/role-name
    ```
  - Utente IAM- Il contesto della richiesta contiene il seguente valore per la chiave di condizione `aws:PrincipalArn`.

    ```
    arn:aws:iam::123456789012:user/user-name
    ```
  - AWS STS principi utente federati: il contesto della richiesta contiene il seguente valore per la chiave di condizione. `aws:PrincipalArn`

    ```
    arn:aws:sts::123456789012:federated-user/user-name
    ```
  - Account AWS utente root: il contesto della richiesta contiene il seguente valore per la chiave di condizione. `aws:PrincipalArn` Quando si specifica l'ARN dell'utente root come valore per la chiave di condizione `aws:PrincipalArn`, limita le autorizzazioni solo per l'utente root del Account AWS. Questo è diverso dallo specificare l'ARN dell'utente root nell'elemento principale di una policy basata sulle risorse, che delega l'autorità al Account AWS. Per ulteriori informazioni sulla specifica dell'ARN dell'utente root nell'elemento principale di una policy basata sulle risorse, consulta Account AWS principals.

    ```
    arn:aws:iam::123456789012:root
    ```

È possibile specificare l'ARN dell'utente root come valore per la chiave di condizione `aws:PrincipalArn` nelle politiche di controllo del AWS Organizations servizio (SCP). Le SPC

sono un tipo di policy dell'organizzazione utilizzate per gestire le autorizzazioni nell'organizzazione e riguardano solo gli account dei membri nell'organizzazione. Una SCP limita le autorizzazioni per i ruoli e gli utenti IAM e negli account membri, compreso l'utente root dell'account membro. Per ulteriori informazioni sull'effetto delle SCP sulle autorizzazioni, consulta Effetti delle SCP sulle autorizzazioni nella Guida per l'utente di AWS Organizations .

era: PrincipalAccount

Utilizzare questa chiave per confrontare l'account a cui appartiene il principale richiedente con l'identificatore dell'account specificato nella policy. Per le richieste anonime, il contesto della richiesta restituisce `anonymous`.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta per tutte le richieste, incluse le richieste anonime.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

Nell'esempio seguente, l'accesso è negato tranne che ai principali con il numero di account `123456789012`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAccessFromPrincipalNotInSpecificAccount",
      "Action": "service:*",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:service:us-east-1:111122223333:resource"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": [
            "123456789012"
          ]
        }
      }
    }
```

```
  ]
}
```

era: PrincipalOrgPaths

Usa questa chiave per confrontare il AWS Organizations percorso del principale che effettua la richiesta con il percorso indicato nella policy. Tale principale può essere un utente IAM, un ruolo IAM, un principale utente federato AWS STS o un Utente root dell'account AWS. In una policy, questa chiave di condizione garantisce che il richiedente sia un membro dell'account all'interno della radice dell'organizzazione specificata o delle unità organizzative (OU) in AWS Organizations. Un AWS Organizations percorso è una rappresentazione testuale della struttura di un' AWS Organizations entità. Per ulteriori informazioni sull'utilizzo e la comprensione dei percorsi, consultare Comprendi il AWS Organizations percorso dell'entità.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta solo se il principale è membro di un'organizzazione. Le richieste anonime non includono questa chiave.
- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

> **Note**
>
> Gli ID organizzazione sono univoci a livello globale, ma gli ID delle unità organizzative e gli ID root sono univoci solo all'interno di un'organizzazione. Ciò significa che non ci sono due organizzazioni che condividono lo stesso ID organizzazione. Tuttavia, un'altra organizzazione potrebbe avere un'unità organizzativa o un root con il tuo stesso ID. Si consiglia di includere sempre l'ID organizzazione quando si specifica un'unità organizzativa o un root.

Ad esempio, la seguente condizione restituisce `true` per le entità negli account collegati direttamente all'unità organizzativa `ou-ab12-22222222`, ma non nelle unità organizzative figlie.

```
"Condition" : { "ForAnyValue:StringEquals" : {
     "aws:PrincipalOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/"]
}}
```

La seguente condizione restituisce `true` per i principali in un account collegato direttamente all'unità organizzativa o a una delle sue unità organizzative figlie. Quando si include un carattere jolly, è necessario utilizzare l'operatore condizionale `StringLike`.

```
"Condition" : { "ForAnyValue:StringLike" : {
     "aws:PrincipalOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/
*"]
}}
```

La seguente condizione restituisce `true` per i principali in un account collegato direttamente a una delle unità organizzative figlie ma non all'unità organizzativa padre. La condizione precedente è per l'unità organizzativa o per qualsiasi figlio. La condizione seguente è solo per i figli (e tutti i figli di quei figli).

```
"Condition" : { "ForAnyValue:StringLike" : {
     "aws:PrincipalOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/
ou-*"]
}}
```

La condizione seguente consente l'accesso per ogni principale nell'organizzazione `o-a1b2c3d4e5`, indipendentemente dalla propria unità organizzativa padre.

```
"Condition" : { "ForAnyValue:StringLike" : {
     "aws:PrincipalOrgPaths":["o-a1b2c3d4e5/*"]
}}
```

`aws:PrincipalOrgPaths` è una chiave di condizione multivalore. Le chiavi multivalore possono avere più di un valore nel contesto della richiesta. Quando si utilizzano più valori con l'operatore condizionale `ForAnyValue`, il percorso del principale deve corrispondere a uno dei percorsi elencati nella policy. Per ulteriori informazioni sulle chiavi di condizione multivalore, consultare Operatori dell'insieme per le chiavi di contesto multivalore.

```
    "Condition": {
        "ForAnyValue:StringLike": {
            "aws:PrincipalOrgPaths": [
                "o-a1b2c3d4e5/r-ab12/ou-ab12-33333333/*",
                "o-a1b2c3d4e5/r-ab12/ou-ab12-22222222/*"
            ]
        }
```

```
    }
```

aws: PrincipalOrg ID

Utilizzate questa chiave per confrontare l'identificativo dell'organizzazione AWS Organizations a cui appartiene il principale richiedente con l'identificatore specificato nella politica.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta solo se il principale è membro di un'organizzazione. Le richieste anonime non includono questa chiave.
- Tipo di dati: stringa
- Tipo di valore: Single-valued

Questa chiave globale fornisce un'alternativa per elencare tutti gli ID account per tutti gli account AWS all'interno di un'organizzazione. È possibile utilizzare questa chiave di condizione per specificare semplicemente l'elemento `Principal` in una policy basata sulle risorse. È possibile specificare l'ID organizzazione nell'elemento condizionale. Quando si aggiunge e si rimuove un account, le policy che includono la chiave `aws:PrincipalOrgID` includono automaticamente anche gli account corretti e non necessitano di aggiornamento manuale.

Ad esempio, la seguente policy del bucket Amazon S3 consente ai membri di qualsiasi account nell'organizzazione `o-xxxxxxxxxxx` di aggiungere un oggetto al bucket `amzn-s3-demo-bucket`.

JSON

```
{
 "Version":"2012-10-17",
 "Statement": {
   "Sid": "AllowPutObject",
   "Effect": "Allow",
   "Principal": "*",
   "Action": "s3:PutObject",
   "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
   "Condition": {"StringEquals":
     {"aws:PrincipalOrgID":"o-xxxxxxxxxxx"}
   }
 }
}
```

> **Note**
>
> Questa condizione globale vale anche per l'account di gestione di un'organizzazione AWS . Questa policy impedisce a tutti principali esterni all'organizzazione specificata di accedere al bucket Amazon S3. Ciò include tutti AWS i servizi che interagiscono con le tue risorse interne, come AWS CloudTrail l'invio di dati di log ai tuoi bucket Amazon S3. Per scoprire come concedere in sicurezza l'accesso ai AWS servizi, consulta. era: PrincipalIsAWSService

Per ulteriori informazioni su AWS Organizations, vedi Che cos'è AWS Organizations? nella Guida AWS Organizations per l'utente.

aws:PrincipalTag/tag-key

Utilizzare questa chiave per confrontare il tag collegato al principale che effettua la richiesta con il tag specificato nella policy. Se il principale ha più di un tag collegato, il contesto della richiesta include una chiave `aws:PrincipalTag` per ogni chiave tag collegata.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta se il principale utilizza un utente IAM con tag collegati. È inclusa per un principale che utilizza un ruolo IAM con tag collegati o tag di sessione. Le richieste anonime non includono questa chiave.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

È possibile aggiungere attributi personalizzati a un utente o a un ruolo sotto forma di coppia chiave-valore. Per ulteriori informazioni sui tag in IAM, consulta Tag per AWS Identity and Access Management risorse. Puoi utilizzare `aws:PrincipalTag` per controllare l'accesso per i principali AWS .

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti con il tag **`department=hr`** per gestire utenti, gruppi o ruoli IAM. Per utilizzare questa politica, sostituisci la politica *`italicized placeholder text`* nell'esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:Get*",
        "iam:List*",
        "iam:Generate*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/department": "hr"
        }
      }
    }
  ]
}
```

era: PrincipalIsAWSService

Usa questo tasto per verificare se la chiamata alla tua risorsa viene effettuata direttamente da un responsabile AWS del servizio. Ad esempio, AWS CloudTrail utilizza il principale del servizio `cloudtrail.amazonaws.com` per scrivere log nel bucket Amazon S3. La chiave di contesto della richiesta è impostata su true (VERO) quando un servizio utilizza un principale del servizio per eseguire un'operazione diretta sulle risorse. La chiave di contesto è impostata su false se il servizio utilizza le credenziali di un principale IAM per effettuare una richiesta per conto del principale. Viene impostata su false anche se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale.

- Disponibilità: questa chiave è presente nel contesto della richiesta per tutte le richieste di API firmate che utilizzano le credenziali AWS . Le richieste anonime non includono questa chiave.
- Tipo di dati: booleano
- Tipo di valore: Single-valued

Puoi utilizzare questa chiave condizionale per limitare l'accesso alle tue identità attendibili e alle posizioni di rete previste, garantendo al contempo l'accesso ai AWS servizi in modo sicuro.

Nel seguente esempio di policy sui bucket di Amazon S3, l'accesso al bucket è limitato a meno che la richiesta non provenga `vpc-111bbb22` o provenga da un principale del servizio, ad esempio. CloudTrail

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ExpectedNetworkServicePrincipal",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/AWS Logs/AccountNumber/*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceVpc": "vpc-111bbb22"
        },
        "BoolIfExists": {
          "aws:PrincipalIsAWSService": "false"
        }
      }
    }
  ]
}
```

Nel seguente video, scopri ulteriori informazioni su come utilizzare la chiave di condizione `aws:PrincipalIsAWSService` in una policy.

Concedi in modo sicuro l'accesso agli utenti autorizzati, alle posizioni di rete previste e ai servizi insieme. AWS

era: PrincipalServiceName

Utilizza questa chiave per confrontare il nome del principale del servizio nella policy con il prinicpale del servizio che effettua richieste alle risorse. È possibile utilizzare questa chiave per verificare se la chiamata viene effettuata da un principale del servizio specifico. Quando un principale del servizio effettua una richiesta diretta alla risorsa, la chiave `aws:PrincipalServiceName` contiene

il nome del principale del servizio. Ad esempio, il nome principale del AWS CloudTrail servizio è`cloudtrail.amazonaws.com`.

- Disponibilità: questa chiave è presente nella richiesta quando la chiamata viene effettuata da un responsabile AWS del servizio. Questa chiave non è presente in alcun'altra situazione, tra cui:
  - Se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale.
  - Se il servizio utilizza le credenziali di un principale IAM per effettuare una richiesta per conto del principale.
  - Se la chiamata viene effettuata direttamente da un principale IAM.
  - Se la chiamata viene effettuata da un richiedente anonimo.
- Tipo di dati: stringa
- Tipo di valore: Single-valued

Puoi utilizzare questa chiave di condizione per limitare l'accesso alle tue identità attendibili e alle posizioni di rete previste, garantendo al contempo l'accesso a un AWS servizio in modo sicuro.

Nel seguente esempio di policy sui bucket di Amazon S3, l'accesso al bucket è limitato a meno che la richiesta non provenga `vpc-111bbb22` o provenga da un principale del servizio, ad esempio. CloudTrail

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ExpectedNetworkServicePrincipal",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/AWS Logs/AccountNumber/*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceVpc": "vpc-111bbb22",
          "aws:PrincipalServiceName": "cloudtrail.amazonaws.com"
        }
      }
```

```
    }
  ]
}
```

era: PrincipalServiceNamesList

Questa chiave fornisce un elenco di tutti i nomi dei principali del servizio che appartengono al servizio. Questa è una chiave di condizione avanzata. È possibile utilizzarlo per impedire al servizio di accedere alla risorsa solo da una regione specifica. Alcuni servizi potrebbero creare committenti di servizi regionali per indicare una particolare istanza del servizio all'interno di una regione specifica. È possibile limitare l'accesso a una risorsa a una particolare istanza del servizio. Quando un principale del servizio effettua una richiesta diretta alla risorsa, `aws:PrincipalServiceNamesList` contiene un elenco non ordinato di tutti i nomi di principali del servizio associati all'istanza regionale del servizio.

- Disponibilità: questa chiave è presente nella richiesta quando la chiamata viene effettuata da un responsabile AWS del servizio. Questa chiave non è presente in alcun'altra situazione, tra cui:
  - Se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale.
  - Se il servizio utilizza le credenziali di un principale IAM per effettuare una richiesta per conto del principale.
  - Se la chiamata viene effettuata direttamente da un principale IAM.
  - Se la chiamata viene effettuata da un richiedente anonimo.
- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

`aws:PrincipalServiceNamesList` è una chiave di condizione multivalore. Le chiavi multivalore possono avere più di un valore nel contesto della richiesta. È necessario utilizzare gli operatori di insieme `ForAnyValue` o `ForAllValues` con gli operatori di condizione di stringa quando si utilizza questa chiave. Per ulteriori informazioni sulle chiavi di condizione multivalore, consultare Operatori dell'insieme per le chiavi di contesto multivalore.

era: PrincipalType

Utilizzare questa chiave per confrontare il tipo di principale che effettua la richiesta con il tipo di principale specificato nella policy. Per ulteriori informazioni, consulta Come specificare un principale. Per esempi specifici di valori chiave `principal`, vedi Valori della chiave dell'entità principale.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta per tutte le richieste, incluse le richieste anonime.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

aws:userid

Utilizzare questa chiave per confrontare l'identificatore del principale richiedente con l'ID specificato nella policy. Per gli utenti IAM, il valore del contesto della richiesta è l'ID utente. Per i ruoli IAM, questo formato di valore può variare. Per informazioni dettagliate su come vengono visualizzate le informazioni per diverse entità, consultare Come specificare un principale. Per esempi specifici di valori chiave `principal`, vedi Valori della chiave dell'entità principale.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta per tutte le richieste, incluse le richieste anonime.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

aws:username

Utilizzare questa chiave per confrontare il nome utente del richiedente con il nome utente specificato nella policy. Per informazioni dettagliate su come vengono visualizzate le informazioni per diverse entità, consultare Come specificare un principale. Per esempi specifici di valori chiave `principal`, vedi Valori della chiave dell'entità principale.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta per gli utenti IAM. Le richieste anonime e le richieste effettuate utilizzando i ruoli Utente root dell'account AWS o IAM non includono questa chiave. Le richieste effettuate utilizzando le credenziali di IAM Identity Center non includono questa chiave nel contesto.
- Tipo di dati: stringa
- Tipo di valore: Single-valued

Proprietà di una sessione di ruolo

Utilizza le seguenti chiavi di condizione per confrontare le proprietà della sessione di ruolo al momento della generazione della sessione. Queste chiavi di condizione sono disponibili solo quando una richiesta viene effettuata da un principale con credenziali di sessione di ruolo o di principale utente federato. I valori di queste chiavi di condizione sono incorporati nel token di sessione del ruolo.

Un ruolo è un tipo di principale. È inoltre possibile utilizzare le chiavi di condizione della sezione Proprietà del principale per valutare le proprietà di un ruolo quando un ruolo effettua una richiesta.

era: AssumedRoot

Usa questa chiave per verificare se la richiesta è stata effettuata utilizzando AssumeRoot. `AssumeRoot`restituisce le credenziali a breve termine per una sessione utente root privilegiata che puoi utilizzare per eseguire azioni privilegiate sugli account dei membri della tua organizzazione. Per ulteriori informazioni, consulta Gestire centralmente l'accesso root per gli account membri.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando il principale utilizza le credenziali di AssumeRoot per effettuare la richiesta.
- Tipo di dati: booleano
- Tipo di valore: Single-valued

Nell'esempio seguente, se utilizzato come politica di controllo del servizio, nega l'utilizzo delle credenziali a lungo termine di un utente root in un account AWS Organizations membro. La policy non impedisce alle sessioni `AssumeRoot` di eseguire le operazioni consentite da una sessione `AssumeRoot`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
       {
          "Effect":"Deny",
          "Action":"*",
          "Resource": "*",
          "Condition":{
             "ArnLike":{
                "aws:PrincipalArn":[
```

```
                "arn:aws:iam::*:root"
            ]
         },
         "Null":{
            "aws:AssumedRoot":"true"
         }
      }
   }
]
}
```

era: FederatedProvider

Utilizzare questa chiave per confrontare il provider dell'identità di emissione (IdP) del principale con l'IdP specificato nella policy. Ciò significa che è stato assunto un ruolo IAM utilizzando l'`AssumeRoleWithWebIdentity` AWS STS operazione. Quando le credenziali temporanee della sessione come ruolo risultante vengono utilizzate per effettuare una richiesta, il contesto della richiesta identifica l'IdP che ha autenticato l'identità federata originale.

- Disponibilità: questa chiave è presente nella sessione di ruolo di un ruolo assunto utilizzando il provider OpenID Connect (OIDC) e nella policy di attendibilità dei ruoli quando viene utilizzato un provider OIDC per effettuare chiamate `AssumeRoleWithWebIdentity`.
- Tipo di dati: stringa*
- Tipo di valore: Single-valued

* Il tipo di dati dipende dal tuo IdP:

- Se utilizzi un AWS IdP integrato, come Amazon Cognito, il valore chiave sarà una stringa. Il valore chiave potrebbe essere simile a:. `cognito-identity.amazonaws.com`
- Se utilizzi un IdP non integrato in Amazon EKS, il valore chiave sarà ARN. AWSGitHub Il valore chiave potrebbe essere simile a:. `arn:aws:iam::`*`111122223333`*`:oidc-provider/oidc.eks.`*`region`*`.amazonaws.com/id/`*`OIDC_Provider_ID`*

Per ulteriori informazioni sulla rete esterna IdPs `AssumeRoleWithWebIdentity`, vedereScenari comuni. Per ulteriori informazioni, consulta Principali della sessione come ruolo.

era: TokenIssueTime

Utilizzare questa chiave per confrontare la data e l'ora in cui sono state emesse le credenziali di sicurezza temporanee con la data e l'ora specificate nella policy.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando il principale utilizza credenziali temporanee per effettuare la richiesta. La chiave non è presente nelle AWS CLI richieste AWS API o AWS SDK effettuate utilizzando le chiavi di accesso.
- Tipo di dati: data
- Tipo di valore: Single-valued

Per sapere quali servizi supportano l'utilizzo di credenziali temporanee, consulta AWS servizi supportati da IAM.

era: SignInSessionArn

Usa questa chiave per confrontare la tua AWS Sign-In OAuth sessione con la sessione specificata nella policy.

Quando utilizzi un flusso OAuth basato come AWS CLI login (`aws login`) o AWS MCP Server, AWS Sign-In include un ARN della sessione di accesso nelle credenziali emesse. AWS Sign-In propaga tale ARN alle richieste successive effettuate utilizzando i token di accesso. OAuth Con questa chiave, puoi correlare l'attività dell'API con la sessione di accesso originaria e applicare le policy IAM alle singole sessioni.

Per i OAuth flussi interattivi, il token di aggiornamento determina il ciclo di vita dell'ARN della sessione di accesso. Un nuovo token di accesso emesso durante un aggiornamento continua a contenere lo stesso ARN della sessione di accesso.

Per i OAuth flussi non interattivi, il token di accesso determina il ciclo di vita dell'ARN della sessione di accesso. Ogni nuovo token di accesso contiene un nuovo ARN della sessione di accesso.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando la richiesta proviene da una sessione. AWS Sign-In OAuth Questa chiave non è disponibile per le sessioni della console.
- Tipo di dati: ARN
- Tipo di valore: Single-valued

Utilizzate gli operatori ARN anziché gli operatori di stringa quando confrontate gli ARN.

La seguente politica nega tutte le azioni associate a una AWS Sign-In sessione specifica, sebbene le richieste provenienti da altre sessioni attive continuino:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenySpecificSignInSession",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:SignInSessionArn": "arn:aws:signin:us-east-1:111122223333:session/
session-id"
        }
      }
    }
  ]
}
```

era: MultiFactorAuthAge

Utilizzare questa chiave per confrontare il numero di secondi da quando il principale richiedente è stato autorizzato utilizzando MFA con il numero specificato nella policy. Per ulteriori informazioni sulla funzionalità MFA, consultare AWS Multi-factor autenticazione in IAM.

> **Important**
>
> Questa chiave di condizione non è presente per le identità federate o le richieste effettuate utilizzando chiavi di accesso per firmare richieste AWS CLI, AWS API o AWS SDK. Per ulteriori informazioni sull'aggiunta della protezione MFA alle operazioni API con credenziali di sicurezza temporanee, consulta Accesso sicuro alle API con MFA.
> Per verificare se l'MFA viene utilizzato per convalidare le identità federate IAM, puoi passare il metodo di autenticazione dal tuo provider di identità a come tag di sessione. AWS Per informazioni dettagliate, vedi Passa i tag della sessione in AWS STS. Per applicare la MFA per le identità del Centro identità IAM, puoi abilitare gli attributi per il controllo degli accessi per passare una dichiarazione di asserzione SAML con il metodo di autenticazione dal tuo provider di identità al Centro identità IAM.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando il principale utilizza credenziali temporanee per effettuare la richiesta. Le policy con condizioni MFA possono essere collegate a:
  - Un utente o un gruppo IAM
  - Una risorsa, ad esempio un bucket Amazon S3, una coda Amazon SQS o un argomento Amazon SNS
  - La policy di attendibilità di un ruolo IAM che può essere assunto da un utente
- Tipo di dati: numerico
- Tipo di valore: Single-valued

era: MultiFactorAuthPresent

Utilizzare questa chiave per verificare se l'autenticazione a più fattori (MFA) è stata utilizzata per convalidare le credenziali di sicurezza temporanee con le quali è stata effettuata la richiesta.

> **Important**
>
> Questa chiave di condizione non è presente per le identità federate o le richieste effettuate utilizzando chiavi di accesso per firmare richieste AWS CLI, AWS API o AWS SDK. Per ulteriori informazioni sull'aggiunta della protezione MFA alle operazioni API con credenziali di sicurezza temporanee, consulta Accesso sicuro alle API con MFA.
> Per verificare se l'MFA viene utilizzato per convalidare le identità federate IAM, puoi passare il metodo di autenticazione dal tuo provider di identità a come tag di sessione. AWS Per informazioni dettagliate, vedi Passa i tag della sessione in AWS STS. Per applicare la MFA per le identità del Centro identità IAM, puoi abilitare gli attributi per il controllo degli accessi per passare una dichiarazione di asserzione SAML con il metodo di autenticazione dal tuo provider di identità al Centro identità IAM.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando il principale utilizza credenziali temporanee per effettuare la richiesta. Le policy con condizioni MFA possono essere collegate a:
  - Un utente o un gruppo IAM
  - Una risorsa, ad esempio un bucket Amazon S3, una coda Amazon SQS o un argomento Amazon SNS
  - La policy di attendibilità di un ruolo IAM che può essere assunto da un utente

- Tipo di dati: booleano
- Tipo di valore: Single-valued

Le credenziali temporanee vengono utilizzate per autenticare i ruoli IAM e gli utenti IAM con token temporanei di AssumeRole o e GetSessionToken gli utenti di. Console di gestione AWS

Le chiavi di accesso utente IAM sono credenziali a lungo termine, ma in alcuni casi AWS creano credenziali temporanee per conto degli utenti IAM per eseguire operazioni. In questi casi, la chiave `aws:MultiFactorAuthPresent` è presente nella richiesta ed è impostata su un valore `false`. Esistono due casi comuni in cui ciò può accadere:

- Gli utenti IAM utilizzano Console di gestione AWS inconsapevolmente credenziali temporanee. Gli utenti accedono alla console utilizzando il nome utente e la password, che sono credenziali a lungo termine. Tuttavia, in background, la console genera credenziali temporanee per conto dell'utente.
- Se un utente IAM effettua una chiamata a un AWS servizio, il servizio riutilizza le credenziali dell'utente per effettuare un'altra richiesta a un servizio diverso. Ad esempio, quando si chiama Athena per accedere a un bucket Amazon S3 o quando si utilizza CloudFormation per creare un'istanza Amazon EC2. Per la richiesta successiva, utilizza credenziali temporanee AWS .

Per sapere quali servizi supportano l'utilizzo di credenziali temporanee, consulta AWS servizi supportati da IAM.

La chiave `aws:MultiFactorAuthPresent` non è presente quando vengono lanciati un'API o un comando della CLI con credenziali a lungo termine, ad esempio coppie di chiavi di accesso. Pertanto, si consiglia, quando si controlla questa chiave, di utilizzare le versioni `...IfExists` degli operatori di condizione.

È importante capire che il seguente elemento `Condition` non è un modo affidabile per controllare se una richiesta è autenticata utilizzando MFA.

```
#####   WARNING: NOT RECOMMENDED   #####
"Effect" : "Deny",
"Condition" : { "Bool" : { "aws:MultiFactorAuthPresent" : "false" } }
```

Questa combinazione di effetto `Deny`, elemento `Bool` e valore `false` nega le richieste che possono essere autenticate utilizzando MFA, ma che non lo sono. Ciò è valido solo per le credenziali temporanee che supportano l'uso di MFA. Questa istruzione non nega l'accesso alle richieste

effettuate utilizzando le credenziali a lungo termine oppure alle richieste autenticate utilizzando MFA. Usa questo esempio con cautela perché la sua logica è complicata e non verifica se MFA-authentication è stato effettivamente utilizzato.

Inoltre, non utilizzare la combinazione di effetto `Deny`, elemento `Null` e `true` perché ha lo stesso comportamento e la logica è ancora più complessa.

Combinazione consigliata

Consigliamo di utilizzare l'operatore `BoolIfExists` per verificare se una richiesta viene autenticata con MFA.

```
"Effect" : "Deny",
"Condition" : { "BoolIfExists" : { "aws:MultiFactorAuthPresent" : "false" } }
```

Questa combinazione di `Deny`, `BoolIfExists` e `false` nega le richieste che non vengono autenticate con MFA. Nello specifico, nega le richieste da credenziali temporanee che non includono MFA. Inoltre, nega le richieste effettuate utilizzando credenziali a lungo termine, ad esempio le AWS CLI operazioni AWS API effettuate utilizzando le chiavi di accesso. L'operatore `*IfExists` verifica la presenza della chiave `aws:MultiFactorAuthPresent` e se potrebbe essere presente o meno, come indicato dalla relativa esistenza. Utilizzalo quando intendi negare qualsiasi richiesta non autenticata con MFA. È più sicuro, ma può violare qualsiasi codice o script che utilizza chiavi di accesso per accedere all' AWS CLI API or. AWS

Combinazioni alternative

Puoi anche utilizzare l'`BoolIfExists`operatore per consentire MFA-authenticated richieste AWS CLI e/o richieste AWS API effettuate utilizzando credenziali a lungo termine.

```
"Effect" : "Allow",
"Condition" : { "BoolIfExists" : { "aws:MultiFactorAuthPresent" : "true" } }
```

Questa condizione è corrispondente se la chiave esiste ed è presente o se la chiave non esiste. Questa combinazione di `Allow`, `BoolIfExists` e `true` consente le richieste autenticate con MFA o le richieste che non possono essere autenticate con MFA. Ciò significa che le AWS CLI operazioni AWS API e AWS SDK sono consentite quando il richiedente utilizza le proprie chiavi di accesso a lungo termine. Questa combinazione non consente le richieste da credenziali temporanee che potrebbero ma non includono MFA.

Quando crei una policy utilizzando l'editor visivo della console IAM e scegli MFA obbligatoria, questa combinazione viene applicata. Questa impostazione richiede MFA per l'accesso alla console, ma consente l'accesso programmatico senza MFA.

In alternativa, puoi utilizzare l'operatore `Bool` per consentire le richieste programmatiche e della console solo quando autenticate tramite MFA.

```
"Effect" : "Allow",
"Condition" : { "Bool" : { "aws:MultiFactorAuthPresent" : "true" } }
```

Questa combinazione di `Allow`, `Bool`, e `true` consente solo MFA-authenticated le richieste. Ciò è valido solo per le credenziali temporanee che supportano l'uso di MFA. Questa istruzione non consente l'accesso alle richieste eseguite utilizzando chiavi di accesso a lungo termine oppure alle richieste eseguite utilizzando credenziali temporanee senza MFA.

Non utilizzare una struttura di policy simile alle seguenti per controllare se la chiave MFA è presente:

```
#####   WARNING: USE WITH CAUTION   #####

"Effect" : "Allow",
"Condition" : { "Null" : { "aws:MultiFactorAuthPresent" : "false" } }
```

Questa combinazione di effetto `Allow`, elemento `Null` e valore `false` consente solo le richieste che possono essere autenticate mediante MFA, indipendentemente dall'autenticazione o meno della richiesta. Ciò consente tutte le richieste eseguite utilizzando credenziali temporanee e nega l'accesso alle credenziali a lungo termine. Utilizzate questo esempio con cautela perché non verifica se MFA-authentication è stato effettivamente utilizzato.

era: ChatbotSourceArn

Utilizzare questa chiave per confrontare l'ARN della configurazione della chat di origine impostato dal principale con l'ARN di configurazione della chat specificato nella policy del ruolo IAM associato alla configurazione del canale. Puoi autorizzare le richieste in base alla sessione di assunzione del ruolo iniziata da Amazon Q Developer nelle applicazioni di chat.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta dal servizio Amazon Q Developer nelle applicazioni di chat ogni volta che si assume una sessione di ruolo. Il valore della chiave è l'ARN di configurazione della chat, ad esempio quando si esegue un comando AWS CLI da un canale di chat.

- Tipo di dati: ARN
- Tipo di valore — Single-valued
- Valore di esempio: `arn:aws::chatbot::123456789021:chat-configuration/slack-channel/private_channel`

La seguente policy nega che Amazon S3 inserisca le richieste nel bucket specificato per tutte le richieste provenienti da un canale Slack.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ExampleS3Deny",
            "Effect": "Deny",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "ArnLike": {
                      "aws:ChatbotSourceArn": "arn:aws:chatbot::*:chat-
configuration/slack-channel/*"
                }
            }
        }
    ]
}
```

era: Ec2InstanceSourceVpc

Questa chiave identifica il VPC al quale sono state consegnate le credenziali del ruolo IAM di Amazon EC2. È possibile utilizzare questa chiave in una policy con la chiave globale `aws:SourceVPC` per verificare se da un VPC (`aws:SourceVPC`) viene effettuata una chiamata che corrisponde al VPC al quale è stata consegnata una credenziale (`aws:Ec2InstanceSourceVpc`).

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che il richiedente firma le richieste con una credenziale di ruolo Amazon EC2. Può essere utilizzata nelle policy IAM, nelle policy di controllo dei servizi, nelle policy degli endpoint VPC e nelle policy delle risorse.

- Tipo di dati: stringa
- Tipo di valore — Single-valued

Questa chiave può essere utilizzata con i valori identificativi del VPC, ma è particolarmente utile se impiegata come variabile in combinazione con la chiave di contesto `aws:SourceVpc`. La chiave di contesto `aws:SourceVpc` viene inclusa nel contesto della richiesta solo se il richiedente utilizza un endpoint VPC per effettuare la richiesta. L'impiego di `aws:Ec2InstanceSourceVpc` con `aws:SourceVpc` consente di utilizzare `aws:Ec2InstanceSourceVpc` in modo più ampio, poiché confronta dei valori che in genere cambiano insieme.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "RequireSameVPC",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
            "aws:SourceVpc": "${aws:Ec2InstanceSourceVpc}"
        },
        "Null": {
          "ec2:SourceInstanceARN": "false"
        },
        "BoolIfExists": {
          "aws:ViaAWSService": "false"
        }
      }
    }
  ]
}
```

Nell'esempio precedente, l'accesso è negato se il valore di `aws:SourceVpc` non corrisponde al valore di `aws:Ec2InstanceSourceVpc`. L'istruzione della policy è limitata ai soli ruoli utilizzati come ruoli dell'istanza Amazon EC2 verificando l'esistenza della chiave di condizione `ec2:SourceInstanceARN`.

La policy consente `aws:ViaAWSService` di AWS autorizzare le richieste quando le richieste vengono effettuate per conto dei ruoli delle tue istanze Amazon EC2. Ad esempio, quando effettui una richiesta da un'istanza Amazon EC2 a un bucket Amazon S3 crittografato, Amazon S3 effettua una chiamata per tuo conto. AWS KMS Alcune chiavi non sono presenti quando viene effettuata la richiesta a. AWS KMS

era: Ec2InstanceSourcePrivateIPv4

Questa chiave identifica l'indirizzo IPv4 privato dell'interfaccia di rete elastica primaria a cui sono state consegnate le credenziali del ruolo IAM di Amazon EC2. È necessario utilizzare questa chiave di condizione con la chiave `aws:Ec2InstanceSourceVpc` complementare per assicurarsi di disporre di una combinazione univoca a livello globale di ID VPC e IP privato di origine. Usa questa chiave `aws:Ec2InstanceSourceVpc` per assicurarti che sia stata effettuata una richiesta dallo stesso indirizzo IP privato a cui sono state consegnate le credenziali.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che il richiedente firma le richieste con una credenziale di ruolo Amazon EC2. Può essere utilizzata nelle policy IAM, nelle policy di controllo dei servizi, nelle policy degli endpoint VPC e nelle policy delle risorse.
- Tipo di dati: indirizzo IP
- Tipo di valore: Single-valued

> **Important**
>
> Questa chiave non deve essere utilizzata da sola in un'istruzione `Allow`. Per definizione, gli indirizzi IP privati non sono univoci a livello globale. Dovresti usare la chiave `aws:Ec2InstanceSourceVpc` ogni volta che utilizzi la chiave `aws:Ec2InstanceSourcePrivateIPv4` per specificare il VPC dal quale possono essere utilizzate le credenziali dell'istanza Amazon EC2.

L'esempio seguente è una policy di controllo del servizio (SCP) che nega l'accesso a tutte le risorse a meno che la richiesta non arrivi tramite un endpoint VPC nello stesso VPC delle credenziali del ruolo. In questo esempio, `aws:Ec2InstanceSourcePrivateIPv4` limita l'origine delle credenziali a una particolare istanza in base all'IP di origine.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action":  "*",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:Ec2InstanceSourceVpc": "${aws:SourceVpc}"
                },
                "Null": {
                    "ec2:SourceInstanceARN": "false"
                },
                "BoolIfExists": {
                    "aws:ViaAWSService": "false"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action":  "*",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:Ec2InstanceSourcePrivateIPv4": "${aws:VpcSourceIp}"
                },
                "Null": {
                    "ec2:SourceInstanceARN": "false"
                },
                "BoolIfExists": {
                    "aws:ViaAWSService": "false"
                }
            }
        }
    ]
}
```

era: SourceIdentity

Utilizza questa chiave per confrontare l'identità di origine impostata dal principale con l'identità di origine specificata nella policy.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta dopo che è stata impostata un'identità di origine quando si assume un ruolo utilizzando qualsiasi comando CLI AWS STS assume-role o operazione API. AWS STS `AssumeRole`
- Tipo di dati: stringa
- Tipo di valore: Single-valued

È possibile utilizzare questa chiave in una policy per consentire ai responsabili che hanno impostato un'identità di origine quando assumono un ruolo. AWS L'attività per l'identità di origine specificata del ruolo viene visualizzata in AWS CloudTrail. In questo modo è più facile per gli amministratori determinare chi o cosa ha eseguito le azioni con un ruolo in. AWS

A differenza di `sts:RoleSessionName`, dopo aver impostato l'identità di origine, il valore non può essere modificato. È presente nel contesto della richiesta di tutte le operazioni intraprese dal ruolo. Il valore persiste nelle sessioni di ruolo successive quando si utilizzano le credenziali di sessione per assumere un altro ruolo. L'assunzione di un ruolo partendo da un altro si chiama concatenamento del ruolo.

La `sts:SourceIdentity` chiave è presente nella richiesta quando il principale imposta inizialmente un'identità di origine mentre assume un ruolo utilizzando qualsiasi comando CLI AWS STS assume-role o operazione API. AWS STS `AssumeRole` La chiave `aws:SourceIdentity` è presente nella richiesta per tutte le operazioni eseguite con una sessione di ruolo con un set di identità di origine.

La policy di attendibilità del ruolo riportata di seguito per `CriticalRole` nell'account `111122223333` contiene una condizione per `aws:SourceIdentity`che impedisce a un principale senza un'identità di origine impostata su Saanvi o Diego di assumere il ruolo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AssumeRoleIfSourceIdentity",
```

```
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::123456789012:role/CriticalRole"},
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringLike": {
                    "aws:SourceIdentity": ["Saanvi","Diego"]
                }
            }
        }
    ]
}
```

Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

ec2: RoleDelivery

Utilizzare questa chiave per confrontare la versione del servizio di metadati dell'istanza nella richiesta firmata con le credenziali del ruolo IAM per Amazon EC2. Il servizio di metadati dell'istanza esegue la distinzione tra richieste IMDSv1 e IMDSv2 a seconda che, per una determinata richiesta, le intestazioni `PUT` o `GET`, che sono univoche per IMDSv2, siano presenti in tale richiesta.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che la sessione del ruolo viene creata da un'istanza Amazon EC2.
- Tipo di dati: numerico
- Tipo di valore — Single-valued
- Valori di esempio: 1.0, 2.0

Puoi configurare il servizio di metadati dell'istanza (IMDS) su ogni istanza in modo che il codice locale o gli utenti utilizzino IMDSv2. Quando specifichi l'utilizzo di IMDSv2, IMDSv1 non funziona più.

- Instance Metadata Service versione 1 (IMDSv1): un metodo request/response
- Servizio di metadati dell'istanza Versione 2 (IMDSv2): un metodo orientato alla sessione

Per informazioni su come configurare l'istanza per l'utilizzo di IMDSv2, consulta Configurare le opzioni dei metadati dell'istanza.

Nell'esempio seguente, l'accesso viene negato se il RoleDelivery valore ec2: nel contesto della richiesta è 1.0 (IMDSv1). Questa istruzione di policy può essere applicata a livello generale perché, se la richiesta non è firmata dalle credenziali del ruolo Amazon EC2, non ha alcun effetto.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
                {
            "Sid": "RequireAllEc2RolesToUseV2",
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "NumericLessThan": {
                    "ec2:RoleDelivery": "2.0"
                }
            }
        }
    ]
}
```

Per ulteriori informazioni, consulta Esempi di policy per l'utilizzo dei metadati delle istanze.

ec2: SourceInstanceArn

Utilizzare questa chiave per confrontare l'ARN dell'istanza da cui è stata generata la sessione del ruolo.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che la sessione del ruolo viene creata da un'istanza Amazon EC2.
- Tipo di dati: ARN
- Tipo di valore — Single-valued
- Valore di esempio: arn:aws:ec2:us-west- 2:1111: -id instance/instance

Per le policy di esempio, consulta Consentire a un'istanza specifica di visualizzare le risorse in altri servizi AWS.

colla: RoleAssumedBy

Il AWS Glue servizio imposta questa chiave di condizione per ogni richiesta AWS API laddove AWS Glue effettua una richiesta utilizzando un ruolo di servizio per conto del cliente (non da un endpoint di lavoro o sviluppatore, ma direttamente dal AWS Glue servizio). Usa questa chiave per verificare se una chiamata a una AWS risorsa proviene dal AWS Glue servizio.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando si AWS Glue effettua una richiesta utilizzando un ruolo di servizio per conto del cliente.
- Tipo di dati: stringa
- Tipo di valore: Single-valued
- Valore di esempio: questa chiave è sempre impostata su `glue.amazonaws.com`.

L'esempio seguente aggiunge una condizione per consentire al AWS Glue servizio di ottenere un oggetto da un bucket Amazon S3.

```
{
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
        "StringEquals": {
            "glue:RoleAssumedBy": "glue.amazonaws.com"
        }
    }
}
```

colla: CredentialIssuingService

Il AWS Glue servizio imposta questa chiave per ogni richiesta AWS API utilizzando un ruolo di servizio proveniente da un endpoint di lavoro o sviluppatore. Usa questa chiave per verificare se una chiamata a una AWS risorsa proviene da un endpoint di AWS Glue lavoro o di sviluppatore.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando si AWS Glue effettua una richiesta proveniente da un endpoint di lavoro o di sviluppatore.
- Tipo di dati: stringa

- Tipo di valore: Single-valued
- Valore di esempio: questa chiave è sempre impostata su `glue.amazonaws.com`.

L'esempio seguente aggiunge una condizione associata a un ruolo IAM utilizzato da un AWS Glue job. Ciò garantisce che determinate azioni si allowed/denied basino sul fatto che la sessione di ruolo venga utilizzata per un ambiente di runtime del AWS Glue lavoro.

```
{
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
        "StringEquals": {
            "glue:CredentialIssuingService": "glue.amazonaws.com"
        }
    }
}
```

codebuild: BuildArn

Questa chiave identifica l'ARN di CodeBuild build a cui sono state fornite le credenziali del ruolo IAM. Usa questa chiave per verificare se una chiamata a una AWS risorsa proviene da una build specifica. CodeBuild

> **Note**
>
> Il valore completo di non `codebuild:BuildArn` è noto in anticipo perché contiene l'ID di build generato dinamicamente.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta ogni volta che una richiesta viene effettuata da un ruolo assunto da CodeBuild.
- Tipo di dati: ARN
- Tipo di valore: Single-valued
- Valore di esempio: arn:aws:codebuild:us-east-1:123456789012 ::12345678-1234-1234-1234-1234-123456789012 build/MyBuildProject

L'esempio seguente consente a una build specifica di accedere al bucket CodeBuild specificato. `s3:GetObject`

```
{
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
        "ArnLike": {
            "codebuild:BuildArn": "arn:aws:codebuild:us-east-1:123456789012:build/
MyBuildProject:*"
        }
    }
}
```

codebuild: ProjectArn

Questa chiave identifica l'ARN del CodeBuild progetto in base al quale le credenziali del ruolo IAM sono state fornite a una build. CodeBuild Usa questa chiave per verificare se una chiamata a una AWS risorsa proviene da un progetto specifico. CodeBuild

- Disponibilità: questa chiave è inclusa nel contesto della richiesta ogni volta che una richiesta viene effettuata da un ruolo assunto da CodeBuild.
- Tipo di dati: ARN
- Tipo di valore: Single-valued
- Valore di esempio: arn:aws:codebuild:us-east- 1:123456789012: project/MyBuildProject

L'esempio seguente consente a qualsiasi build di un progetto specifico di avere accesso al bucket specificato. CodeBuild `s3:GetObject`

```
{
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
        "ArnEquals": {
            "codebuild:ProjectArn": "arn:aws:codebuild:us-east-1:123456789012:project/
MyBuildProject"
        }
```

```
    }
}
```

lambda: SourceFunctionArn

Utilizza questa chiave per identificare l'ARN della funzione Lambda a cui sono state consegnate le credenziali del ruolo IAM. Il servizio Lambda imposta questa chiave per ogni richiesta AWS API proveniente dall'ambiente di esecuzione della funzione. Usa questa chiave per verificare se una chiamata a una AWS risorsa proviene dal codice di una specifica funzione Lambda. Lambda imposta questa chiave anche per alcune richieste che provengono dall'esterno dell'ambiente di esecuzione, come la scrittura di log CloudWatch e l'invio di tracce a. X-Ray

- Disponibilità: questa chiave è inclusa nel contesto della richiesta ogni volta che viene richiamato il codice della funzione Lambda.
- Tipo di dati: ARN
- Tipo di valore: Single-valued
- Valore di esempio: arn:aws:lambda:us-east- 1:123456789012:function: TestFunction

L'esempio seguente consente a una funzione Lambda specifica definire l'accesso di `s3:PutObject` al bucket specificato.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ExampleSourceFunctionArn",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "ArnEquals": {
                    "lambda:SourceFunctionArn": "arn:aws:lambda:us-
east-1:123456789012:function:source_lambda"
                }
            }
        }
    ]
```

```
}
```

Per ulteriori informazioni, consulta Lavorare con le credenziali dell'ambiente di esecuzione Lambda nella Guida per gli sviluppatori di AWS Lambda .

sms: SourceInstanceArn

Usa questa chiave per identificare l'ARN dell'istanza AWS Systems Manager gestita a cui sono state fornite le credenziali del ruolo IAM. Questa chiave di condizione non è presente quando la richiesta proviene da un'istanza gestita con un ruolo IAM associato al profilo dell'istanza Amazon EC2.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta ogni volta che le credenziali del ruolo vengono consegnate a un'istanza gestita da AWS Systems Manager .
- Tipo di dati: ARN
- Tipo di valore: Single-valued
- Valore di esempio: arn:aws:ec2:us-west- 2:1111: -id instance/instance

archivio di identità: UserId

Utilizzare questa chiave per confrontare l'identità della forza lavoro del Centro identità IAM nella richiesta firmata con l'identità specificata nella policy.

- Disponibilità: questa chiave viene inclusa quando il chiamante della richiesta è un utente del Centro identità IAM.
- Tipo di dati: stringa
- Tipo di valore — Single-valued
- Valore di esempio: 94482488-3041-7026-18f3-be45837cd0e4

Puoi trovare il nome UserId di un utente in IAM Identity Center effettuando una richiesta all'GetUserIdAPI utilizzando AWS CLI, AWS API o AWS SDK.

Proprietà della rete

Utilizza le chiavi di condizione seguenti per confrontare i dettagli della rete da cui è stata originata la richiesta o inviati con le proprietà della rete specificate nella policy.

era: SourceIp

Utilizzare questa chiave per confrontare l'indirizzo IP del richiedente con l'indirizzo IP specificato nella policy. La chiave di condizione `aws:SourceIp` può essere utilizzata solo per intervalli di indirizzi IP pubblici.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta, tranne quando il richiedente utilizza un endpoint VPC per effettuare la richiesta.
- Tipo di dati: indirizzo IP
- Tipo di valore — Single-valued

La chiave di condizione `aws:SourceIp` può essere utilizzata in una policy per consentire ai principali di effettuare richieste solo all'interno di un intervallo IP specificato.

> **Note**
>
> `aws:SourceIp` supporta indirizzi sia IPv4 sia IPv6 e intervalli di indirizzi IP. Per un elenco di quelli Servizi AWS che supportano IPv6, consulta Servizi AWS il supporto IPv6 nella Amazon VPC User Guide.

Ad esempio, puoi collegare la seguente policy basata sull'identità a un ruolo IAM. Questa policy consente all'utente di collocare oggetti nel bucket di Amazon S3 `amzn-s3-demo-bucket3` se effettua la chiamata dall'intervallo di indirizzi IPv4 specificato. Questa politica consente anche a un AWS servizio che esegue questa operazione Inoltro delle sessioni di accesso per tuo conto.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PrincipalPutObjectIfIpAddress",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket3/*",
            "Condition": {
                "IpAddress": {
```

```
                    "aws:SourceIp": "203.0.113.0/24"
                }
            }
        }
    ]
}
```

Se devi limitare l'accesso a reti che supportano l'indirizzamento sia IPv4 che IPv6, puoi includere l'indirizzo o gli intervalli di indirizzi IP IPv4 e IPv6 nella condizione della policy IAM. La seguente policy basata sull'identità consente all'utente di collocare oggetti nel bucket di Amazon S3 `amzn-s3-demo-bucket3` se l'utente effettua la chiamata dall'intervallo di indirizzi IPv4 o IPv6 specificato. Prima di includere gli intervalli di indirizzi IPv6 nella tua policy IAM, verifica che quella con Servizio AWS cui stai lavorando supporti IPv6. Per un elenco di quelli Servizi AWS che supportano l'IPv6, consulta Servizi AWS il supporto IPv6 nella Amazon VPC User Guide.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PrincipalPutObjectIfIpAddress",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket3/*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "203.0.113.0/24",
                        "2001:DB8:1234:5678::/64"
                    ]
                }
            }
        }
    ]
}
```

Se la richiesta proviene da un host che utilizza un endpoint Amazon VPC, la chiave `aws:SourceIp` non è disponibile. Dovresti invece usare una VPC-specific chiave come aws:. VpcSourceIp Per

ulteriori informazioni sull'utilizzo degli endpoint VPC, consulta la sezione Gestione delle identità e degli accessi per endpoint VPC e servizi endpoint VPC nella Guida di AWS PrivateLink .

> **Note**
>
> Quando Servizi AWS effettui chiamate ad altri per tuo Servizi AWS conto (chiamate da servizio a servizio), alcuni contesti di autorizzazione specifici della rete vengono oscurati. Se la tua policy utilizza questa chiave di condizione con le `Deny` dichiarazioni, i principali potrebbero essere bloccati involontariamente. Servizio AWS Per consentire ai Servizi AWS di funzionare correttamente mantenendo i requisiti di sicurezza, escludi i principali del servizio dalle istruzioni `Deny` aggiungendo la chiave di condizione `aws:PrincipalIsAWSService` con il valore `false`.

era: SourceVpc

Utilizzare questa chiave per verificare se la richiesta viaggia attraverso il VPC al quale è collegato l'endpoint VPC. In una policy è possibile utilizzare questa chiave per consentire l'accesso solo a un VPC specifico. Per ulteriori informazioni, consulta Limitazione dell'accesso a un VPC specifico nella Guida per l'utente di Amazon Simple Storage Service.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo se il richiedente utilizza un endpoint VPC per effettuare la richiesta.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

In una policy è possibile utilizzare questa chiave per consentire o limitare l'accesso a un VPC specifico.

Ad esempio, puoi allegare la seguente policy basata sull'identità a un ruolo IAM da negare al bucket `amzn-s3-demo-bucket3` Amazon S3, `PutObject` a meno che la richiesta non venga effettuata dall'ID VPC specificato o non utilizzi sessioni di accesso diretto (FAS) per effettuare richieste per conto del ruolo. Servizi AWS A differenza di era: SourceIp, è necessario utilizzare aws:ViaAWSService o ha detto: CalledVia per consentire le richieste FAS, poiché il VPC di origine della richiesta iniziale non viene conservato.

> **Note**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "PutObjectIfNotVPCID",
      "Effect": "Deny",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket3/*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceVpc": "vpc-1234567890abcdef0"
        },
        "Bool": {
          "aws:ViaAWSService": "false"
        }
      }
    }
  ]
}
```

> **Note**
>
> AWS consiglia di utilizzare al `aws:SourceVpcArn` posto di `aws:SourceVpc` if `aws:SourceVpcArn` è supportato dal servizio a cui ti rivolgi. Fai riferimento a aws: SourceVpcArn per l'elenco dei servizi supportati.

> **Note**
>
> Quando Servizi AWS effettui chiamate ad altri per tuo Servizi AWS conto (chiamate da servizio a servizio), alcuni contesti di autorizzazione specifici della rete vengono oscurati.

Se la tua policy utilizza questa chiave di condizione con le `Deny` dichiarazioni, i principali potrebbero essere bloccati involontariamente. Servizio AWS Per consentire ai Servizi AWS di funzionare correttamente mantenendo i requisiti di sicurezza, escludi i principali del servizio dalle istruzioni `Deny` aggiungendo la chiave di condizione `aws:PrincipalIsAWSService` con il valore `false`.

era: SourceVpcArn

Usa questa chiave per verificare l'ARN del VPC attraversato da una richiesta utilizzando un endpoint VPC. Questa chiave restituisce l'ARN del VPC a cui è collegato l'endpoint VPC.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta per i servizi supportati quando una richiesta viene effettuata tramite un endpoint VPC. La chiave non è inclusa per le richieste effettuate tramite endpoint di servizio pubblico. I seguenti servizi supportano questa chiave:
  - AWS App Runner (prefisso:) `apprunner`
  - AWS Application Discovery Service (prefisso:) `discovery`
  - Amazon Athena (prefisso: `athena`)
  - AWS Cloud Map (prefisso:) `servicediscovery`
  - Amazon CloudWatch Application Insights (prefisso:) `applicationinsights`
  - AWS CloudFormation (prefisso:) `cloudformation`
  - Amazon Comprehend Medical (prefisso: `comprehendmedical`)
  - AWS Compute Optimizer (prefisso:) `compute-optimizer`
  - Amazon; Elastic Container Registry (prefisso:) `ecr`
  - Amazon Elastic Container Service (prefisso:) `ecs`
  - Amazon Kinesis Analytics (prefisso:) `kinesisanalytics`
  - Amazon Route 53 (prefisso:) `route53`
  - AWS DataSync (prefisso:) `datasync`
  - Amazon Elastic Block Store (prefisso: `ebs`)
  - Amazon EventBridge Scheduler (prefisso:) `scheduler`
  - Amazon Data Firehose (prefisso: `firehose`)
  - AWS HealthImaging (prefisso:) `medical-imaging`
  - AWS HealthLake (prefisso:) `healthlake`

- AWS HealthOmics (prefisso:) `omics`
- AWS Identity and Access Management (ad eccezione dell'`iam:PassRole`azione) (prefisso:) `iam`
- AWS IoT FleetWise (prefisso:) `iotfleetwise`
- Wireless AWS IoT (prefisso:) `iotwireless`
- AWS Key Management Service (prefisso:) `kms`
- AWS Lambda (prefisso:) `lambda`
- AWS Payment Cryptography (prefisso:) `payment-cryptography`
- Amazon Polly (prefisso: `polly`)
- AWS Autorità di certificazione privata (prefisso:) `acm-pca`
- AWS Cestino (prefisso:) `rbin`
- Amazon Rekognition (prefisso: `rekognition`)
- Service Quotas (prefisso: `servicequotas`)
- Amazon Simple Storage Service (prefisso: `s3`)
- Gateway di archiviazione AWS (prefisso:) `storagegateway`
- Strumento di gestione degli incidenti AWS Systems Manager Contatti (prefisso:) `ssm-contacts`
- Amazon Textract (prefisso: `textract`)
- Amazon Transcribe (prefisso: `transcribe`)
- AWS Transfer Family (prefisso:) `transfer`

- Tipo di dati: ARN

  AWS consiglia di utilizzare gli operatori ARN anziché gli operatori di stringa quando si confrontano gli ARN.
- Tipo di valore: Single-valued
- Valore di esempio: `arn:aws:ec2:us-east-1:123456789012:vpc/vpc-0e9801d129EXAMPLE`

Di seguito è riportato un esempio di policy bucket che nega l'accesso `amzn-s3-demo-bucket` e ai relativi oggetti da qualsiasi luogo esterno a VPC. `vpc-1a2b3c4d`

```
{
   "Version":"2012-10-17",
   "Statement": [
```

```
    {
      "Sid": "Access-to-specific-VPC-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket",
                   "arn:aws:s3:::amzn-s3-demo-bucket/*"],
      "Condition": {
        "ArnNotEquals": {
          "aws:SourceVpcArn": "arn:aws:ec2:us-east-1:*:vpc/vpc-1a2b3c4d"
        }
      }
    }
  ]
}
```

> **Note**
>
> Quando si Servizi AWS effettuano chiamate ad altri Servizi AWS per conto dell'utente (chiamate da servizio a servizio), alcuni contesti di autorizzazione specifici della rete vengono oscurati. Se la tua policy utilizza questa chiave di condizione con le `Deny` dichiarazioni, i principali potrebbero essere bloccati involontariamente. Servizio AWS Per consentire ai Servizi AWS di funzionare correttamente mantenendo i requisiti di sicurezza, escludi i principali del servizio dalle istruzioni `Deny` aggiungendo la chiave di condizione `aws:PrincipalIsAWSService` con il valore `false`.

era: SourceVpce

Utilizzare questa chiave per confrontare l'identificatore dell'endpoint VPC della richiesta con l'ID endpoint specificato nella policy.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo se il richiedente utilizza un endpoint VPC per effettuare la richiesta.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

In una policy è possibile utilizzare questa chiave per limitare l'accesso a un endpoint VPC specifico. Per ulteriori informazioni, consulta Limitazione dell'accesso a un VPC specifico nella Guida per

l'utente di Amazon Simple Storage Service. Analogamente all'utilizzoera: SourceVpc, è necessario ha detto: CalledVia utilizzare aws:ViaAWSService o consentire le richieste effettuate Servizi AWS utilizzando le sessioni di accesso diretto (FAS). Questo perché l'endpoint VPC di origine della richiesta iniziale non viene conservato.

> **Note**
>
> Quando si Servizi AWS effettuano chiamate ad altri Servizi AWS per conto dell'utente (chiamate da servizio a servizio), alcuni contesti di autorizzazione specifici della rete vengono oscurati. Se la tua policy utilizza questa chiave di condizione con le `Deny` dichiarazioni, i principali potrebbero essere bloccati involontariamente. Servizio AWS Per consentire ai Servizi AWS di funzionare correttamente mantenendo i requisiti di sicurezza, escludi i principali del servizio dalle istruzioni `Deny` aggiungendo la chiave di condizione `aws:PrincipalIsAWSService` con il valore `false`.

era: VpceAccount

Usa questa chiave per confrontare l'ID dell' AWS account proprietario dell'endpoint VPC tramite il quale è stata effettuata la richiesta con l'ID dell'account specificato nella policy. Questa chiave di condizione consente di stabilire controlli del perimetro di rete assicurando che le richieste arrivino tramite endpoint VPC di proprietà di account specifici.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando una richiesta viene effettuata tramite un endpoint VPC. La chiave non è inclusa per le richieste effettuate tramite endpoint di servizio pubblico.

  I seguenti servizi supportano questa chiave:

  - AWS App Runner (prefisso:) `apprunner`
  - AWS Application Discovery Service (prefisso:) `discovery`
  - Amazon Athena (prefisso: `athena`)
  - AWS Cloud Map (prefisso:) `servicediscovery`
  - Amazon CloudWatch Application Insights (prefisso:) `applicationinsights`
  - AWS CloudFormation (prefisso:) `cloudformation`
  - Amazon Comprehend Medical (prefisso: `comprehendmedical`)
  - AWS Compute Optimizer (prefisso:) `compute-optimizer`

- Amazon; Elastic Container Registry (prefisso:) `ecr`
- Amazon Elastic Container Service (prefisso:) `ecs`
- Amazon Kinesis Analytics (prefisso:) `kinesisanalytics`
- Amazon Route 53 (prefisso:) `route53`
- AWS DataSync (prefisso:) `datasync`
- Amazon DynamoDB (prefisso:) `dynamodb`
- Amazon Elastic Block Store (prefisso: `ebs`)
- Amazon EventBridge Scheduler (prefisso:) `scheduler`
- Amazon Data Firehose (prefisso: `firehose`)
- Amazon GameLift (prefisso:) `gamelift`
- AWS HealthImaging (prefisso:) `medical-imaging`
- AWS HealthLake (prefisso:) `healthlake`
- AWS HealthOmics (prefisso:) `omics`
- AWS Identity and Access Management (ad eccezione dell'`iam:PassRole`azione) (prefisso:) `iam`
- AWS IoT FleetWise (prefisso:) `iotfleetwise`
- Wireless AWS IoT (prefisso:) `iotwireless`
- AWS Key Management Service (prefisso:) `kms`
- AWS Lambda (prefisso:) `lambda`
- AWS Payment Cryptography (prefisso:) `payment-cryptography`
- Amazon Polly (prefisso: `polly`)
- AWS Autorità di certificazione privata (prefisso:) `acm-pca`
- AWS Cestino (prefisso:) `rbin`
- Amazon Rekognition (prefisso: `rekognition`)
- Service Quotas (prefisso: `servicequotas`)
- Amazon Simple Storage Service (prefisso: `s3`)
- AWS Shield (prefisso:) `shield`
- Gateway di archiviazione AWS (prefisso:) `storagegateway`
- Strumento di gestione degli incidenti AWS Systems Manager Contatti (prefisso:) `ssm-contacts`
- Amazon Textract (prefisso: `textract`)

  - Amazon Transcribe (prefisso: `transcribe`)
  - AWS Transfer Family (prefisso:) `transfer`
- Tipo di dati: stringa
- Tipo di valore: Single-valued
- Valore di esempio: `123456789012`

Puoi utilizzare questa chiave di condizione per limitare l'accesso alle risorse in modo che le richieste arrivino tramite endpoint VPC di proprietà del tuo account. Il seguente esempio di policy di bucket di Amazon S3 consente l'accesso quando la richiesta arriva tramite un endpoint VPC di proprietà dell'account specificato:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AccessToSpecificVpceAccountOnly",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/RoleName"
            },
            "Action": "s3:GetObject",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "StringEquals": {
                    "aws:VpceAccount": "111122223333"
                }
            }
        }
    ]
}
```

> **Note**
>
> Questa chiave condizionale è attualmente supportata per un set selezionato di AWS servizi. L'utilizzo di questa chiave con servizi non supportati può portare a esiti di autorizzazione non intenzionali. Definisci sempre la chiave di condizione per i servizi supportati nelle tue policy.

Alcuni AWS servizi accedono alle tue risorse dalle loro reti quando agiscono per tuo conto. Se utilizzi tali servizi, dovrai modificare l'esempio di politica sopra riportato per consentire ai AWS servizi di accedere alle tue risorse dall'esterno della tua rete. Per ulteriori informazioni sui modelli di accesso da tenere in considerazione quando si applicano i controlli di accesso in base all'origine della richiesta, consulta Establish permissions guardrails using data perimeters.

aws: VpceOrg ID

Usa questa chiave per confrontare l'identificatore dell'organizzazione proprietaria dell'endpoint VPC da cui è stata effettuata la richiesta con l'identificatore specificato nella policy. AWS Organizations Questa chiave di condizione offre l'approccio più scalabile ai controlli del perimetro di rete, includendo automaticamente tutti gli endpoint VPC di proprietà degli account all'interno dell'organizzazione.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando una richiesta viene effettuata tramite un endpoint VPC e l'account del proprietario dell'endpoint VPC è membro di un'organizzazione. AWS La chiave non è inclusa per le richieste effettuate tramite altri percorsi di rete o quando l'account del proprietario dell'endpoint VPC non fa parte di un'organizzazione.

  I seguenti servizi supportano questa chiave:

  - AWS App Runner (prefisso:) `apprunner`
  - AWS Application Discovery Service (prefisso:) `discovery`
  - Amazon Athena (prefisso: `athena`)
  - AWS B2B Data Interchange (prefisso:) `b2bi`
  - AWS Cloud Map (prefisso:) `servicediscovery`
  - AWS CloudHSM (prefisso:) `cloudhsmv2`
  - Amazon CloudWatch Application Insights (prefisso:) `applicationinsights`
  - AWS CloudFormation (prefisso:) `cloudformation`
  - Amazon Cognito (prefisso:) `cognito-identity`
  - Amazon Comprehend Medical (prefisso: `comprehendmedical`)

- AWS Compute Optimizer (prefisso:) `compute-optimizer`
- AWS Database Migration Service (prefisso:) `dms`
- AWS Dati del servizio di elenco (prefisso:) `ds-data`
- Amazon DynamoDB (prefisso:) `dynamodb`
- Amazon; Elastic Container Registry (prefisso:) `ecr`
- Amazon Elastic Container Service (prefisso:) `ecs`
- Amazon Kinesis Data Streams (prefisso:) `kinesis`
- Amazon Kinesis Analytics (prefisso:) `kinesisanalytics`
- Amazon Q Business (prefisso:) `qbusiness`
- Amazon Route 53 (prefisso:) `route53`
- AWS DataSync (prefisso:) `datasync`
- Amazon Elastic Block Store (prefisso: `ebs`)
- Amazon EventBridge Scheduler (prefisso:) `scheduler`
- Amazon Data Firehose (prefisso: `firehose`)
- Amazon GameLift (prefisso:) `gamelift`
- AWS HealthImaging (prefisso:) `medical-imaging`
- AWS HealthLake (prefisso:) `healthlake`
- AWS HealthOmics (prefisso:) `omics`
- AWS Identity and Access Management (ad eccezione dell'`iam:PassRole`azione) (prefisso:) `iam`
- AWS Identity Store (prefisso:) `identitystore`
- AWS IoT FleetWise (prefisso:) `iotfleetwise`
- AWS IoT TwinMaker (prefisso:) `iottwinmaker`
- Wireless AWS IoT (prefisso:) `iotwireless`
- Amazon Keyspaces (per Apache Cassandra) (prefisso:) `cassandra`
- AWS Key Management Service (prefisso:) `kms`
- AWS Lambda (prefisso:) `lambda`
- AWS Network Firewall (prefisso:) `network-firewall`
- AWS Payment Cryptography (prefisso:) `payment-cryptography`
- Servizio SMS e voce Amazon Pinpoint (prefisso:) `sms-voice`

  - Amazon Polly (prefisso: `polly`)
  - Listino prezzi AWS (prefisso:) `pricing`
  - AWS Autorità di certificazione privata (prefisso: `acm-pca`)
  - AWS Cestino (prefisso:) `rbin`
  - Amazon Rekognition (prefisso: `rekognition`)
  - Service Quotas (prefisso: `servicequotas`)
  - Amazon Simple Email Service (prefisso:) `ses`
  - AWS Shield (prefisso:) `shield`
  - Amazon Simple Storage Service (prefisso: `s3`)
  - Amazon Simple Queue Service (prefisso:) `sqs`
  - Gateway di archiviazione AWS (prefisso:) `storagegateway`
  - Strumento di gestione degli incidenti AWS Systems Manager Contatti (prefisso:) `ssm-contacts`
  - Amazon Textract (prefisso: `textract`)
  - Amazon Transcribe (prefisso: `transcribe`)
  - AWS Transfer Family (prefisso:) `transfer`
  - Amazon WorkMail (prefisso:) `workmail`
- Tipo di dati: stringa
- Tipo di valore: Single-valued
- Valori di esempio: `o-a1b2c3d4e5`

Il seguente esempio di politica di controllo delle risorse nega l'accesso ad Amazon S3 e alle AWS Key Management Service risorse a meno che la richiesta non arrivi tramite endpoint VPC di proprietà dell'organizzazione specificata o da reti di AWS servizi che agiscono per tuo conto. Alcune organizzazioni potrebbero dover modificare ulteriormente questa politica per soddisfare le esigenze della propria organizzazione, ad esempio, consentire l'accesso a partner di terze parti. Per ulteriori informazioni sui modelli di accesso da tenere in considerazione quando si applicano i controlli di accesso in base all'origine della richiesta, consulta Establish permissions guardrails using data perimeters.

JSON

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceNetworkPerimeterVpceOrgID",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:*",
        "kms:*"
      ],
      "Resource": "*",
      "Condition": {
        "BoolIfExists": {
          "aws:PrincipalIsAWSService": "false",
          "aws:ViaAWSService": "false"
        },
        "StringNotEqualsIfExists": {
            "aws:VpceOrgID": "o-abcdef0123",
            "aws:PrincipalTag/network-perimeter-exception": "true"
        }
      }
    }
  ]
}
```

> **Note**
>
> Questa chiave condizionale è attualmente supportata per un set selezionato di AWS servizi. L'utilizzo di questa chiave con servizi non supportati può portare a esiti di autorizzazione non intenzionali. Definisci sempre la chiave di condizione per i servizi supportati nelle tue policy.

> **Note**
>
> Quando si Servizi AWS effettuano chiamate ad altri Servizi AWS per conto dell'utente (chiamate da servizio a servizio), alcuni contesti di autorizzazione specifici della rete vengono oscurati. Se la tua policy utilizza questa chiave di condizione con le `Deny` dichiarazioni, i principali potrebbero essere bloccati involontariamente. Servizio AWS Per consentire ai Servizi AWS di funzionare correttamente mantenendo i requisiti di sicurezza,

escludi i principali del servizio dalle istruzioni `Deny` aggiungendo la chiave di condizione `aws:PrincipalIsAWSService` con il valore `false`.

era: VpceOrgPaths

Usa questa chiave per confrontare il AWS Organizations percorso dell'endpoint VPC da cui è stata effettuata la richiesta con il percorso specificato nella policy. Questa chiave di condizione consente di implementare i controlli del perimetro di rete a livello di unità organizzativa (UO), scalando automaticamente in base all'utilizzo degli endpoint VPC man mano che si aggiungono nuovi endpoint all'interno delle unità organizzative specificate.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta quando viene effettuata una richiesta tramite un endpoint VPC e l'account del proprietario dell'endpoint VPC è membro di un'organizzazione. La chiave non è inclusa per le richieste effettuate tramite altri percorsi di rete o quando l'account del proprietario dell'endpoint VPC non fa parte di un'organizzazione.

  I seguenti servizi supportano questa chiave:

  - AWS App Runner (prefisso:) `apprunner`
  - AWS Application Discovery Service (prefisso:) `discovery`
  - Amazon Athena (prefisso: `athena`)
  - AWS Cloud Map (prefisso:) `servicediscovery`
  - Amazon CloudWatch Application Insights (prefisso:) `applicationinsights`
  - AWS CloudFormation (prefisso:) `cloudformation`
  - Amazon Comprehend Medical (prefisso: `comprehendmedical`)
  - AWS Compute Optimizer (prefisso:) `compute-optimizer`
  - Amazon; Elastic Container Registry (prefisso:) `ecr`
  - Amazon Elastic Container Service (prefisso:) `ecs`
  - Amazon Kinesis Analytics (prefisso:) `kinesisanalytics`
  - Amazon Route 53 (prefisso:) `route53`
  - AWS DataSync (prefisso:) `datasync`
  - Amazon DynamoDB (prefisso:) `dynamodb`
  - Amazon Elastic Block Store (prefisso: `ebs`)
  - Amazon EventBridge Scheduler (prefisso:) `scheduler`

- Amazon Data Firehose (prefisso: `firehose`)
- Amazon GameLift (prefisso:) `gamelift`
- AWS HealthImaging (prefisso:) `medical-imaging`
- AWS HealthLake (prefisso:) `healthlake`
- AWS HealthOmics (prefisso:) `omics`
- AWS Identity and Access Management (ad eccezione dell'`iam:PassRole`azione) (prefisso:) `iam`
- AWS IoT FleetWise (prefisso:) `iotfleetwise`
- Wireless AWS IoT (prefisso:) `iotwireless`
- AWS Key Management Service (prefisso:) `kms`
- AWS Lambda (prefisso:) `lambda`
- AWS Payment Cryptography (prefisso:) `payment-cryptography`
- Amazon Polly (prefisso: `polly`)
- AWS Autorità di certificazione privata (prefisso:) `acm-pca`
- AWS Cestino (prefisso:) `rbin`
- Amazon Rekognition (prefisso: `rekognition`)
- Service Quotas (prefisso: `servicequotas`)
- AWS Secrets Manager (prefisso:) `secretsmanager`
- Servizio di token di sicurezza AWS (prefisso:) `sts`
- Amazon Simple Storage Service (prefisso: `s3`)
- AWS Scudo (prefisso:) `shield`
- Gateway di archiviazione AWS (prefisso:) `storagegateway`
- Strumento di gestione degli incidenti AWS Systems Manager Contatti (prefisso:) `ssm-contacts`
- Amazon Textract (prefisso: `textract`)
- Amazon Transcribe (prefisso: `transcribe`)
- AWS Transfer Family (prefisso:) `transfer`

- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore
- Valori di esempio: `o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/`

Poiché `aws:VpceOrgPaths` è una chiave di condizione multivalore, è necessario utilizzare gli operatori dell'insieme `ForAnyValue` o `ForAllValues` con gli operatori di condizione stringa per questa chiave. Il seguente esempio di policy di bucket di Amazon S3 consente l'accesso solo quando le richieste arrivano tramite endpoint VPC di proprietà di account in unità organizzative specifiche:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessFromSpecificOrgPaths",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/RoleName"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "aws:VpceOrgPaths": [
            "o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/*",
            "o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-33333333/*"
          ]
        }
      }
    }
  ]
}
```

> **Note**
>
> Questa chiave di condizione è attualmente supportata per un set selezionato di AWS servizi. L'utilizzo di questa chiave con servizi non supportati può portare a esiti di autorizzazione non intenzionali. Definisci sempre la chiave di condizione per i servizi supportati nelle tue policy.

Alcuni AWS servizi accedono alle tue risorse dalle loro reti quando agiscono per tuo conto. Se utilizzi tali servizi, dovrai modificare l'esempio di politica sopra riportato per consentire ai AWS servizi di

accedere alle tue risorse dall'esterno della tua rete. Per ulteriori informazioni sui modelli di accesso da tenere in considerazione quando si applicano i controlli di accesso in base all'origine della richiesta, consulta Establish permissions guardrails using data perimeters.

leggi: VpcSourceIp

Utilizzare questa chiave per confrontare l'indirizzo IP da cui è stata effettuata una richiesta con l'indirizzo IP specificato nella policy. In una policy, la chiave corrisponde solo se la richiesta proviene dall'indirizzo IP specificato e passa attraverso un endpoint VPC.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo se la richiesta viene effettuata utilizzando un endpoint VPC.
- Tipo di dati: indirizzo IP
- Tipo di valore — Single-valued

Per ulteriori informazioni, consulta Controllo degli accessi agli endpoint VPC tramite le policy degli endpoint nella Guida per l'utente di Amazon VPC. Analogamente all'utilizzoera: SourceVpc, è necessario ha detto: CalledVia utilizzare aws:ViaAWSService o consentire le richieste effettuate Servizi AWS utilizzando le sessioni di accesso diretto (FAS). Questo perché l'IP di origine della richiesta iniziale effettuata tramite un endpoint VPC non viene conservato nelle richieste FAS.

> **Note**
>
> `aws:VpcSourceIp` supporta indirizzi sia IPv4 sia IPv6 e intervalli di indirizzi IP. Per un elenco di quelli Servizi AWS che supportano IPv6, consulta Servizi AWS il supporto IPv6 nella Amazon VPC User Guide.
> Usa sempre la chiave di `aws:VpcSourceIp` condizione con le chiavi di condizione`aws:SourceVpc`, o`aws:SourceVpce`. `aws:SourceVpcArn` In caso contrario, una policy potrebbe consentire chiamate API da un VPC inaspettato che utilizza lo stesso IP CIDR o si sovrappone. Ciò può accadere perché i CIDR IP di due VPC non correlati possono essere uguali o sovrapporsi.

> **Note**
>
> Quando si Servizi AWS effettuano chiamate ad altri Servizi AWS per conto dell'utente (chiamate da servizio a servizio), alcuni contesti di autorizzazione specifici della rete vengono oscurati. Se la tua policy utilizza questa chiave di condizione con le `Deny`

dichiarazioni, i principali potrebbero essere bloccati involontariamente. Servizio AWS Per consentire ai Servizi AWS di funzionare correttamente mantenendo i requisiti di sicurezza, escludi i principali del servizio dalle istruzioni `Deny` aggiungendo la chiave di condizione `aws:PrincipalIsAWSService` con il valore `false`.

Proprietà della risorsa

Utilizzare le chiavi di condizione seguenti per confrontare i dettagli della risorsa che è la destinazione della richiesta con le proprietà della risorsa specificate nella policy.

era: ResourceAccount

Utilizza questa chiave per confrontare l'ID Account AWS del proprietario della risorsa richiesta con l'account della risorsa nella policy. Dopodiché, puoi consentire o negare l'accesso a tale risorsa in base all'account proprietario della stessa.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta per la maggior parte delle operazioni dei servizi. Le seguenti operazioni non supportano questa chiave:
  - AWS Audit Manager
    - `auditmanager:UpdateAssessmentFrameworkShare`
  - Amazon Detective
    - `detective:AcceptInvitation`
  - Servizio di directory AWS
    - `ds:AcceptSharedDirectory`
  - Amazon Elastic Block Store: tutte le operazioni
  - Amazon EC2
    - `ec2:AcceptTransitGatewayPeeringAttachment`
    - `ec2:AcceptVpcEndpointConnections`
    - `ec2:AcceptVpcPeeringConnection`
    - `ec2:CreateTransitGatewayPeeringAttachment`
    - `ec2:CreateVpcEndpoint`
    - `ec2:CreateVpcPeeringConnection`
  - Amazon EventBridge

    - `events:PutEvents`— EventBridge `PutEvents` chiamate su un event bus in un altro account, se tale event bus è stato configurato come EventBridge obiettivo per più account prima del 2 marzo 2023. Per ulteriori informazioni, consulta Concedere autorizzazioni per consentire eventi da altri AWS account nella Amazon EventBridge User Guide.
  - Amazon GuardDuty
    - `guardduty:AcceptAdministratorInvitation`
  - Amazon Macie
    - `macie2:AcceptInvitation`
  - OpenSearch Servizio Amazon
    - `es:AcceptInboundConnection`
  - Amazon Route 53
    - `route53:AssociateVpcWithHostedZone`
    - `route53:CreateVPCAssociationAuthorization`
  - AWS Security Hub CSPM
    - `securityhub:AcceptAdministratorInvitation`
- Tipo di dati: stringa
- Tipo di valore: Single-valued

> **Note**
>
> Per ulteriori considerazioni sulle operazioni non supportate di cui sopra, consulta il repository di esempi di policy del perimetro di dati.

Questa chiave è uguale all' Account AWS ID dell'account con le risorse valutate nella richiesta.

Per la maggior parte delle risorse presenti nel tuo account, l'ARN contiene l'ID account del proprietario della rispettiva risorsa. Per alcune risorse, come i bucket Amazon S3, l'ARN della risorsa non include l'ID account. I due esempi seguenti mostrano la differenza tra una risorsa il cui ARN contiene un ID account e un ARN Amazon S3 privo di un ID account:

- `arn:aws:iam::123456789012:role/AWSExampleRole`: ruolo IAM creato e di proprietà all'interno dell'account 123456789012.

- `arn:aws:s3:::amzn-s3-demo-bucket2`: bucket Amazon S3 creato e posseduto all'interno dell'account `111122223333`, non visualizzato nell'ARN.

Usa la AWS console, l'API o la CLI, per trovare tutte le tue risorse e gli ARN corrispondenti.

Scrivi una policy che nega le autorizzazioni alle risorse in base all'ID account del proprietario della risorsa. Ad esempio, la seguente policy basata sull'identità nega l'accesso alla risorsa specificata se la risorsa non appartiene all'account specificato.

Per utilizzare questa policy, sostituisci il testo segnaposto in corsivo con le tue informazioni.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza invece l'effetto `Deny`, che nega esplicitamente l'accesso a tutte le risorse elencate nell'istruzione che non appartengono all'account elencato. Utilizza questa policy in combinazione con altre policy che consentono l'accesso a risorse specifiche.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyInteractionWithResourcesNotInSpecificAccount",
      "Action": "service:*",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:service:us-east-1:111122223333:*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "111122223333"
          ]
        }
      }
    }
  ]
```

```
}
```

Questa policy nega l'accesso a tutte le risorse per un AWS servizio specifico, a meno che la risorsa non sia di Account AWS proprietà dell'utente specificato.

> **Note**
>
> Alcuni Servizi AWS richiedono l'accesso a risorse AWS di proprietà ospitate in un altro Account AWS. L'utilizzo di `aws:ResourceAccount` nelle tue policy basate sull'identità potrebbe influire sulla capacità della tua identità di accedere a queste risorse.

Alcuni AWS servizi, ad esempio AWS Data Exchange, si basano sull'accesso a risorse esterne all'utente Account AWS per le normali operazioni. Se usi l'elemento `aws:ResourceAccount` nelle tue policy, includi istruzioni aggiuntive per creare delle esenzioni per tali servizi. La policy AWS: nega l'accesso alle risorse Amazon S3 al di fuori del tuo account tranne AWS Data Exchange di esempio illustra come negare l'accesso in base all'account della risorsa definendo al contempo delle eccezioni per le risorse di proprietà del servizio.

Utilizza questo esempio di policy come modello per creare le tue policy personalizzate. Per ulteriori informazioni, consulta la documentazione del servizio.

era: ResourceOrgPaths

Usa questa chiave per confrontare il AWS Organizations percorso della risorsa a cui si accede con il percorso nella policy. In una policy, questa chiave della condizione garantisce che la risorsa appartenga a un membro dell'account all'interno del root dell'organizzazione o delle unità organizzative (UO) specificate in AWS Organizations. Un AWS Organizations percorso è una rappresentazione testuale della struttura di un'entità dell'organizzazione. Per ulteriori informazioni sull'utilizzo e la comprensione dei percorsi, consulta la sezione Comprendi il AWS Organizations percorso dell'entità.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta solo se l'account che possiede la risorsa è membro di un'organizzazione. Questa chiave della condizione globale non supporta le seguenti operazioni:
  - AWS Audit Manager
    - `auditmanager:UpdateAssessmentFrameworkShare`

- Amazon Detective
  - `detective:AcceptInvitation`
- Servizio di directory AWS
  - `ds:AcceptSharedDirectory`
- Amazon Elastic Block Store: tutte le operazioni
- Amazon EC2
  - `ec2:AcceptTransitGatewayPeeringAttachment`
  - `ec2:AcceptVpcEndpointConnections`
  - `ec2:AcceptVpcPeeringConnection`
  - `ec2:CreateTransitGatewayPeeringAttachment`
  - `ec2:CreateVpcEndpoint`
  - `ec2:CreateVpcPeeringConnection`
- Amazon EventBridge
  - `events:PutEvents`— EventBridge `PutEvents` chiamate su un event bus in un altro account, se tale event bus è stato configurato come EventBridge obiettivo per più account prima del 2 marzo 2023. Per ulteriori informazioni, consulta Concedere autorizzazioni per consentire eventi da altri AWS account nella Amazon EventBridge User Guide.
- Amazon GuardDuty
  - `guardduty:AcceptAdministratorInvitation`
- Amazon Macie
  - `macie2:AcceptInvitation`
- OpenSearch Servizio Amazon
  - `es:AcceptInboundConnection`
- Amazon Route 53
  - `route53:AssociateVpcWithHostedZone`
  - `route53:CreateVPCAssociationAuthorization`
- AWS Security Hub CSPM
  - `securityhub:AcceptAdministratorInvitation`
- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

> **Note**
>
> Per ulteriori considerazioni sulle operazioni non supportate di cui sopra, consulta il repository di esempi di policy del perimetro di dati.

`aws:ResourceOrgPaths` è una chiave di condizione multivalore. Le chiavi multivalore possono avere più di un valore nel contesto della richiesta. È necessario utilizzare gli operatori di insieme `ForAnyValue` o `ForAllValues` con gli operatori di condizione di stringa quando si utilizza questa chiave. Per ulteriori informazioni sulle chiavi di condizione multivalore, consultare Operatori dell'insieme per le chiavi di contesto multivalore.

Ad esempio, la seguente condizione restituisce `True` per le risorse appartenenti all'organizzazione `o-a1b2c3d4e5`. Quando si include un carattere jolly, è necessario utilizzare l'operatore condizionale StringLike.

```
"Condition": {
      "ForAnyValue:StringLike": {
            "aws:ResourceOrgPaths":["o-a1b2c3d4e5/*"]
   }
}
```

La condizione seguente restituisce `True` alle risorse con l'ID dell'unità organizzativa `ou-ab12-11111111`. Assocerà le risorse di proprietà degli account collegati all'unità organizzativa ou-ab12-11111111 o a qualsiasi unità organizzativa figlia.

```
"Condition": { "ForAnyValue:StringLike" : {
     "aws:ResourceOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/*"]
}}
```

La seguente condizione restituisce `True` per le risorse di proprietà degli account collegati direttamente all'ID dell'unità organizzativa `ou-ab12-22222222`, ma non alle unità figlie. L'esempio seguente utilizza l'operatore di StringEquals condizione per specificare il requisito di corrispondenza esatta per l'ID OU e non una corrispondenza con caratteri jolly.

```
"Condition": { "ForAnyValue:StringEquals" : {
     "aws:ResourceOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/"]
}}
```

> **Note**
>
> Alcuni Servizi AWS richiedono l'accesso a risorse AWS di proprietà ospitate in un altro Account AWS. L'utilizzo di `aws:ResourceOrgPaths` nelle tue policy basate sull'identità potrebbe influire sulla capacità della tua identità di accedere a queste risorse.

Alcuni AWS servizi, ad esempio AWS Data Exchange, si basano sull'accesso a risorse esterne all'utente Account AWS per le normali operazioni. Se usi la chiave `aws:ResourceOrgPaths` nelle tue policy, includi istruzioni aggiuntive per creare delle esenzioni per tali servizi. La policy AWS: nega l'accesso alle risorse Amazon S3 al di fuori del tuo account tranne AWS Data Exchange di esempio illustra come negare l'accesso in base all'account della risorsa definendo al contempo delle eccezioni per le risorse di proprietà del servizio. Puoi creare una policy simile per limitare l'accesso alle risorse all'interno di un'unita organizzativa (OU) utilizzando la chiave `aws:ResourceOrgPaths`, tenendo conto delle risorse di proprietà del servizio.

Utilizza questo esempio di policy come modello per creare le tue policy personalizzate. Per ulteriori informazioni, consulta la documentazione del servizio.

aws: ResourceOrg ID

Utilizza questa chiave per confrontare l'identificatore dell'organizzazione AWS Organizations a cui appartiene la risorsa richiesta con l'identificatore specificato nella policy.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta solo se l'account che possiede la risorsa è membro di un'organizzazione. Questa chiave della condizione globale non supporta le seguenti operazioni:
  - AWS Audit Manager
    - `auditmanager:UpdateAssessmentFrameworkShare`
  - Amazon Detective
    - `detective:AcceptInvitation`
  - Servizio di directory AWS
    - `ds:AcceptSharedDirectory`
  - Amazon Elastic Block Store: tutte le operazioni
  - Amazon EC2
    - `ec2:AcceptTransitGatewayPeeringAttachment`

    - `ec2:AcceptVpcEndpointConnections`
    - `ec2:AcceptVpcPeeringConnection`
    - `ec2:CreateTransitGatewayPeeringAttachment`
    - `ec2:CreateVpcEndpoint`
    - `ec2:CreateVpcPeeringConnection`
  - Amazon EventBridge
    - `events:PutEvents`— EventBridge `PutEvents` chiamate su un event bus in un altro account, se tale event bus è stato configurato come EventBridge obiettivo per più account prima del 2 marzo 2023. Per ulteriori informazioni, consulta Concedere autorizzazioni per consentire eventi da altri AWS account nella Amazon EventBridge User Guide.
  - Amazon GuardDuty
    - `guardduty:AcceptAdministratorInvitation`
  - Amazon Macie
    - `macie2:AcceptInvitation`
  - OpenSearch Servizio Amazon
    - `es:AcceptInboundConnection`
  - Amazon Route 53
    - `route53:AssociateVpcWithHostedZone`
    - `route53:CreateVPCAssociationAuthorization`
  - AWS Security Hub CSPM
    - `securityhub:AcceptAdministratorInvitation`
- Tipo di dati: stringa
- Tipo di valore: Single-valued

> **Note**
>
> Per ulteriori considerazioni sulle operazioni non supportate di cui sopra, consulta il repository di esempi di policy del perimetro di dati.

Questa chiave globale restituisce l'ID dell'organizzazione della risorsa per una determinata richiesta. Consente di creare regole che si applicano a tutte le risorse di un'organizzazione che sono

specificate nell'elemento `Resource` di una policy basata sull'identità. È possibile specificare l'ID organizzazione nell'elemento condizionale. Quando aggiungi e rimuovi degli account, le policy che includono la chiave `aws:ResourceOrgID` includono automaticamente anche gli account corretti e non necessitano dell'aggiornamento manuale.

Ad esempio, la seguente policy impedisce al principale di aggiungere oggetti alla risorsa `policy-genius-dev`, a meno che la risorsa Amazon S3 non appartenga alla stessa organizzazione del principale che effettua la richiesta.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza invece l'effetto `Deny`, che nega esplicitamente l'accesso a tutte le risorse elencate nell'istruzione che non appartengono all'account elencato. Utilizza questa policy in combinazione con altre policy che consentono l'accesso a risorse specifiche.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Sid": "DenyPutObjectToS3ResourcesOutsideMyOrganization",
        "Effect": "Deny",
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3:::policy-genius-dev/*",
        "Condition": {
            "StringNotEquals": {
                "aws:ResourceOrgID": "${aws:PrincipalOrgID}"
            }
        }
    }
}
```

> **Note**
>
> Alcuni Servizi AWS richiedono l'accesso a risorse AWS di proprietà ospitate in un altro Account AWS. L'utilizzo di `aws:ResourceOrgID` nelle tue policy basate sull'identità potrebbe influire sulla capacità della tua identità di accedere a queste risorse.

Alcuni AWS servizi, ad esempio AWS Data Exchange, si basano sull'accesso a risorse esterne all'utente Account AWS per le normali operazioni. Se usi la chiave `aws:ResourceOrgID` nelle tue policy, includi istruzioni aggiuntive per creare delle esenzioni per tali servizi. La policy AWS: nega l'accesso alle risorse Amazon S3 al di fuori del tuo account tranne AWS Data Exchange di esempio illustra come negare l'accesso in base all'account della risorsa definendo al contempo delle eccezioni per le risorse di proprietà del servizio. Puoi creare una policy simile per limitare l'accesso alle risorse all'interno dell'organizzazione utilizzando la chiave `aws:ResourceOrgID`, tenendo conto delle risorse di proprietà del servizio.

Utilizza questo esempio di policy come modello per creare le tue policy personalizzate. Per ulteriori informazioni, consulta la documentazione del servizio.

Nel seguente video, scopri ulteriori informazioni su come utilizzare la chiave di condizione `aws:ResourceOrgID` in una policy.

Assicurati che le identità e le reti possano essere utilizzate solo per accedere a risorse attendibili.

era:ResourceTag/tag-key

Utilizzare questa chiave per confrontare la coppia chiave-valore del tag specificata nella policy con la coppia chiave-valore associata alla risorsa. Ad esempio, puoi richiedere che l'accesso a una risorsa sia consentito solo se la risorsa dispone di una chiave di tag `"Dept"` collegata al valore `"Marketing"`. Per ulteriori informazioni, consulta Controllo dell'accesso alle AWS risorse.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando la risorsa richiesta dispone già di tag collegati o nelle richieste che creano una risorsa con un tag collegato. Questa chiave viene restituita solo per le risorse che supportano l'autorizzazione basata sui tag. È presente una chiave di contesto per ogni coppia chiave-valore del tag.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

Questa chiave di contesto è formattata `"aws:ResourceTag/tag-key":"tag-value"` dove *`tag-key`* e *`tag-value`* rappresenta una coppia chiave tag e valore. Le chiavi dei tag non prevedono una distinzione tra lettere maiuscole e minuscole. Questo significa che se specifichi `"aws:ResourceTag/TagKey1": "Value1"` nell'elemento condizione della policy, la condizione corrisponderà a una chiave di tag della risorsa denominata `TagKey1` o `tagkey1`, ma non a entrambe. I valori in queste key/value coppie di tag fanno distinzione tra maiuscole e minuscole. Questo significa che se specifichi `"aws:ResourceTag/TagKey1": "Production"` nell'elemento condizione della policy, la condizione corrisponderà a un valore di tag della risorsa denominato `Production`, ma non corrisponderà a `production` o `PRODUCTION`.

Per esempi sull'utilizzo della chiave `aws:ResourceTag` per controllare l'accesso alle risorse IAM, consulta Controllo dell'accesso alle AWS risorse.

Per esempi di utilizzo della `aws:ResourceTag` chiave per controllare l'accesso ad altre AWS risorse, vedereControllo dell'accesso alle AWS risorse tramite tag.

Per un'esercitazione sull'utilizzo della chiave di condizione `aws:ResourceTag` per il controllo degli accessi basato su attributi (ABAC), consulta Tutorial IAM: Definizione delle autorizzazioni di accesso AWS risorse basate sui tag.

Proprietà della richiesta

Utilizzare le chiavi di condizione seguenti per confrontare i dettagli della richiesta stessa e il contenuto della richiesta con le proprietà della richiesta specificate nella policy.

ha detto: CalledVia

Utilizza questa chiave per confrontare i servizi nella policy con i servizi che hanno effettuato richieste per conto del principale IAM (utente o ruolo). Quando un principale fa una richiesta a un AWS servizio, quel servizio potrebbe utilizzare le credenziali del principale per effettuare richieste successive ad altri servizi. Quando la richiesta viene effettuata utilizzando le sessioni di accesso inoltrato (FAS), questa chiave viene impostata con il valore del principale del servizio. La chiave `aws:CalledVia` contiene un elenco ordinato di ciascun servizio nella catena che ha effettuato le richieste per conto dell'entità principale.

Per ulteriori informazioni, consulta Inoltro delle sessioni di accesso.

- Disponibilità: questa chiave è presente nella richiesta quando un servizio che supporta `aws:CalledVia` utilizza le credenziali di un principale IAM per effettuare una richiesta a un altro

servizio. Questa chiave non è presente se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale. Questa chiave non è presente anche quando il principale effettua la chiamata direttamente.

- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

Per utilizzare la chiave `aws:CalledVia` condizionale in una policy, è necessario fornire i responsabili del servizio per consentire o rifiutare le richieste di servizio. AWS Ad esempio, puoi utilizzare AWS CloudFormation per leggere e scrivere da una tabella Amazon DynamoDB. DynamoDB utilizza quindi la crittografia fornita da (). AWS Key Management Service AWS KMS

Per consentire o negare l'accesso quando qualsiasi servizio effettua una richiesta utilizzando le credenziali del principale, utilizzare la chiave di condizione `aws:ViaAWSService`. Questa chiave condizionale supporta i AWS servizi.

La chiave `aws:CalledVia` è una chiave multivalore. Tuttavia, non è possibile applicare l'ordine utilizzando questa chiave in una condizione. Usando l'esempio precedente, l'utente 1 effettua una richiesta a CloudFormation, che chiama DynamoDB, che a sua volta chiama AWS KMS. Si tratta di tre richieste distinte. L'ultima chiamata a AWS KMS viene eseguita dall'Utente 1 CloudFormation e poi da DynamoDB.
Esempio di utilizzo di aws:. CalledVia

In questo caso, la chiave `aws:CalledVia` nel contesto della richiesta include `cloudformation.amazonaws.com` e `dynamodb.amazonaws.com`, in tale ordine. Se ti interessa solo che la chiamata sia stata effettuata tramite DynamoDB da qualche parte nella catena di richieste, puoi utilizzare questa chiave di condizione nella tua policy.

Ad esempio, la seguente policy consente di gestire la AWS KMS chiave denominata`my-example-key`, ma solo se DynamoDB è uno dei servizi richiedenti. L'operatore di condizione `ForAnyValue:StringEquals` assicura che DynamoDB sia uno dei servizi a effettuare chiamate. Se il principale effettua la chiamata AWS KMS direttamente, la condizione restituisce `false` e la richiesta non è consentita da questa policy.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "KmsActionsIfCalledViaDynamodb",
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/my-example-key",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "dynamodb.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Se si desidera stabilire quale servizio effettua la prima o l'ultima chiamata nella catena, è possibile utilizzare le chiavi `aws:CalledViaFirst` e `aws:CalledViaLast`. Ad esempio, la seguente politica consente di gestire la chiave denominata in. `my-example-key` AWS KMS Queste AWS KMS operazioni sono consentite solo se nella catena sono state incluse più richieste. La prima richiesta deve essere effettuata tramite CloudFormation e l'ultima tramite DynamoDB. Se altri servizi fanno richieste nel mezzo della catena, l'operazione è ancora consentita.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "KmsActionsIfCalledViaChain",
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
```

```
                "kms:ReEncrypt*",
                "kms:GenerateDataKey",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/my-example-key",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "cloudformation.amazonaws.com",
                    "aws:CalledViaLast": "dynamodb.amazonaws.com"
                }
            }
        }
    ]
}
```

Le chiavi `aws:CalledViaFirst` e `aws:CalledViaLast` sono presenti nella richiesta quando un servizio utilizza le credenziali di un'entità IAM per chiamare un altro servizio. Indicano il primo e l'ultimo servizio che ha effettuato chiamate nella catena di richieste. Ad esempio, supponiamo che CloudFormation chiami un altro servizio denominato`X Service`, che chiama DynamoDB, che quindi chiama. AWS KMS L'ultima chiamata a AWS KMS viene eseguita da `User 1` through CloudFormation`X Service`, then e quindi DynamoDB. È stato chiamato prima tramite CloudFormation e l'ultima chiamata tramite DynamoDB.
Esempio di utilizzo di aws: CalledViaFirst e aws:. CalledViaLast

aws: CalledViaFirst

Utilizza questa chiave per confrontare i servizi nella policy con il primo servizio che ha effettuato una richiesta per conto del principale IAM (utente o ruolo). Per ulteriori informazioni, consulta `aws:CalledVia`.

- Disponibilità: questa chiave è presente nella richiesta quando un servizio utilizza le credenziali di un principale IAM per effettuare almeno un'altra richiesta a un servizio diverso. Questa chiave non è presente se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale. Questa chiave non è presente anche quando il principale effettua la chiamata direttamente.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

era: CalledViaLast

Utilizza questa chiave per confrontare i servizi nella policy con l'ultimo servizio che ha effettuato una richiesta per conto del principale IAM (utente o ruolo). Per ulteriori informazioni, consulta `aws:CalledVia`.

- Disponibilità: questa chiave è presente nella richiesta quando un servizio utilizza le credenziali di un principale IAM per effettuare almeno un'altra richiesta a un servizio diverso. Questa chiave non è presente se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale. Questa chiave non è presente anche quando il principale effettua la chiamata direttamente.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

aws:ViaAWSService

Usa questa chiave per verificare se an Servizio AWS effettua una richiesta a un altro servizio per tuo conto utilizzando le sessioni di accesso diretto (FAS).

La chiave di contesto della richiesta restituisce `true` quando un servizio utilizza sessioni di accesso in avanti per effettuare una richiesta per conto del principale IAM originale. La chiave di contesto della richiesta restituisce anche `false` quando il principale effettua direttamente la chiamata.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: booleano
- Tipo di valore: Single-valued

era: ViaCustomerDomain

Usa questa chiave per verificare se un principale ha effettuato una richiesta utilizzando le credenziali ottenute da Temporary Access Delegation (TAD). Quando un partner o un fornitore ottiene credenziali temporanee tramite uno scambio di token TAD, impostiamo questa chiave sull'identificativo dell'account del partner richiedente.

Utilizzate questa chiave condizionale nelle policy di AWS Identity and Access Management (IAM) per definire i controlli di accesso specificamente alle sessioni stabilite da una delega di accesso temporanea, senza influire sulle vostre sessioni. Ad esempio, puoi utilizzare questa chiave per revocare solo le sessioni basate sulla delega preservando l'accesso diretto alla console.

- Disponibilità: questa chiave è presente nel contesto della richiesta quando un principale utilizza le credenziali ottenute tramite uno scambio di token TAD.
- Tipo di dati: stringa
- Tipo di valore: Single-valued

L'esempio seguente mostra una politica di rifiuto per le sessioni ottenute tramite una delega di accesso temporanea da un account partner specifico. La politica si applica solo alle sessioni effettuate prima di un orario specificato. Questo esempio utilizza un carattere jolly (*) per `Action` e `Resource` nell'`Deny`istruzione, che è il modello standard per revocare tutti gli accessi da una sessione specifica. Per revocare solo azioni specifiche, sostituisci il carattere jolly con l'elenco di azioni appropriato.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "DateLessThan": {
                    "aws:TokenIssueTime": "2026-07-30T01:00:37.819Z"
                },
                "StringEquals": {
                    "aws:ViaCustomerDomain": "123456789012"
                }
            }
        }
    ]
}
```

aws: AWSMCP CalledVia

Usa questa chiave per confrontare i servizi della policy con i servizi AWS MCP che hanno effettuato richieste per conto del responsabile IAM (utente o ruolo). Quando un principale effettua una richiesta a un servizio AWS MCP, tale servizio utilizza le credenziali del principale per effettuare richieste successive ad altri servizi. Quando la richiesta viene effettuata utilizzando un servizio AWS MCP, questa chiave viene impostata con il valore dell'entità del servizio. La `aws:CalledViaAWSMCP`

chiave contiene il nome principale del servizio MCP che ha effettuato le richieste per conto del principale.

- Disponibilità: questa chiave è presente nella richiesta quando un servizio AWS MCP utilizza le credenziali di un responsabile IAM per effettuare una richiesta a un servizio. AWS Questa chiave non è presente anche quando il principale effettua la chiamata direttamente.
- Tipo di dati: stringa
- Tipo di valore: Single-valued

È possibile utilizzare questa chiave di condizione per consentire o negare l'accesso in base allo specifico server MCP che ha avviato la richiesta. Ad esempio, la seguente politica nega le operazioni di eliminazione sensibili quando vengono avviate tramite un server MCP specifico:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenySensitiveActionsViaSpecificMCP",
            "Effect": "Deny",
            "Action": [
                "s3:DeleteBucket",
                "s3:DeleteObject",
                "dynamodb:DeleteTable"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaAWSMCP": "aws-mcp.amazonaws.com"
                }
            }
        }
    ]
}
```

AWS: tramite il servizio AWS MCP

Usa questa chiave per verificare se un servizio AWS MCP effettua una richiesta a un altro AWS servizio per tuo conto utilizzando le sessioni di accesso diretto (FAS). La chiave di contesto della richiesta `true` viene restituita quando un servizio AWS MCP inoltra una richiesta a un AWS servizio

per conto del principale IAM originale. La chiave di contesto della richiesta restituisce anche `false` quando il principale effettua direttamente la chiamata.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando un server AWS MCP effettua una richiesta a un AWS servizio downstream per conto di un responsabile IAM.
- Tipo di dati: booleano
- Tipo di valore: Single-valued

È possibile utilizzare questa chiave per limitare azioni specifiche quando arrivano attraverso i server MCP. Ad esempio, la seguente politica nega le operazioni di eliminazione sensibili quando vengono avviate tramite qualsiasi AWS server MCP:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenySensitiveActionsViaMCP",
            "Effect": "Deny",
            "Action": [
                "s3:DeleteBucket",
                "s3:DeleteObject",
                "dynamodb:DeleteTable"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "aws:ViaAWSMCPService": "true"
                }
            }
        }
    ]
}
```

era: CurrentTime

Utilizzare questa chiave per confrontare la data e l'ora della richiesta con la data e l'ora specificate nella policy. Per visualizzare una policy di esempio che utilizza la chiave di condizione, consulta AWS: consente l'accesso in base alla data e all'ora.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.

- Tipo di dati: data
- Tipo di valore — Single-valued

era: EpochTime

Utilizzare questa chiave per confrontare la data e l'ora della richiesta in formato epoch o ora Unix con il valore specificato nella policy. Questa chiave accetta anche il numero di secondi dal 1 gennaio 1970.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: data, numerico
- Tipo di valore — Single-valued

aws:Referer

Utilizzare questa chiave per confrontare il referrer della richiesta nel browser client con il referrer specificato nella policy. Il valore del contesto della richiesta `aws:referer` viene fornito dal chiamante in un'intestazione HTTP. L'intestazione `Referer` viene inclusa in una richiesta del browser Web quando si seleziona un link in una pagina Web. L'intestazione `Referer` contiene l'URL della pagina Web in cui è stato selezionato il link.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo se la richiesta alla AWS risorsa è stata richiamata tramite un collegamento dall'URL di una pagina Web nel browser. Questa chiave non è inclusa per le richieste a livello di programmazione perché non utilizza un link del browser per accedere alla risorsa AWS .
- Tipo di dati: stringa
- Tipo di valore: Single-valued

Ad esempio, è possibile accedere a un oggetto Amazon S3 direttamente utilizzando un URL o utilizzando l'invocazione API diretta. Per ulteriori informazioni, consulta Operazioni delle API Amazon S3 direttamente tramite un browser Web. Quando si accede a un oggetto Amazon S3 da un URL presente in una pagina Web, l'URL della pagina Web di origine viene utilizzato in `aws:referer`. Quando si accede a un oggetto Amazon S3 digitando l'URL nel browser, `aws:referer` non è presente. Quando si richiama direttamente l'API, `aws:referer` non è presente. È possibile utilizzare la chiave di condizione `aws:referer` in una policy per autorizzare le richieste effettuate da un referente specifico, ad esempio un link su una pagina Web nel dominio dell'azienda.

> **Warning**
>
> Questa chiave deve essere utilizzata con attenzione. È pericoloso includere un valore dell'intestazione del referrer pubblicamente noto. Parti non autorizzate possono utilizzare browser modificati o personalizzati per fornire qualsiasi valore `aws:referer` scelto. Di conseguenza, non `aws:referer` deve essere utilizzato per impedire a parti non autorizzate di effettuare AWS richieste dirette. È disponibile solo per consentire ai clienti di proteggere i propri contenuti digitali, come i contenuti memorizzati su Amazon S3, da riferimenti su siti di terze parti non autorizzate.

era: RequestedRegion

Usa questa chiave per confrontare la AWS regione chiamata nella richiesta con la regione specificata nella policy. È possibile utilizzare questa chiave di condizione globale per controllare quali regioni possono essere richieste. Per visualizzare le AWS regioni per ciascun servizio, consulta Endpoint e quote del servizio in. Riferimenti generali di Amazon Web Services

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: stringa
- Tipo di valore: Single-valued

I servizi globali, ad esempio IAM, dispongono di un unico endpoint. Poiché questo endpoint si trova fisicamente nella regione Stati Uniti orientali (Virginia settentrionale), le chiamate IAM vengono sempre effettuate alla regione us-east-1. Ad esempio, se crei una policy che nega l'accesso a tutti i servizi qualora la regione richiesta non fosse us-west-2 allora le chiamate IAM non andranno mai a buon fine. Per visualizzare un esempio di come ovviare a questo problema, consulta NotAction con Deny.

> **Note**
>
> La chiave di condizione `aws:RequestedRegion` consente di controllare quale endpoint di un servizio è richiamato ma non controlla l'impatto dell'operazione. Alcuni servizi hanno impatti su più regioni.
> Ad esempio, Amazon S3 dispone di operazioni API che si estendono a diverse regioni.

- È possibile richiamare `s3:PutBucketReplication` in una regione (che è interessata dalla chiave di condizione `aws:RequestedRegion`) e altre regioni vengono interessate in base alle impostazioni di configurazione delle repliche.
- Puoi richiamare `s3:CreateBucket`per creare un bucket in un'altra regione e utilizzare la `s3:LocationConstraint`chiave di condizione per controllare le regioni applicabili.

Puoi utilizzare questa chiave di contesto per limitare l'accesso ai AWS servizi all'interno di un determinato set di regioni. Ad esempio, la seguente policy consente a un utente di visualizzare tutte le istanze Amazon EC2 nella Console di gestione AWS. Tuttavia consente loro di modificare solo le istanze in Irlanda (eu-west-1), a Londra (eu-west-2) o Parigi (eu-west-3).

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "InstanceConsoleReadOnly",
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:Export*",
                "ec2:Get*",
                "ec2:Search*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "InstanceWriteRegionRestricted",
            "Effect": "Allow",
            "Action": [
                "ec2:Associate*",
                "ec2:Import*",
                "ec2:Modify*",
                "ec2:Monitor*",
                "ec2:Reset*",
                "ec2:Run*",
                "ec2:Start*",
                "ec2:Stop*",
```

```
                "ec2:Terminate*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": [
                        "eu-west-1",
                        "eu-west-2",
                        "eu-west-3"
                    ]
                }
            }
        }
    ]
}
```

era:RequestTag/tag-key

Utilizzare questa chiave per confrontare la coppia chiave-valore del tag passata nella richiesta con la coppia del tag specificata nella policy. Ad esempio, è possibile controllare che la richiesta includa la chiave del tag `"Dept"` e che abbia il valore `"Accounting"`. Per ulteriori informazioni, consulta Controllo dell'accesso durante AWS le richieste.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando le coppie chiave-valore dei tag vengono passate nella richiesta. Quando più tag vengono passati nella richiesta, è presente una chiave di contesto per ogni coppia chiave-valore dei tag.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

Questa chiave di contesto è formattata `"aws:RequestTag/`*`tag-key`*`":"`*`tag-value`*`"` dove *`tag-key`* e *`tag-value`* rappresenta una coppia chiave tag e valore. Le chiavi dei tag non prevedono una distinzione tra lettere maiuscole e minuscole. Questo significa che se specifichi `"aws:RequestTag/TagKey1": "Value1"` nell'elemento condizione della policy, la condizione corrisponderà a una chiave di tag della richiesta denominata `TagKey1` o `tagkey1`, ma non a entrambi. I valori in queste key/value coppie di tag fanno distinzione tra maiuscole e minuscole. Questo significa che se specifichi `"aws:RequestTag/TagKey1": "Production"` nell'elemento condizione della policy, la condizione corrisponderà a un valore di tag della richiesta denominato `Production`, ma non corrisponderà a `production` o `PRODUCTION`.

Questo esempio mostra che, sebbene la chiave abbia un singolo valore, è comunque possibile utilizzare più coppie chiave-valore in una richiesta se le chiavi sono diverse.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2::111122223333:instance/*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/environment": [
          "preprod",
          "production"
        ],
        "aws:RequestTag/team": [
          "engineering"
        ]
      }
    }
  }
}
```

ha detto: TagKeys

Utilizzare questa chiave per confrontare le chiavi dei tag in una richiesta con quelle specificate nella policy. Nell'utilizzo delle policy per controllare gli accessi tramite tag, è consigliabile utilizzare la chiave di condizione `aws:TagKeys` per definire le chiavi di tag ammesse. Per esempi di policy e ulteriori informazioni, consultare the section called "Controllo dell'accesso in base alle chiavi di tag"

- Disponibilità: questa chiave è inclusa nel contesto della richiesta se l'operazione supporta il passaggio di tag nella richiesta.
- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

Questa chiave di contesto è formattata `"aws:TagKeys":"`*tag-key*`"` dove *tag-key* c'è un elenco di chiavi di tag senza valori (ad esempio,`["Dept","Cost-Center"]`).

Poiché è possibile includere più coppie chiave-valore dei tag in una richiesta, il contenuto della richiesta potrebbe essere una richiesta multivalore. In questo caso, devi usare gli operatori su set `ForAllValues` o `ForAnyValue`. Per ulteriori informazioni, consulta Operatori dell'insieme per le chiavi di contesto multivalore.

Alcuni servizi supportano il tagging con operazioni sulle risorse, come la creazione, la modifica o l'eliminazione di una risorsa. Per consentire il tagging e le operazioni come chiamata singola, è necessario creare una policy che comprende le operazioni di tagging e di modifica della risorsa. È quindi possibile utilizzare la chiave di condizione `aws:TagKeys` per implementare nella richiesta specifiche chiavi di tag. Ad esempio, per limitare i tag quando un utente crea uno snapshot Amazon EC2, è necessario includere nella policy l'operazione di creazione `ec2:CreateSnapshot` e l'operazione di tagging `ec2:CreateTags`. Per visualizzare una policy per questo scenario che utilizza `aws:TagKeys`, consulta Creazione di uno snapshot con tag nella Guida per l'utente di Amazon EC2.

era: SecureTransport

Utilizzare questa chiave per verificare se la richiesta è stata inviata utilizzando TLS. Il contesto della richiesta restituisce `true` o `false`. In una policy, è possibile consentire operazioni specifiche solo se la richiesta viene inviata tramite TLS.

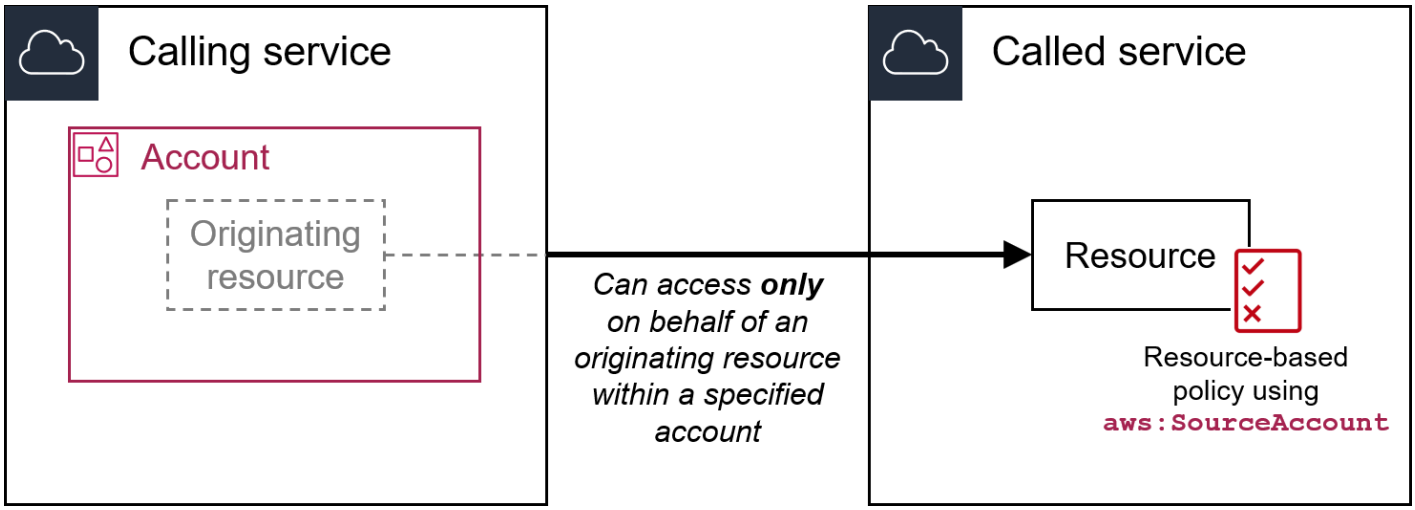
- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: booleano
- Tipo di valore — Single-valued

era: SourceAccount

Usa questa chiave per confrontare l'ID dell'account della risorsa che effettua una richiesta da servizio a servizio con l'ID dell'account specificato nella policy, ma solo quando la richiesta viene effettuata da un AWS responsabile del servizio.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando la chiamata alla risorsa viene effettuata direttamente da un principale del servizio AWS per conto di una risorsa per la quale la configurazione ha attivato la richiesta da servizio a servizio. Il servizio chiamante deve passare l'ID account della risorsa originale al servizio chiamato.

> **Note**
>
> Questa chiave fornisce un meccanismo uniforme per imporre un controllo confused deputy tra i Servizi AWS. Tuttavia, non tutte le integrazioni di servizi richiedono l'uso di questa chiave di condizione globale. Consulta la documentazione utilizzata per ulteriori informazioni sui meccanismi specifici del servizio per mitigare i rischi secondari di confusione tra i servizi. Servizi AWS



- Tipo di dati: stringa
- Tipo di valore — Single-valued

Puoi utilizzare questa chiave di condizione per assicurarti che un servizio chiamante possa accedere alla tua risorsa solo quando la richiesta proviene da un account specifico. Ad esempio, puoi allegare la seguente policy di controllo delle risorse (RCP) per negare le richieste dei principali del servizio nei confronti dei bucket Amazon S3, a meno che non siano state attivate da una risorsa nell'account specificato. Questa policy applica il controllo solo sulle richieste dei principali del servizio (`"Bool": {"aws:PrincipalIsAWSService": "true"}`) che hanno la chiave `aws:SourceAccount` (`"Null": {"aws:SourceAccount": "false"}`), in modo che le integrazioni di servizi che non richiedono l'uso di questa chiave e le chiamate da parte dei principali non vengano influenzate. Se la chiave `aws:SourceAccount` è presente nel contesto della richiesta, la condizione `Null` verrà valutata come uguale a `true`, determinando l'applicazione della chiave `aws:SourceAccount`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "RCPEnforceConfusedDeputyProtection",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceAccount": "111122223333"
        },
        "Null": {
          "aws:SourceAccount": "false"
        },
        "Bool": {
          "aws:PrincipalIsAWSService": "true"
        }
      }
    }
  ]
}
```

Nelle policy basate sulle risorse in cui il principale è un Servizio AWS principale, utilizza la chiave per limitare le autorizzazioni concesse al servizio. Ad esempio, quando un bucket Amazon S3 è configurato per inviare notifiche a un argomento Amazon SNS, il servizio Amazon S3 richiama l'operazione API `sns:Publish` per tutti gli eventi configurati. Nella policy di argomento che autorizza l'operazione `sns:Publish`, imposta il valore della chiave di condizione sull'ID account del bucket Amazon S3.

leggi: SourceArn

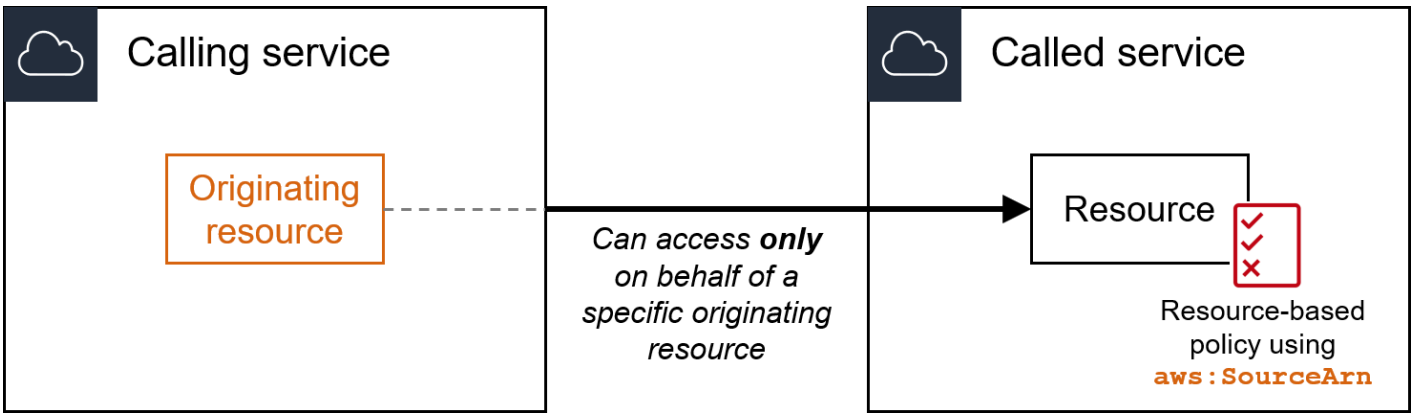
Utilizza questa chiave per confrontare il nome della risorsa Amazon (ARN) della risorsa che effettua una richiesta di servizio con l'ARN specificato nella policy, ma solo quando la richiesta è effettuata

da un principale del servizio AWS . Quando l'ARN d'origine include l'ID account, non è necessario utilizzare `aws:SourceAccount` con `aws:SourceArn`.

Questa chiave non funziona con l'ARN del principale che effettua la richiesta. Utilizza invece `era:PrincipalArn`.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando la chiamata alla risorsa viene effettuata direttamente da un principale del servizio AWS per conto di una risorsa per la quale la configurazione ha attivato la richiesta da servizio a servizio. Il servizio chiamante passa l'ARN della risorsa originale al servizio chiamato.

> **Note**
>
> Questa chiave fornisce un meccanismo uniforme per imporre un controllo confused deputy tra i Servizi AWS. Tuttavia, non tutte le integrazioni di servizi richiedono l'uso di questa chiave di condizione globale. Consulta la documentazione utilizzata per ulteriori informazioni sui meccanismi specifici dei servizi per mitigare i rischi secondari di confusione tra i servizi. Servizi AWS



- Tipo di dati: ARN

  AWS consiglia di utilizzare gli operatori ARN anziché gli operatori di stringa quando si confrontano gli ARN.

- Tipo di valore: Single-valued

Puoi utilizzare questa chiave di condizione per assicurarti che un servizio chiamante possa accedere alla tua risorsa solo quando la richiesta proviene da una risorsa specifica. Quando utilizzi una policy

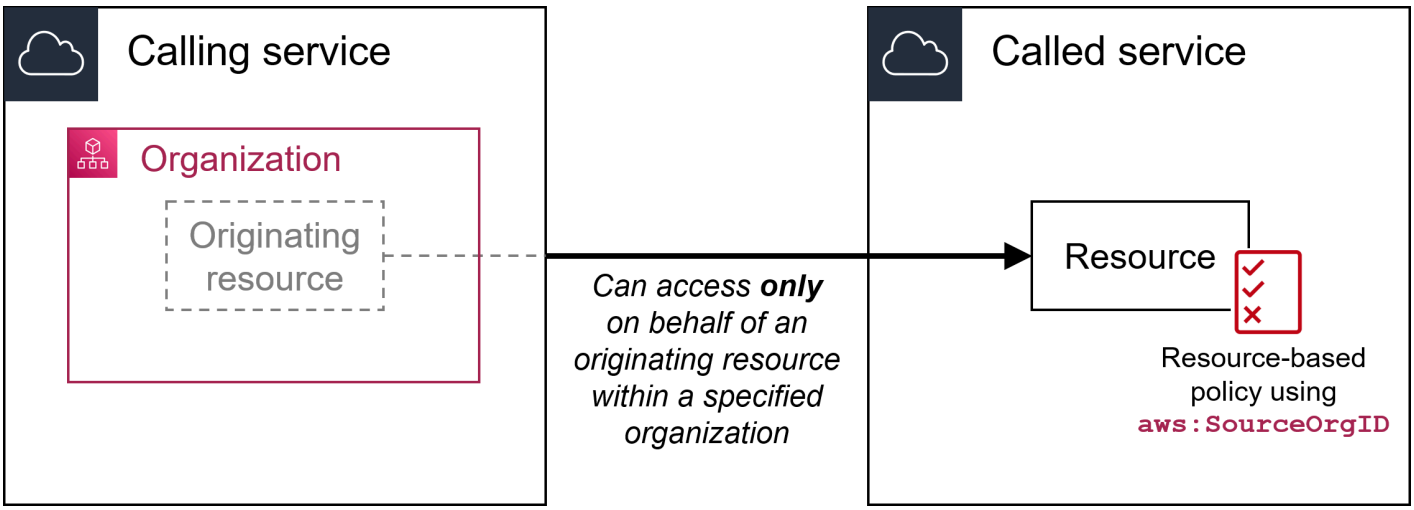
basata sulle risorse con un Servizio AWS principale come principale`Principal`, imposta il valore di questa chiave di condizione sull'ARN della risorsa a cui desideri limitare l'accesso. Ad esempio, quando un bucket Amazon S3 è configurato per inviare notifiche a un argomento Amazon SNS, il servizio Amazon S3 richiama l'operazione API `sns:Publish` per tutti gli eventi configurati. Nella policy di argomento che consente l'operazione `sns:Publish`, imposta il valore della chiave di condizione sull'ARN del bucket Amazon S3. Per i suggerimenti su quando utilizzare questa chiave di condizione nelle policy basate sulle risorse, consulta la documentazione per i Servizi AWS che stai utilizzando.

aws: ID SourceOrg

Utilizzate questa chiave per confrontare l'ID dell'organizzazione della risorsa che effettua una richiesta da servizio a servizio con l'ID dell'organizzazione specificato nella policy, ma solo quando la richiesta viene effettuata da un AWS responsabile del servizio. Quando aggiungi e rimuovi gli account da un'organizzazione in AWS Organizations, le policy che includono la chiave `aws:SourceOrgID` includono automaticamente anche gli account corretti e non necessitano dell'aggiornamento manuale.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando la chiamata alla risorsa viene effettuata direttamente da un principale del servizio AWS per conto di una risorsa di proprietà di un account membro di un'organizzazione. Il servizio chiamante passa l'ID dell'organizzazione della risorsa originale al servizio chiamato.

> **Note**
>
> Questa chiave fornisce un meccanismo uniforme per imporre un controllo confused deputy tra i Servizi AWS. Tuttavia, non tutte le integrazioni di servizi richiedono l'uso di questa chiave di condizione globale. Consulta la documentazione utilizzata per ulteriori informazioni sui meccanismi specifici del servizio per mitigare i rischi secondari di confusione tra i servizi. Servizi AWS



- Tipo di dati: stringa
- Tipo di valore: Single-valued

Puoi utilizzare questa chiave di condizione per assicurarti che un servizio chiamante possa accedere alla tua risorsa solo quando la richiesta proviene da un'organizzazione specifica. Ad esempio, puoi allegare la seguente policy di controllo delle risorse (RCP) per rifiutare le richieste dei responsabili del servizio rispetto ai bucket Amazon S3, a meno che non siano state attivate da una risorsa nell'organizzazione specificata. AWS Questa policy applica il controllo solo sulle richieste dei principali del servizio (`"Bool": {"aws:PrincipalIsAWSService": "true"}`) che hanno la chiave `aws:SourceAccount` (`"Null": {"aws:SourceAccount": "false"}`), in modo che le integrazioni di servizi che non richiedono l'uso di questa chiave e le chiamate da parte dei principali non vengano influenzate. Se la chiave `aws:SourceAccount` è presente nel contesto della richiesta, la condizione `Null` verrà valutata come uguale a `true`, determinando l'applicazione della chiave `aws:SourceOrgID`. Utilizziamo invece `aws:SourceAccount` al posto di `aws:SourceOrgID` nell'operatore di condizione `Null` in modo che il controllo si applichi ancora se la richiesta proviene da un account che non appartiene a un'organizzazione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "RCPEnforceConfusedDeputyProtection",
```

```
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceOrgID": "o-xxxxxxxxxx"
        },
        "Null": {
          "aws:SourceAccount": "false"
        },
        "Bool": {
          "aws:PrincipalIsAWSService": "true"
        }
      }
    }
  ]
}
```

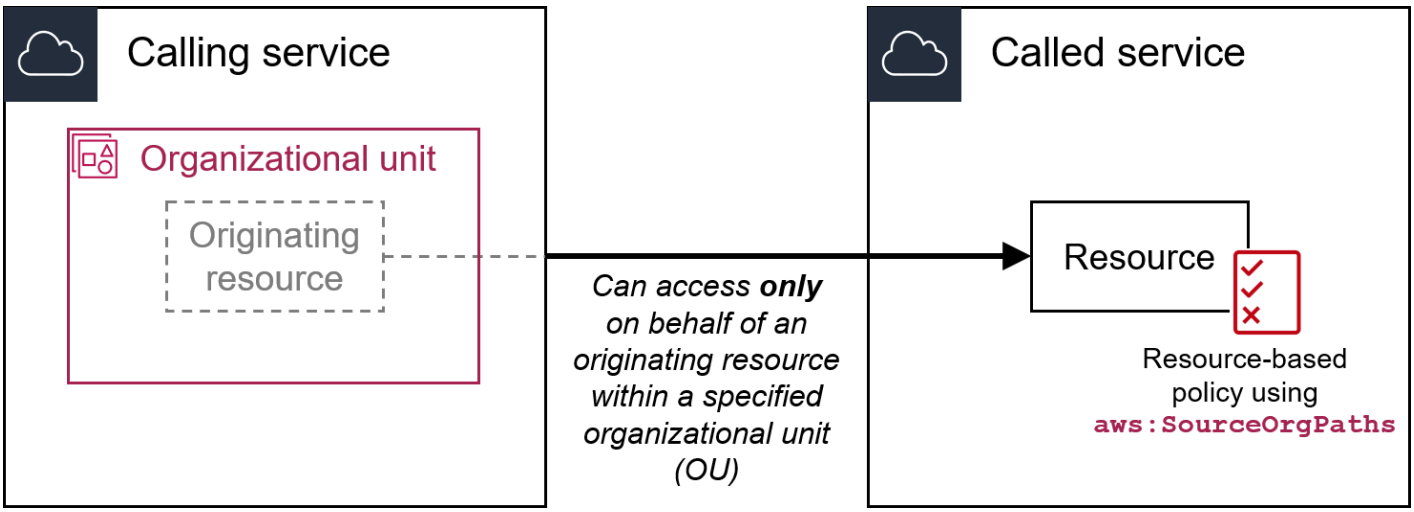
leggi: SourceOrgPaths

Usa questa chiave per confrontare il AWS Organizations percorso della risorsa che effettua una richiesta da servizio a servizio con il percorso dell'organizzazione specificato nella policy, ma solo quando la richiesta viene effettuata da un AWS responsabile del servizio. Un AWS Organizations percorso è una rappresentazione testuale della struttura di un'entità. AWS Organizations Per ulteriori informazioni sull'utilizzo e la conoscenza dei percorsi, consulta Comprendere il percorso dell'entità AWS Organizations.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando la chiamata alla risorsa viene effettuata direttamente da un principale del servizio AWS per conto di una risorsa di proprietà di un account membro di un'organizzazione. Il servizio chiamante passa il percorso dell'organizzazione della risorsa originale al servizio chiamato.

> **Note**
>
> Questa chiave fornisce un meccanismo uniforme per imporre un controllo confused deputy tra i Servizi AWS. Tuttavia, non tutte le integrazioni di servizi richiedono l'uso di questa chiave di condizione globale. Consulta la documentazione utilizzata per ulteriori

informazioni sui meccanismi specifici dei servizi per mitigare i rischi secondari di confusione tra servizi. Servizi AWS



- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

Usa questa chiave di condizione per assicurarti che un servizio chiamante possa accedere alla tua risorsa solo quando la richiesta proviene da una specifica unità organizzativa (OU) in AWS Organizations.

Analogamente a `aws:SourceOrgID`, per evitare l'impatto sulle integrazioni di servizi che non richiedono l'uso di questa chiave, utilizza l'operatore di condizione `Null` con la chiave di condizione `aws:SourceAccount` in modo che il controllo continui ad essere applicato se la richiesta proviene da un account che non appartiene a un'organizzazione.

```
{
      "Condition": {
        "ForAllValues:StringNotLikeIfExists": {
            "aws:SourceOrgPaths": "o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-
ab12-22222222/"
        },
        "Null": {
          "aws:SourceAccount": "false"
        },
        "Bool": {
          "aws:PrincipalIsAWSService": "true"
```

```
        }
      }
}
```

`aws:SourceOrgPaths` è una chiave di condizione multivalore. Le chiavi multivalore possono avere più di un valore nel contesto della richiesta. È necessario utilizzare gli operatori di insieme `ForAnyValue` o `ForAllValues` con gli operatori di condizione di stringa quando si utilizza questa chiave. Per ulteriori informazioni sulle chiavi di condizione multivalore, consultare Operatori dell'insieme per le chiavi di contesto multivalore.

era: UserAgent

Utilizzare questa chiave per confrontare l'applicazione client del richiedente con l'applicazione specificata nella policy.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: stringa
- Tipo di valore — Single-valued

> **Warning**
>
> Questa chiave deve essere utilizzata con attenzione. Poiché il valore `aws:UserAgent` viene fornito dall'intermediario in un'intestazione HTTP, le parti non autorizzate possono utilizzare browser modificati o personalizzati per fornire qualsiasi valore `aws:UserAgent` da essi scelto. Di conseguenza, non `aws:UserAgent` deve essere utilizzato per impedire a parti non autorizzate di effettuare AWS richieste dirette. Puoi utilizzarlo per consentire solo applicazioni client specifiche e solo dopo il test della policy.

era: IsMcpServiceAction

Usa questa chiave per verificare che l'azione autorizzata sia un'azione del servizio MCP. Questa chiave non si riferisce alle azioni intraprese dal servizio MCP verso altri AWS servizi.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta e impostata su True solo quando il servizio MCP autorizza un'azione del servizio MCP.
- Tipo di dati: booleano
- Tipo di valore: Single-valued

Altre chiavi di condizione cross-service

AWS STS supporta le chiavi di condizione di SAML-based federazione e le chiavi di condizione tra servizi per la federazione OIDC. Queste chiavi sono disponibili quando un utente federato tramite OIDC o SAML esegue operazioni in altri servizi. AWS

IAM e AWS STS chiavi di contesto delle condizioni

È possibile utilizzare l'`Condition`elemento in una policy JSON per testare il valore delle chiavi incluse nel contesto della richiesta di tutte le AWS richieste. Queste chiavi forniscono informazioni sulla richiesta in sé o sulle risorse a cui la richiesta fa riferimento. È possibile controllare che le chiavi abbiano determinati valori prima di consentire l'operazione richiesta dall'utente. Ciò consente un controllo granulare sulla corrispondenza o meno delle istruzioni della policy JSON rispetto a una richiesta API in ingresso. Per informazioni su come utilizzare l'elemento `Condition` in una policy JSON, consulta Elementi della policy JSON IAM: Condizione.

Questo argomento descrive le chiavi definite e fornite dal servizio IAM (con un `iam:` prefisso) e dal servizio Servizio di token di sicurezza AWS (AWS STS) (con un `sts:` prefisso). Diversi altri AWS servizi forniscono anche chiavi specifiche del servizio che sono pertinenti alle azioni e alle risorse definite da quel servizio. Per ulteriori informazioni, consulta Azioni, risorse e chiavi di condizione per AWS i servizi. La documentazione relativa a un servizio che supporta le chiavi di condizione contiene spesso ulteriori informazioni. Ad esempio, per informazioni sulle chiavi che puoi utilizzare nelle policy per le risorse Amazon S3, consulta Chiavi di policy Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

Argomenti

- Chiavi disponibili per IAM
- Chiavi disponibili per AWS Federazione OIDC
- Chiavi disponibili per SAML-based AWS STS federazione
- Cross-service SAML-based AWS STS chiavi di contesto della federazione
- Chiavi disponibili per AWS STS

Chiavi disponibili per IAM

È possibile utilizzare le seguenti chiavi di condizione nelle policy che controllano l'accesso alle risorse IAM:

Sono: AccountPropertyNamespaces

Lavora con operatori stringa.

Specifica i namespace delle proprietà dell'account che vengono letti o modificati nella richiesta. Le proprietà dell'account sono organizzate in namespace (ad esempio,). `RoleManager` Quando chiami `PutAccountProperties` o`GetAccountProperties`, utilizza questa chiave di condizione per limitare i namespace delle proprietà a cui può accedere un principale.

sono: AssociatedResourceArn

Lavora con operatori ARN.

Specifica l'ARN della risorsa a cui verrà associato questo ruolo al servizio di destinazione. La risorsa in genere appartiene al servizio a cui l'entità sta passando il ruolo. A volte, la risorsa potrebbe appartenere a un terzo servizio. Ad esempio, potresti passare un ruolo a Amazon EC2 Auto Scaling che può essere utilizzato in un'istanza di Amazon EC2. In questo caso, la condizione corrisponderebbe all'ARN dell'istanza Amazon EC2.

Questa chiave di condizione si applica solo all'PassRoleazione in una policy. Non può essere usata per limitare altre operazioni.

> **Important**
>
> Quando si utilizza la `iam:AssociatedResourceArn` condizione in una policy per limitare l'PassRoleazione, si applicano considerazioni speciali se la policy è destinata a definire l'accesso all'AddRoleToInstanceProfileazione. In questo caso, non è possibile specificare una regione o un ID istanza nell'ARN dell'istanza EC2. Il valore dell'ARN deve essere `arn:aws:ec2:*:CallerAccountId:instance/*`. L'utilizzo di qualsiasi altro valore ARN potrebbe portare a risultati di valutazione imprevisti.

Utilizza questa chiave di condizione in una policy basata su identità per consentire a un'entità di passare un ruolo, ma solo se tale ruolo è associato alla risorsa specificata. Ad esempio, è possibile consentire a un utente o a un ruolo IAM di passare qualsiasi ruolo al servizio Amazon EC2 da utilizzare con istanze nell' Account AWS. L'utente o il ruolo IAM non è autorizzato a passare ruoli ad altri servizi.

```
{
```

```
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": "ec2.amazonaws.com"
        },
        "ArnLike": {
            "iam:AssociatedResourceARN": [
                "arn:aws:ec2:*:111122223333:instance/*"
            ]
        }
    }
}
```

> **Note**
>
> AWS servizi che supportano iam: supportano PassedToService anche questa chiave di condizione.

sono: AWSServiceName

Lavora con operatori stringa.

Specifica il AWS servizio a cui è associato questo ruolo.

Questa chiave di condizione è supportata dall'operazione API `CreateServiceLinkedRole`.

> **Tip**
>
> Per informazioni sui servizi che supportano l'utilizzo di ruoli collegati ai servizi, visualizza AWS servizi supportati da IAM e cerca i servizi con Sì nella Service-Linked colonna Ruolo. Scegliere Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

In questo esempio, consenti a un'entità di creare un ruolo collegato al servizio utilizzando l'operazione API `CreateServiceLinkedRole` se il nome del servizio è access-analyzer.amazonaws.com.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
       "Effect": "Allow",
       "Action": "iam:CreateServiceLinkedRole",
       "Resource": "*",
       "Condition": {
         "StringLike": {
           "iam:AWSServiceName": "access-analyzer.amazonaws.com"
         }
       }
     }]
 }
```

Sono: FIDO-certification

Lavora con operatori stringa.

Verifica il livello di certificazione FIDO del dispositivo MFA al momento della registrazione di una chiave di sicurezza FIDO. La certificazione del dispositivo viene recuperata dal FIDO Alliance Metadata Service (MDS). Se lo stato o il livello di certificazione della chiave di sicurezza FIDO cambia, questa non verrà aggiornata a meno che la registrazione del dispositivo non sia stata annullata e poi effettuata nuovamente per recuperare le informazioni di certificazione aggiornate.

Valori possibili di L1, L1plus, L2, L2plus, L3, L3plus

In questo esempio, registri una chiave di sicurezza e recuperi la certificazione FIDO Level 1 plus per il tuo dispositivo.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
```

```
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-certification": "L1plus"
                }
            }
        }
    ]

}
```

sono: FIDO-FIPS-140-2-certification

Lavora con operatori stringa.

Verifica il livello di certificazione di FIPS-140-2 convalida del dispositivo MFA al momento della registrazione di una chiave di sicurezza FIDO. La certificazione del dispositivo viene recuperata dal FIDO Alliance Metadata Service (MDS). Se lo stato o il livello di certificazione della chiave di sicurezza FIDO cambia, questa non verrà aggiornata a meno che la registrazione del dispositivo non sia stata annullata e poi effettuata nuovamente per recuperare le informazioni di certificazione aggiornate.

Valori possibili di L1, L2, L3, L4

In questo esempio, registri una chiave di sicurezza e recuperi la certificazione di FIPS-140-2 livello 2 per il tuo dispositivo.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-2-certification": "L2"
                }
            }
        }
    ]

}
```

Sono: FIDO-FIPS-140-3-certification

Lavora con operatori stringa.

Verifica il livello di certificazione di FIPS-140-3 convalida del dispositivo MFA al momento della registrazione di una chiave di sicurezza FIDO. La certificazione del dispositivo viene recuperata dal FIDO Alliance Metadata Service (MDS). Se lo stato o il livello di certificazione della chiave di sicurezza FIDO cambia, questa non verrà aggiornata a meno che la registrazione del dispositivo non sia stata annullata e poi effettuata nuovamente per recuperare le informazioni di certificazione aggiornate.

Valori possibili di L1, L2, L3, L4

In questo esempio, registri una chiave di sicurezza e recuperi la certificazione di FIPS-140-3 livello 3 per il tuo dispositivo.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-3-certification": "L3"
                }
            }
        }
    ]

}
```

Sono: OrganizationsPolicyId

Lavora con operatori stringa.

Verifica che la policy con l' AWS Organizations ID specificato corrisponda alla policy utilizzata nella richiesta. Per visualizzare una policy IAM di esempio che utilizza la chiave di condizione, consulta IAM: visualizza le informazioni sull'ultimo accesso al servizio per una AWS Organizations policy.

sono: PassedToService

Lavora con operatori stringa.

Specifica il principale del servizio a cui un ruolo può essere passato. Questa chiave di condizione si applica solo all'PassRoleazione in una policy. Non può essere usata per limitare altre operazioni.

Quando si utilizza questa chiave di condizione in una policy, specificare il servizio utilizzando un principale del servizio. Il principale di un servizio è il nome di un servizio che può essere specificato nell'elemento `Principal` di una policy. Il formato tipico è `SERVICE_NAME_URL.amazonaws.com`.

Puoi utilizzare `iam:PassedToService` per limitare gli utenti in modo che possano passare ruoli solo a servizi specifici. Ad esempio, un utente potrebbe creare un ruolo di servizio che si fidi di CloudWatch scrivere dati di log in un bucket Amazon S3 per suo conto. L'utente deve quindi collegare una policy di autorizzazione e una policy di affidabilità al nuovo ruolo di servizio. In questo caso, la policy di affidabilità deve specificare `cloudwatch.amazonaws.com` nell'elemento `Principal`. Per visualizzare una policy che consente all'utente di passare il ruolo a CloudWatch, consulta. IAM: passa un ruolo IAM a un AWS servizio specifico

Utilizzando questa chiave di condizione, puoi assicurarti che gli utenti creino ruoli di servizio solo per i servizi specificati. Ad esempio, se un utente con la policy precedente tenta di creare un ruolo di servizio per Amazon EC2, l'operazione avrà esito negativo. L'errore si verifica perché l'utente non dispone dell'autorizzazione per trasferire il ruolo ad Amazon EC2.

A volte si passa un ruolo a un servizio che poi a sua volta lo passa a un servizio diverso. `iam:PassedToService` include solo il servizio finale che assume il ruolo, non il servizio intermedio che lo passa.

> **Note**
>
> Alcuni servizi non supportano questa chiave di condizione.

sono: PermissionsBoundary

Lavora con operatori ARN.

Verifica che la policy specificata è collegata come limite delle autorizzazioni sulla risorsa del principale IAM. Per ulteriori informazioni, consulta la sezione Limiti delle autorizzazioni per le entità IAM

iam:PolicyARN

Lavora con operatori ARN.

Controlla l'Amazon Resource Name (ARN) di una policy gestita nelle richieste che implicano una policy gestita. Per ulteriori informazioni, consulta Controllo dell'accesso alle policy.

nome: RoleTemplate ARN

Lavora con operatori ARN.

Specifica l'ARN del modello di ruolo utilizzato per creare o gestire un ruolo. Quando un AWS servizio acquisisce un ruolo da un modello di ruolo, l'ARN del modello di ruolo viene incluso come chiave di contesto nei controlli di autorizzazione per`CreateRole`,, `TagRolePutRolePolicy`, `AttachRolePolicy` e. `PutRolePermissionsBoundary` `GetRole`

sono: RegisterSecurityKey

Lavora con operatori stringa.

Verifica lo stato corrente dell'abilitazione dei dispositivi MFA.

Valori possibili di `Create` o `Activate`.

In questo esempio, registri una chiave di sicurezza e recuperi la certificazione di FIPS-140-3 livello 1 per il tuo dispositivo.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-3-certification": "L1"
                }
            }
        }
    ]

 }
```

Sono:/ResourceTag*key-name*

Lavora con operatori stringa.

Controlla che il tag collegato alla risorsa dell'identità (utente o ruolo) corrisponda al nome e al valore della chiave specificata.

> **Note**
>
> IAM e AWS STS supportano sia la chiave di condizione `iam:ResourceTag` IAM che la chiave di condizione `aws:ResourceTag` globale.

Puoi aggiungere attributi personalizzati alle risorse IAM sotto forma di coppia chiave-valore. Per ulteriori informazioni sui tag per le risorse IAM, consulta the section called "Tag per risorse IAM". Puoi utilizzare `ResourceTag` per controllare l'accesso alle risorse AWS , incluse le risorse IAM. Tuttavia, poiché IAM non supporta i tag per i gruppi, non puoi utilizzare i tag per controllare l'accesso ai gruppi.

Questo esempio mostra come creare una policy basata sull'identità che consenta di eliminare gli utenti con il tag **`status=terminated`**. Per utilizzare questa policy, sostituisci la *`italicized placeholder text`* policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "iam:DeleteUser",
        "Resource": "*",
        "Condition": {"StringEquals": {"iam:ResourceTag/status":
 "terminated"}}
    }]
}
```

sono: ServiceSpecificCredentialAgeDays

Lavora con operatori numerici.

Questa chiave di condizione limita la creazione di credenziali specifiche del servizio in base alle relative impostazioni di scadenza. Consente di controllare l'età massima, in giorni, delle credenziali specifiche del servizio che possono essere create.

L'intervallo valido per i giorni è compreso tra 1 e 36600 (minimo 1 giorno, massimo 36600 giorni).

Questa chiave di condizione è supportata dall'operazione API `CreateServiceSpecificCredential`.

In questo esempio, consenti a un utente di creare credenziali specifiche per il servizio Amazon Bedrock solo se scadono entro 90 giorni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceSpecificCredential",
            "Resource": "arn:aws:iam::111122223333:user/username",
            "Condition": {
                "StringEquals": {
                    "iam:ServiceSpecificCredentialServiceName":
 "bedrock.amazonaws.com"
                },
                "NumericLessThanEquals": {
                    "iam:ServiceSpecificCredentialAgeDays": "90"
                }
            }
        }
    ]
}
```

sono: ServiceSpecificCredentialServiceName

Lavora con operatori stringa.

Specifica quali AWS servizi possono essere utilizzati per la gestione delle credenziali specifiche del servizio. Questa chiave di condizione consente di limitare i AWS servizi consentiti durante la gestione delle credenziali specifiche del servizio.

Questa chiave di condizione è supportata dalle seguenti operazioni API:

- CreateServiceSpecificCredential
- DeleteServiceSpecificCredential
- ResetServiceSpecificCredential
- UpdateServiceSpecificCredential

I seguenti servizi sono supportati per le credenziali specifiche del servizio con la formattazione esatta dei valori:

- `bedrock.amazonaws.com`
- `cassandra.amazonaws.com`
- `codecommit.amazonaws.com`

In questo esempio, consenti a un utente di creare credenziali specifiche del servizio utilizzando l'operazione API CreateServiceSpecificCredential solo per il servizio Amazon Bedrock.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceSpecificCredential",
            "Resource": "arn:aws:iam::111122223333:user/username",
            "Condition": {
                "StringEquals": {
                    "iam:ServiceSpecificCredentialServiceName":
 "bedrock.amazonaws.com"
                }
            }
        }
    ]
}
```

sono: DelegationDuration

Lavora con operatori numerici.

Filtra l'accesso in base alla durata dell'accesso temporaneo richiesto in una richiesta di delega.

I fornitori di prodotti possono utilizzare questa chiave condizionale per controllare la durata massima consentita nelle richieste di delega inviate ai clienti. La durata è specificata in secondi

e determina per quanto tempo le credenziali temporanee rimangono valide dopo che il cliente ha rilasciato il token di scambio. Questo aiuta i fornitori di prodotti ad applicare politiche interne sui limiti di durata dell'accesso in base al loro caso d'uso.

Questa chiave di condizione è supportata dall'operazione API `CreateDelegationRequest`.

In questo esempio, è possibile creare richieste di delega solo se la durata richiesta è pari o inferiore a 7200 secondi (2 ore).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateDelegationRequest",
            "Resource": "*",
            "Condition": {
                "NumericLessThanEquals": {
                    "iam:DelegationDuration": "7200"
                }
            }
        }
    ]
}
```

sono: NotificationChannel

Lavora con operatori ARN.

Filtra l'accesso in base all'argomento ARN di Amazon SNS specificato per la ricezione delle notifiche di richiesta di delega.

I fornitori di prodotti possono utilizzare questa chiave di condizione per limitare gli argomenti SNS che possono essere utilizzati per le notifiche di richieste di delega nella chiamata API. CreateDelegationRequest I fornitori di prodotti devono specificare un argomento SNS per ricevere notifiche di modifica dello stato e scambiare token. Ciò garantisce che le notifiche vengano inviate solo ai canali approvati all'interno dell'organizzazione del fornitore del prodotto.

Questa chiave di condizione è supportata dall'operazione API `CreateDelegationRequest`.

In questo esempio, consentite la creazione di richieste di delega solo se utilizzano un argomento SNS specifico per le notifiche.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateDelegationRequest",
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "iam:NotificationChannel": "arn:aws:sns:us-
east-1:123456789012:delegation-notifications"
                }
            }
        }
    ]
}
```

Sono: TemplateArn

Lavora con operatori ARN.

Filtra l'accesso in base al modello di policy ARN utilizzato per definire le autorizzazioni in una richiesta di delega.

I fornitori di prodotti possono utilizzare questa chiave di condizione per controllare quali modelli di policy possono essere utilizzati nella CreateDelegationRequest chiamata API. I modelli di policy definiscono le autorizzazioni temporanee richieste dai fornitori di prodotti negli account dei clienti. Ciò consente ai fornitori di prodotti di limitare i modelli di policy registrati che possono essere utilizzati durante la creazione di richieste di delega.

Questa chiave di condizione è supportata dall'operazione API `CreateDelegationRequest`.

In questo esempio, è possibile creare richieste di delega solo se utilizzano un modello di policy di un dominio partner specifico.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateDelegationRequest",
            "Resource": "*",
```

```
            "Condition": {
                "ArnLike": {
                    "iam:TemplateArn": "arn:aws:iam::*:delegation-template/
partner_*"
                }
            }
        }
    ]
}
```

sono: DelegationRequestOwner

Lavora con operatori ARN.

Filtra l'accesso in base all' AWS identità o al principale proprietario della richiesta di delega.

I clienti possono utilizzare questa chiave condizionale per controllare chi può eseguire azioni sulle richieste di delega in base alla proprietà. Il proprietario di una richiesta di delega è l' AWS identità o il titolare dell'account cliente che ha avviato o ricevuto la richiesta di delega.

Questa chiave di condizione è supportata dalle seguenti operazioni API:

- `GetDelegationRequest`
- `AcceptDelegationRequest`
- `RejectDelegationRequest`
- `SendDelegationToken`
- `ListDelegationRequests`
- `UpdateDelegationRequest`

In questo esempio, si consente agli utenti di gestire solo le richieste di delega di loro proprietà.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetDelegationRequest",
                "iam:AcceptDelegationRequest",
                "iam:RejectDelegationRequest",
```

```
                "iam:SendDelegationToken",
                "iam:UpdateDelegationRequest",
                "iam:ListDelegationRequests"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "iam:DelegationRequestOwner": "${aws:PrincipalArn}"
                }
            }
        }
    ]
}
```

Chiavi disponibili per AWS Federazione OIDC

Puoi utilizzare la federazione OIDC per fornire credenziali di sicurezza temporanee agli utenti che sono stati autenticati tramite un provider di identità (IdP) compatibile con OpenID Connect (IdP) a un provider di identità IAM OpenID Connect (OIDC) nel tuo account. AWS Esempi di tali provider includono Amazon Cognito GitHub, Login with Amazon e Google. È possibile utilizzare token di identità e token di accesso del proprio IdP, nonché token di account di servizio concessi ai carichi di lavoro di Amazon Elastic Kubernetes Service.

Puoi utilizzare le chiavi di contesto delle condizioni AWS OIDC per scrivere policy che limitino l'accesso dei federated principal alle risorse associate a un provider, un'app o un utente specifici. Queste chiavi vengono in genere utilizzate nelle policy di trust di un ruolo. Definisci le chiavi di condizione utilizzando il nome del provider OIDC (`token.actions.githubusercontent.com`) seguito da un'attestazione (`:aud`): **`token.actions.githubusercontent.com:aud`**.

Alcune chiavi di condizioni di federazione OIDC possono essere utilizzate nella sessione di ruolo per autorizzare l'accesso alle risorse. Se il valore è Sì nella colonna Disponibile nella sessione, puoi utilizzare queste chiavi di condizione nelle policy per definire a quali utenti possono accedere in altri servizi. AWS Quando un claim non è disponibile nella sessione, la chiave di contesto della condizione OIDC può essere utilizzata solo in una policy di fiducia dei ruoli per l'autenticazione iniziale AssumeRoleWithWebIdentity.

Seleziona il tuo IdP per vedere in che modo le attestazioni del tuo IdP vengono mappate alle chiavi di contesto delle condizioni IAM in AWS. Ulteriori informazioni sulle chiavi per GitHub e Google sono disponibili nella scheda Predefinito.

Default

L'impostazione predefinita elenca le dichiarazioni OIDC standard e il modo in cui vengono associate alle chiavi AWS STS di contesto in. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP. Usa questa mappatura se il tuo IdP non è elencato nelle opzioni della scheda.

GitHub Actions Workflows e Google sono alcuni esempi di come utilizzano IdPs l'implementazione predefinita nel token ID JWT OIDC.

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
amr	amr	Sì
aud	azp Se non è impostato alcun valore per `azp`, la chiave di condizione `aud` corrisponde all'attestazione `aud`.	Sì
e-mail	e-mail	No
oaud	aud	No
sub	sub	Sì

Per ulteriori informazioni sull'utilizzo delle chiavi di contesto delle condizioni OIDC con GitHub, vedere. Configurazione di un ruolo per il provider di GitHub identità OIDC Per ulteriori informazioni sui campi `aud` e `azp` di Google, consulta la Guida OpenID Connect di Google Identity Platform .

amr

Lavora con operatori stringa. La chiave è multivalore, il che significa che è possibile testarla in una policy con operatori di definizione di condizioni.

Esempio: `token.actions.githubusercontent.com:amr`

Il riferimento ai metodi di autenticazione include le informazioni di accesso relative all'utente. La chiave può contenere i seguenti valori:

- Se l'utente non è autenticato, la chiave contiene solo `unauthenticated`.
- Se l'utente è autenticato, la chiave contiene il valore `authenticated` e il nome del provider di accesso utilizzato nella chiamata (`accounts.google.com`).

aud

Lavora con operatori stringa.

Esempi:

- `accounts.google.com:aud`
- `token.actions.githubusercontent.com:aud`

Utilizza la chiave di condizione `aud` per verificare che il pubblico corrisponda a quello specificato nella policy È possibile utilizzare la chiave aud con la chiave sub per lo stesso provider di identità.

Questa chiave di condizione è impostata dai seguenti campi di token:

- `aud` per gli ID client Google OAuth 2.0 dell'applicazione, quando il campo `azp` non è impostato. Quando il campo `azp` è impostato, il campo `aud` corrisponde alla chiave della condizione `accounts.google.com:oaud`.
- `azp` quando il campo `azp` è impostato. Questo può accadere per app ibride in cui un'applicazione Web e un'app Android hanno un ID client Google OAuth 2.0 diverso ma condividono lo stesso progetto delle API di Google.

Quando si scrive una policy utilizzando la chiave di condizione `accounts.google.com:aud`, occorre sapere se l'app è un'app ibrida che imposta il campo `azp`.

Campo `azp` non impostato

La policy di esempio seguente funziona per le app non ibride che non impostano il campo `azp`. In questo caso, il valore del campo `aud` del token ID di Google corrisponde a entrambi i valori della chiave di condizione `accounts.google.com:aud` e `accounts.google.com:oaud`.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {"Federated": "accounts.google.com"},
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "accounts.google.com:aud": "aud-value",
                    "accounts.google.com:oaud": "aud-value",
                    "accounts.google.com:sub": "sub-value"
                }
            }
        }
    ]
}
```

Campo `azp` impostato

La policy di esempio seguente funziona per app ibride che impostano il campo `azp`. In questo caso, il valore del campo `aud` del token ID di Google corrisponde solo al valore della chiave di condizione `accounts.google.com:oaud`. Il valore del campo `azp` corrisponde al valore della chiave di condizione `accounts.google.com:aud`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {"Federated": "accounts.google.com"},
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "accounts.google.com:aud": "azp-value",
                    "accounts.google.com:oaud": "aud-value",
                    "accounts.google.com:sub": "sub-value"
                }
            }
        }
    ]
}
```

e-mail

Lavora con operatori stringa.

Esempio: `accounts.google.com:email`

Questa chiave di condizione convalida l'indirizzo e-mail dell'utente. Il valore di questa dichiarazione potrebbe non essere univoco per questo account e potrebbe cambiare nel tempo, pertanto non dovresti utilizzare questo valore come identificatore principale per verificare il tuo record utente.

oaud

Lavora con operatori stringa.

Esempio: `accounts.google.com:oaud`

Questa chiave specifica l'altro pubblico (`aud`) a cui è rivolto questo token ID. Deve essere uno degli ID client OAuth 2.0 dell'applicazione.

sub

Lavora con operatori stringa.

Esempi:

- `accounts.google.com:sub`
- token.actions.githubusercontent.com:sub

Utilizzare queste chiavi per verificare che il soggetto corrisponda a quello specificato nella policy. È possibile utilizzare la chiave `sub` con la chiave `aud` per lo stesso provider di identità.

Nella seguente politica di fiducia dei ruoli, la chiave `sub` condizionale limita il ruolo alla GitHub filiale denominata`demo`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
```

```
        "Federated": "arn:aws:iam::111122223333:oidc-provider/
token.actions.githubusercontent.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "token.actions.githubusercontent.com:aud": "sts.amazonaws.com",
          "token.actions.githubusercontent.com:sub": "repo:org-name/repo-
name:ref:refs/heads/demo"
        }
      }
    }
  ]
}
```

Amazon Cognito

Questa scheda spiega come Amazon Cognito mappa le dichiarazioni OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP.

Per i ruoli utilizzati da Amazon Cognito, le chiavi vengono definite utilizzando `cognito-identity.amazonaws.com` seguita dall'attestazione.

Per ulteriori informazioni sulla mappatura delle attestazioni del pool di identità, consulta Mappature dei provider predefinite nella Guida per gli sviluppatori di Amazon Cognito. Per ulteriori informazioni sulla mappatura delle attestazioni del pool di utenti, consulta Utilizzo del token ID nella Guida per gli sviluppatori di Amazon Cognito.

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
amr	amr	Sì
aud	aud	Sì
oaud	aud	No
sub	sub	Sì

amr

Lavora con operatori stringa. La chiave è multivalore, il che significa che è possibile testarla in una policy con operatori di definizione di condizioni.

Esempio: `cognito-identity.amazonaws.com:amr`

Il riferimento ai metodi di autenticazione include le informazioni di accesso relative all'utente. La chiave può contenere i seguenti valori:

- Se l'utente non è autenticato, la chiave contiene solo `unauthenticated`.
- Se l'utente è autenticato, la chiave contiene il valore `authenticated` e il nome del provider di accesso utilizzato nella chiamata (`cognito-identity.amazonaws.com`).

Ad esempio, la seguente condizione nella policy di attendibilità di un ruolo Amazon Cognito verifica se l'utente non è autenticato.

```
"Condition": {
  "StringEquals":
    { "cognito-identity.amazonaws.com:aud": "us-east-2:identity-pool-id" },
  "ForAnyValue:StringLike":
    { "cognito-identity.amazonaws.com:amr": "unauthenticated" }
}
```

aud

Lavora con operatori stringa.

Esempio: `cognito-identity.amazonaws.com:aud`

Il client dell'app del pool di utenti che ha autenticato l'utente. Amazon Cognito restituisce lo stesso valore nell'attestazione `client_id` del token di accesso.

oaud

Lavora con operatori stringa.

Esempio: `cognito-identity.amazonaws.com:oaud`

Il client dell'app del pool di utenti che ha autenticato l'utente. Amazon Cognito restituisce lo stesso valore nell'attestazione `client_id` del token di accesso.

sub

Lavora con operatori stringa.

Esempio: `cognito-identity.amazonaws.com:sub`

L'identificatore univoco (UUID), o soggetto, dell'utente autenticato. Il nome utente potrebbe non essere univoco nel pool di utenti. L'attestazione sub è il modo migliore per identificare un determinato utente. È possibile utilizzare la chiave `sub` con la chiave `aud` per lo stesso provider di identità.

```
"Condition": {
         "StringEquals": {
            "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-abcd-abcd-
abcd-123456790ab",
            "cognito-identity.amazonaws.com:sub": [
               "us-east-1:12345678-1234-1234-1234-123456790ab",
               "us-east-1:98765432-1234-1234-1243-123456790ab"
            ]
         }
      }
```

Login with Amazon

Questa scheda spiega come Login with Amazon mappa le dichiarazioni di OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP.

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
app_id	ID applicazione	Sì
sub	ID utente	Sì
user_id	ID utente	Sì

app_id

Lavora con operatori stringa.

Esempio: `www.amazon.com:app_id`

Questa chiave specifica il contesto del pubblico che corrisponde al campo `aud` utilizzato da altri provider di identità.

sub

Lavora con operatori stringa.

Esempio: `www.amazon.com:sub`

Questa chiave verifica che l'ID utente corrisponda a quello specificato nella policy. È possibile utilizzare la chiave `sub` con la chiave `aud` per lo stesso provider di identità.

user_id

Lavora con operatori stringa.

Esempio: `www.amazon.com:user_id`

Questa chiave specifica il contesto del pubblico che corrisponde al campo `aud` utilizzato da altri provider di identità. È possibile utilizzare la chiave `user_id` con la chiave `id` per lo stesso provider di identità.

Facebook

Questa scheda spiega come Facebook mappa le dichiarazioni OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP.

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
app_id	ID applicazione	Sì
id	id	Sì

app_id

Lavora con operatori stringa.

Esempio: `graph.facebook.com:app_id`

Questa chiave verifica che il contesto del pubblico corrisponda al campo `aud` utilizzato da altri provider di identità.

id

Lavora con operatori stringa.

Esempio: `graph.facebook.com:id`

Questa chiave verifica che l'ID applicazione (o del sito) corrisponda a quello specificato nella policy.

GitHub

Questa scheda spiega come GitHub Actions mappa i claim OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP.

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
attore	attore	No
actor_id	actor_id	No
job_workflow_ref	job_workflow_ref	No
repository	repository	No
repository_id	repository_id	No
repository_owner_id	repository_owner_id	No
flusso di lavoro	flusso di lavoro	No
rif	rif	No
ambiente	ambiente	No
id_impresa	id_azienda	No

attore

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:actor`

Questa chiave identifica l'account personale che ha avviato l'esecuzione del flusso di lavoro. Usalo per limitare l'accesso a determinati attori.

actor_id

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:actor_id`

Questa chiave verifica l'ID dell'account personale che ha avviato l'esecuzione del flusso di lavoro. Gli Actor ID sono generati da GitHub e sono immutabili.

job_workflow_ref

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:job_workflow_ref`

Questa chiave contiene il percorso di riferimento al flusso di lavoro riutilizzabile per i lavori che utilizzano un flusso di lavoro riutilizzabile. Utilizzatela per limitare l'accesso a flussi di lavoro specifici e per assicurarvi che solo i flussi di lavoro approvati possano assumere ruoli.

repository

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:repository`

Questa chiave identifica il repository da cui viene eseguito il flusso di lavoro. Usalo per limitare l'accesso a repository specifici GitHub .

repository_id

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:repository_id`

Questa chiave verifica l'ID del repository da cui viene eseguito il flusso di lavoro. Gli ID del repository sono immutabili e non cambiano anche se il repository viene rinominato.

repository_owner_id

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:repository_owner_id`

Questa chiave verifica l'ID dell'organizzazione in cui è archiviato il repository. L'ID del proprietario del repository è un identificatore univoco e stabile che non cambia.

flusso di lavoro

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:workflow`

Questa chiave contiene il nome del flusso di lavoro. Usalo per limitare l'accesso a flussi di lavoro specifici all'interno dei tuoi repository.

rif

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:ref`

Questa chiave identifica il git ref (ramo o tag) che ha attivato l'esecuzione del flusso di lavoro. Usalo per limitare l'accesso in base a rami specifici, ad esempio permettendo solo `main` o `production` rami.

environment

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:environment`

Questa chiave contiene il nome dell'ambiente utilizzato dal job. Utilizzatela per implementare controlli di accesso basati sull'ambiente, ad esempio autorizzazioni separate per gli ambienti di sviluppo, staging e produzione.

> **Note**
>
> Se l'attestazione relativa all'ambiente è inclusa nella politica di fiducia, è necessario configurare e fornire un ambiente nel flusso di lavoro. GitHub

enterprise_id

Lavora con operatori stringa.

Esempio: `token.actions.githubusercontent.com:enterprise_id`

Questa chiave verifica l'ID dell'azienda che contiene il repository da cui viene eseguito il flusso di lavoro. Utilizzala per assicurarti che l'accesso sia limitato ai repository all'interno della tua GitHub organizzazione Enterprise.

> **Usa identificatori immutabili, non nomi**
>
> Sì GitHub, i nomi del repository, dell'organizzazione e degli utenti possono cambiare. Un nome che viene liberato rinominando o eliminando può essere richiesto da un altro account. Le policy che si basano esclusivamente su dichiarazioni mutabili basate sul nome (come `repository` `oactor`) potrebbero consentire l'accesso a identità indesiderate. Per evitare che ciò accada, utilizzate gli identificatori immutabili GitHub forniti, ad esempio, o, nel vostro ruolo `repository_idrepository_owner_id`, le politiche di fiducia o le politiche di controllo delle `actor_id` risorse.

Il seguente esempio di politica di fiducia utilizza attestazioni personalizzate nel token GitHub OIDC per limitare l'accesso a un ruolo.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Federated": "arn:aws:iam::111122223333:oidc-provider/
token.actions.githubusercontent.com"
         },
         "Action": "sts:AssumeRoleWithWebIdentity",
         "Condition": {
            "StringEquals": {
                "token.actions.githubusercontent.com:aud": "sts.amazonaws.com",
                "token.actions.githubusercontent.com:repository_owner_id": "123456",
                "token.actions.githubusercontent.com:repository_id": "1296269",
                "token.actions.githubusercontent.com:actor_id": "1",
                "token.actions.githubusercontent.com:ref": "refs/heads/main",
```

```
                "token.actions.githubusercontent.com:enterprise_id": "345"
              }
            }
        }
    ]
}
```

Google

Questa scheda spiega in che modo Google mappa le dichiarazioni OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP.

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
google/organization_number	google:numero_organizzazione	No

google/organization_number

Lavora con operatori numerici.

Esempio: `accounts.google.com:google/organization_number`

Questa chiave verifica che un token rappresenti un'identità Google appartenente a una specifica organizzazione Google Cloud o Google Workspace. Utilizzala per limitare l'accesso agli utenti di organizzazioni specifiche, assicurando che solo le identità della tua organizzazione possano assumere il ruolo.

L'esempio seguente di politica di fiducia utilizza l'`google/organization_number`attestazione per limitare l'accesso a un ruolo.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
```

```
            "Federated": "accounts.google.com"
         },
         "Action": "sts:AssumeRoleWithWebIdentity",
         "Condition": {
            "NumericEquals": {
                "accounts.google.com:google/organization_number": "123456"
               }
           }
        }
    ]
}
```

CircleCI

Questa scheda spiega come CircleCI mappa i claim OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP.

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
oidc.circleci. com/project-id	oidc.circleci. com/project-id	No

oidc.circleci. com/project-id

Lavora con operatori stringa.

Esempio: `circleci-issuer-url:oidc.circleci.com/project-id`

Questa chiave identifica il progetto CircleCI in cui il lavoro è in esecuzione. Il suo valore è una stringa contenente un UUID che identifica in modo univoco il progetto CircleCI. Usalo per limitare l'accesso a specifici progetti CircleCI.

Il seguente esempio di politica di fiducia utilizza il `oidc.circleci.com/project-id` claim per limitare l'accesso a un ruolo.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::123456789012:oidc-provider/oidc.circleci.com/
org/12345"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.circleci.com/org/12345:aud": "sts.amazonaws.com",
          "oidc.circleci.com/org/12345:oidc.circleci.com/project-id": "76543210-
ba98-fedc-3210-edcba0987654"
        }
      }
    }
  ]
}
```

Oracle Cloud Infrastructure (OCI)

Questa scheda spiega come Oracle Cloud Infrastructure mappa i claim OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP.

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
rpst_id	rpst_id	No

rpst_id

Lavora con operatori stringa.

Esempio: `oci-issuer-url:rpst_id`

Questa chiave identifica in modo univoco la risorsa principale in OCI. Usalo per limitare l'accesso a risorse principali OCI specifiche. Il rpst_id (Resource Principal Session Token ID) fornisce un identificatore stabile per l'autenticazione basata su risorse OCI.

L'esempio seguente di policy di fiducia utilizza l'attestazione per limitare l'accesso a un `rpst_id` ruolo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::123456789012:oidc-provider/idcs-
abc123ef5678901234abcd.identity.oraclecloud.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "idcs-abc123ef5678901234abcd.identity.oraclecloud.com:aud":
 "sts.amazonaws.com",
          "idcs-abc123ef5678901234abcd.identity.oraclecloud.com:rpst_id": "your-
rpst-id"
        }
      }
    }
  ]
}
```

GitLab.com

Questa scheda spiega come GitLab.com CI/CD mappa i claim OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP.

> **Note**
>
> Queste chiavi di condizione personalizzate sono supportate solo quando si utilizza l'istanza SaaS GitLab.com multi-tenant come provider di identità OIDC. Non è possibile utilizzare queste chiavi di condizione con istanze dedicate o. GitLab GitLab Self-Managed

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
namespace_id	namespace_id	No
project_id	project_id	No
user_id	user_id	No
user_login	user_login	No
user_email	user_email	No
livello_accesso utente	livello_accesso utente	No
ref_protected	ref_protected	No
pipeline_source	sorgente_pipeline	No
ambiente_runner	ambiente_runner	No

namespace_id

Lavora con operatori stringa.

Esempio: `gitlab.com:namespace_id`

Questa chiave identifica l'ID dello spazio dei nomi (gruppo) del progetto che esegue il job. CI/CD L'ID dello spazio dei nomi è un identificatore stabile e univoco che non cambia se il gruppo viene rinominato.

project_id

Lavora con operatori stringa.

Esempio: `gitlab.com:project_id`

Questa chiave identifica l'ID del progetto che esegue il job. CI/CD L'ID del progetto è un identificatore univoco e stabile che non cambia se il progetto viene rinominato o spostato.

user_id

Lavora con operatori stringa.

Esempio: `gitlab.com:user_id`

Questa chiave identifica l'ID dell'utente che esegue il lavoro. CI/CD L'ID utente è un identificatore univoco e stabile.

user_login

Lavora con operatori stringa.

Esempio: `gitlab.com:user_login`

Questa chiave identifica il nome utente dell'utente che esegue il job. CI/CD I nomi utente possono essere modificati; per un identificatore stabile, utilizzali invece. `user_id`

user_email

Lavora con operatori stringa.

Esempio: `gitlab.com:user_email`

Questa chiave identifica l'indirizzo email dell'utente che esegue il job. CI/CD Il valore di questa dichiarazione potrebbe cambiare nel tempo; per un identificatore stabile, `user_id` utilizzalo invece.

user_access_level

Lavora con operatori stringa.

Esempio: `gitlab.com:user_access_level`

Questa chiave identifica il livello di accesso dell'utente all'interno del progetto (ad esempio,,). `maintainer` `developer` `owner`

ref_protected

Lavora con operatori stringa.

Esempio: `gitlab.com:ref_protected`

Questa chiave indica se il riferimento Git che ha attivato il processo è protetto. Il valore è `true` se protetto, `false` altrimenti.

pipeline_source

Lavora con operatori stringa.

Esempio: `gitlab.com:pipeline_source`

Questa chiave identifica la fonte che ha attivato la pipeline (ad esempio,,,,`push`). `web` `schedule` `api` `merge_request_event`

runner_environment

Lavora con operatori stringa.

Esempio: `gitlab.com:runner_environment`

Questa chiave identifica il tipo di runner utilizzato dal lavoro. CI/CD

> **Important**
>
> In modalità GitLab.com SaaS, i percorsi di gruppo o progetto eliminati in precedenza potevano essere ricreati da utenti diversi, il che poteva far sì che le policy basate esclusivamente su dichiarazioni basate sui percorsi (ad esempio`sub`) concedessero l'accesso a identità indesiderate. A partire dal 1 giugno 2026, GitLab ha implementato un aggiornamento lato piattaforma che impedisce l'emissione di nuovi token ID CI per percorsi di progetto che avevano precedentemente emesso token, riducendo questo rischio in futuro. Come ulteriore livello di protezione, GitLab consigliamo di includere condizioni relative a identificatori univoci AWS e stabili, come `namespace_id` o, nelle politiche di fiducia o nelle politiche di controllo delle `project_id` risorse.

L'esempio seguente di politica di fiducia utilizza il `namespace_id` claim insieme `sub` a per limitare l'accesso a un GitLab.com progetto e a una filiale specifici.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::123456789012:oidc-provider/gitlab.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "gitlab.com:sub": "project_path:mygroup/
myproject:ref_type:branch:ref:main",
          "gitlab.com:namespace_id": "12345"
```

```
        }
      }
    }
  ]
}
```

Buildkite

Questa scheda spiega come Buildkite mappa i claim OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi di condizioni AWS STS con i valori nella colonna delle attestazioni di JWT dell'IdP.

AWS STS chiave di condizione	Attestazione JWT IdP	Disponibile nella sessione
organization_slug	organizzazione_slug	No
organization_id	organization_id	No
pipeline_slug	pipeline_slug	No
pipeline_id	id_pipeline	No
nome_cluster	nome_cluster	No
cluster_id	cluster_id	No
build_branch	build_branch	No

organizzazione_slug

Lavora con operatori stringa.

Esempio: `agent.buildkite.com:organization_slug`

Questa chiave identifica l'organizzazione Buildkite da cui viene eseguito il job. Usalo per limitare l'accesso a organizzazioni specifiche.

organization_id

Lavora con operatori stringa.

Esempio: `agent.buildkite.com:organization_id`

Questa chiave verifica l'ID dell'organizzazione Buildkite da cui viene eseguito il lavoro. Gli ID dell'organizzazione sono generati da Buildkite e sono immutabili.

pipeline_slug

Lavora con operatori stringa.

Esempio: `agent.buildkite.com:pipeline_slug`

Questa chiave identifica la pipeline in cui il processo è in esecuzione. Usalo per limitare l'accesso a specifiche pipeline Buildkite.

pipeline_id

Lavora con operatori stringa.

Esempio: `agent.buildkite.com:pipeline_id`

Questa chiave verifica l'ID della pipeline in cui il processo è in esecuzione. Gli ID della pipeline sono generati da Buildkite e sono immutabili.

cluster_name

Lavora con operatori stringa.

Esempio: `agent.buildkite.com:cluster_name`

Questa chiave identifica il cluster su cui è in esecuzione il job. Usalo per limitare l'accesso a cluster Buildkite specifici.

cluster_id

Lavora con operatori stringa.

Esempio: `agent.buildkite.com:cluster_id`

Questa chiave verifica l'ID del cluster su cui è in esecuzione il job. Gli ID del cluster sono generati da Buildkite e sono immutabili.

build_branch

Lavora con operatori stringa.

Esempio: `agent.buildkite.com:build_branch`

Questa chiave identifica il ramo git che ha attivato la compilazione. Usalo per limitare l'accesso in base a rami specifici, ad esempio permettendo solo `main` o `production` rami.

> **Usa identificatori immutabili, non slug**
>
> Su Buildkite, un'organizzazione o uno slug di pipeline eliminato può essere immediatamente nuovamente registrato da un'altra organizzazione o pipeline. Buildkite emette token OIDC per questi slug dallo stesso emittente (). `agent.buildkite.com` Di conseguenza, le policy che si basano esclusivamente su dichiarazioni basate sugli slugs (come `organization_slug` or) potrebbero consentire l'accesso a identità indesiderate. `pipeline_slug` Per proteggerti da questo problema, utilizza identificatori stabili e immutabili, come `organization_id` or, nel tuo ruolo, politiche di fiducia o politiche di `pipeline_id` controllo delle risorse.

Il seguente esempio di politica di fiducia utilizza attestazioni personalizzate nel token OIDC Buildkite per limitare l'accesso a un ruolo.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Federated": "arn:aws:iam::123456789012:oidc-provider/
agent.buildkite.com"
        },
        "Action": "sts:AssumeRoleWithWebIdentity",
        "Condition": {
            "StringEquals": {
                "agent.buildkite.com:aud": "sts.amazonaws.com",
                "agent.buildkite.com:organization_id": "0191e7a0-1234-7abc-
def0-123456789abc",
                "agent.buildkite.com:pipeline_id": "0191e7a1-5678-7abc-
def0-123456789abc",
                "agent.buildkite.com:build_branch": "main"
            }
        }
```

```
    }
    ]
}
```

Ulteriori informazioni sulla federazione OIDC

- Guida per l'utente di Amazon Cognito
- Federazione OIDC

Chiavi disponibili per SAML-based AWS STS federazione

Se si utilizza la SAML-based federazione utilizzando Servizio di token di sicurezza AWS (AWS STS), è possibile includere chiavi di condizione aggiuntive nella policy.

Policy di affidabilità di un ruolo SAML

Nella policy di affidabilità di un ruolo è possibile includere le chiavi seguenti, che consentono di stabilire se il chiamante è autorizzato ad assumere il ruolo. Salvo per `saml:doc`, tutti i valori sono derivati dall'asserzione SAML. Tutti gli elementi nell'elenco sono disponibili nell'editor visivo della console IAM quando crei o modifichi una policy con condizioni. Gli elementi contrassegnati con `[]` possono avere un valore che è un elenco del tipo specificato.

saml:aud

Lavora con operatori stringa.

L'URL di un endpoint a cui vengono presentate le asserzioni SAML. Il valore di questa chiave proviene dal campo `SAML Recipient` dell'asserzione, non dal campo `Audience`.

saml:commonName[]

Lavora con operatori stringa.

Questo è un attributo `commonName`.

saml:cn[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:doc

Lavora con operatori stringa.

Rappresenta il principale utilizzato per assumere il ruolo. Il formato è*account-ID*/*provider-friendly-name*, ad esempio`123456789012/SAMLProviderName`. Il valore ID account si riferisce all'account proprietario del provider SAML.

saml:edupersonaffiliation[]

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonassurance[]

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonentitlement[]

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonnickname[]

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonorgdn

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonorgunitdn[]

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonprimaryaffiliation

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonprimaryorgunitdn

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonprincipalname

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonscopedaffiliation[]

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersontargetedid[]

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:eduorghomepageuri[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:eduorgidentityauthnpolicyuri[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:eduorglegalname[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:eduorgsuperioruri[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:eduorgwhitepagesuri[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:givenName[]

Lavora con operatori stringa.

Questo è un attributo `givenName`.

saml:iss

Lavora con operatori stringa.

L'approvatore, che è rappresentato da un URN.

saml:mail[]

Lavora con operatori stringa.

Questo è un attributo `mail`.

saml:name[]

Lavora con operatori stringa.

Questo è un attributo `name`.

saml:namequalifier

Lavora con operatori stringa.

Un valore hash basato sul nome descrittivo del provider SAML. Il valore è la concatenazione dei seguenti valori, in ordine e separati da un carattere '/':

1. Il valore di risposta `Issuer` (`saml:iss`)
2. L'ID dell'account `AWS`
3. Il nome descrittivo (l'ultima parte dell'ARN) del provider SAML in IAM

La concatenazione dell'ID account e del nome descrittivo del provider SAML è disponibile per le policy IAM sotto forma di chiave `saml:doc`. Per ulteriori informazioni, consulta Identificazione univoca degli SAML-based utenti nella federazione.

saml:organizationStatus[]

Lavora con operatori stringa.

Questo è un attributo `organizationStatus`.

saml:primaryGroupSID[]

Lavora con operatori stringa.

Questo è un attributo `primaryGroupSID`.

saml:sub

Lavora con operatori stringa.

Questo è l'oggetto della richiesta, che include un valore che identifica in modo univoco un singolo utente in un'organizzazione (ad esempio, `_cbb88bf52c2510eabe00c1642d4643f41430fe25e3`).

saml:sub_type

Lavora con operatori stringa.

Questa chiave può avere il valore `persistent` o `transient` oppure consistere dell'URI `Format` completo, tratto dagli elementi `Subject` e `NameID` utilizzati nell'asserzione SAML. Il valore `persistent` indica che il valore in `saml:sub` è lo stesso per un utente da una sessione all'altra. Se il valore è `transient`, l'utente dispone di un valore `saml:sub` diverso per ogni sessione. Per ulteriori informazioni sull'attributo `Format` dell'elemento `NameID`, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

saml:surname[]

Lavora con operatori stringa.

Questo è un attributo `surnameuid`.

saml:uid[]

Lavora con operatori stringa.

Questo è un attributo `uid`.

saml:x500 [] UniqueIdentifier

Lavora con operatori stringa.

Questo è un attributo `x500UniqueIdentifier`.

Per informazioni generali sugli attributi `eduPerson` ed `eduOrg`, consulta il sito Web REFEDS. Per un elenco di `eduPerson` attributi, consulta la specifica della classe di oggetti eduPerson (201602).

Le chiavi di condizione il cui tipo è un elenco possono includere più valori. Per creare condizioni nelle policy per valori con elenchi, è possibile utilizzare gli operatori di definizione (`ForAllValues`, `ForAnyValue`). Ad esempio, per consentire l'accesso a qualsiasi utente la cui affiliazione è "facoltà" o "staff" (ma non "studente") è possibile utilizzare una condizione come la seguente:

```
"Condition": {
   "ForAllValues:StringLike": {
     "saml:edupersonaffiliation":[ "faculty", "staff"]
   }
}
```

Cross-service SAML-based AWS STS chiavi di contesto della federazione

Alcune chiavi di condizione di SAML-based federazione possono essere utilizzate nelle richieste successive per autorizzare AWS le operazioni in altri servizi e `AssumeRole` chiamate. Queste sono le seguenti chiavi di condizione che possono essere utilizzate nelle politiche di fiducia dei ruoli quando i principali federati assumono un altro ruolo e nelle politiche delle risorse di altri AWS servizi per autorizzare l'accesso alle risorse da parte dei responsabili federati. Per ulteriori informazioni sull'utilizzo di queste chiavi, consulta Informazioni sulla federazione basata su SAML 2.0.

Seleziona una chiave di condizione per visualizzarne la descrizione.

- saml:namequalifier
- saml:sub
- saml:sub_type

> **Note**
>
> Nessun'altra chiave di condizione di SAML-based federazione è disponibile per l'uso dopo la risposta iniziale di autenticazione del provider di identità esterno (IdP).

Chiavi disponibili per AWS STS

È possibile utilizzare le seguenti chiavi di condizione nelle politiche di fiducia dei ruoli IAM per i ruoli che vengono assunti utilizzando le operazioni Servizio di token di sicurezza AWS (AWS STS).

saml:sub

Lavora con operatori stringa.

Questo è l'oggetto della richiesta, che include un valore che identifica in modo univoco un singolo utente in un'organizzazione (ad esempio, `_cbb88bf52c2510eabe00c1642d4643f41430fe25e3`).

set: AWSServiceName

Lavora con operatori stringa.

Utilizzare questa chiave per specificare il servizio in cui è possibile utilizzare un token al portatore. Quando si utilizza questa chiave di condizione in una policy, specificare il servizio utilizzando un principale del servizio. Il principale di un servizio è il nome di un servizio che può essere specificato nell'elemento `Principal` di una policy. Ad esempio, `codeartifact.amazonaws.com` è il principale del AWS CodeArtifact servizio.

Disponibilità: questa chiave è presente nelle richieste che ottengono un token di connessione. Non è possibile effettuare una chiamata diretta per AWS STS ottenere un token al portatore. Quando si eseguono alcune operazioni in altri servizi, il servizio richiede il token del portatore per conto dell'utente.

Alcuni AWS servizi richiedono l'autorizzazione per ottenere un AWS STS service bearer token prima di poter accedere alle loro risorse a livello di codice. Ad esempio, AWS CodeArtifact richiede che le entità utilizzino token portatori per eseguire alcune operazioni. Il comando `aws codeartifact get-authorization-token` restituisce un token di connessione. È quindi possibile utilizzare il token al portatore per eseguire operazioni. AWS CodeArtifact Per ulteriori informazioni sui token del portatore, vedere Token di connessione al servizio.

È possibile utilizzare questa chiave di condizione per consentire ai principal di ottenere un token di portatore da utilizzare con un servizio specifico.

st: DurationSeconds

Lavora con operatori numerici.

Usa questa chiave per specificare la durata (in secondi) che un principale può utilizzare per ottenere un token al AWS AWS STS portatore o un token Web JSON dall'GetWebIdentityTokenAPI.

Disponibilità: questa chiave è presente nelle richieste che ottengono un token al portatore o un token Web JSON dall'API. GetWebIdentityToken Non è possibile effettuare una chiamata diretta

a per ottenere un AWS STS token al portatore. Quando si eseguono alcune operazioni in altri servizi, il servizio richiede il token del portatore per conto dell'utente. La chiave non è applicabile alle operazioni che prevedono AWS STS un ruolo.

Alcuni AWS servizi richiedono l'autorizzazione per ottenere un AWS STS service bearer token prima di poter accedere alle relative risorse a livello di programmazione. Ad esempio, AWS CodeArtifact richiede che le entità utilizzino token portatori per eseguire alcune operazioni. Il comando `aws codeartifact get-authorization-token` restituisce un token di connessione. È quindi possibile utilizzare il token al portatore per eseguire operazioni. AWS CodeArtifact Per ulteriori informazioni sui token del portatore, vedere Token di connessione al servizio.

st: IdentityTokenAudience

Lavora con operatori stringa.

Usa questa chiave per specificare il pubblico per il quale un principal IAM può richiedere JSON Web Tokens (JWT) utilizzando l'API. GetWebIdentityToken Quando questa chiave di condizione è presente in una policy IAM, i principali IAM possono richiedere token solo per i segmenti di pubblico specificati nella policy. I servizi esterni convalidano la dichiarazione relativa all'audience («aud») contenuta nel JSON Web Token per assicurarsi che il token sia destinato a loro.

Disponibilità: questa chiave è presente nelle richieste all' GetWebIdentityToken API utilizzata per ottenere JSON Web Tokens (JWT) per l'autenticazione con servizi esterni.

Quando utilizzi questa chiave condizionale in una policy, specifica il valore del pubblico che corrisponde all'identificatore del destinatario previsto (ad esempio,). https://api.example.com

La seguente policy di esempio consente a un principale di richiedere token per i servizi esterni specificati:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:GetWebIdentityToken",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "sts:IdentityTokenAudience": [
```

```
                    "https://api2.example.com",
                    "https://api1.example.com"
                ]
            }
        }
    ]
}
```

set: SigningAlgorithm

Lavora con operatori stringa.

Usa questa chiave per specificare l'algoritmo crittografico AWS AWS STS utilizzato per firmare i JSON Web Tokens (JWT) generati dall'API. GetWebIdentityToken Quando utilizzate questa chiave di condizione in una policy, specificate ES384 (ECDSA con P-384 curva e) o RS256 (RSA con). SHA-384 SHA-256

Disponibilità: questa chiave è presente nelle richieste all' GetWebIdentityToken API utilizzata per ottenere JSON Web Tokens (JWT) per l'autenticazione con servizi esterni.

Puoi utilizzare questa chiave condizionale per far sì che i principali IAM richiedano i token utilizzando algoritmi di firma compatibili con i tuoi requisiti di sicurezza o con i servizi esterni con cui ti integri. L'ES384 offre sicurezza e prestazioni ottimali, mentre RS256 offre una più ampia compatibilità con i sistemi che non supportano l'ECDSA.

La seguente politica di esempio richiede che i principali utilizzino l'algoritmo di firma ES384:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:GetWebIdentityToken",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "sts:SigningAlgorithm": "ES384"
                }
            }
        }
    ]
```

```
}
```

set: ExternalId

Lavora con operatori stringa.

Utilizza questa chiave per richiedere che un'entità principale fornisca un identificatore specifico quando assume un ruolo IAM.

Disponibilità: questa chiave è presente nella richiesta quando il principale fornisce un ID esterno mentre assume un ruolo utilizzando l' AWS API AWS CLI or.

Un identificatore univoco che può essere richiesto quando assumi un ruolo in un altro account. Se l'amministratore dell'account a cui appartiene il ruolo ha fornito un ID esterno, specifica questo valore nel parametro `ExternalId`. Questo valore può essere qualsiasi stringa, ad esempio una passphrase o un numero di account. La funzione principale dell'ID esterno è quella di risolvere e prevenire il problema del "confused deputy" (delegato confuso). Per ulteriori informazioni sull'ID esterno e il problema del "confused deputy", consulta Accesso a Account AWS di proprietà di terzi.

Il valore `ExternalId` deve avere un minimo di 2 caratteri e un massimo di 1.224 caratteri. Il valore deve essere alfanumerico senza spazi. Può anche includere i seguenti simboli: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), due punti (:), barra (/) e trattino (-).

sts:RequestContext//context-key

Lavora con operatori stringa.

Utilizza questa chiave per confrontare le coppie chiave-valore del contesto di sessione incorporate nell'asserzione di contesto firmata dall'emittente del token affidabile passata nella richiesta con i valori chiave-valore del contesto specificati nella policy di attendibilità del ruolo.

Disponibilità: questa chiave è presente nella richiesta quando viene fornita un'asserzione di contesto nel parametro di `ProvidedContexts` richiesta mentre si assume un ruolo utilizzando l'operazione API. AWS STS AssumeRole

Questa chiave di contesto è formattata come `"sts:RequestContext/context-key":"context-value"` dove `context-key` e `context-value` rappresentano una coppia chiave-valore di contesto. Quando più chiavi di contesto sono incorporate nell'asserzione di contesto firmata passata nella richiesta, è presente una chiave di contesto per ogni coppia chiave-valore. È necessario concedere l'autorizzazione per l'azione `sts:SetContext` nella policy di

attendibilità del ruolo per consentire a un principale di impostare le chiavi di contesto all'interno del token di sessione risultante. Per ulteriori informazioni sulle chiavi di contesto del Centro identità IAM supportate che possono essere utilizzate con questa chiave, consulta Chiavi di condizione AWS STS per il Centro identità IAM nella Guida per l'utente di Centro identità AWS IAM .

È possibile utilizzare questa chiave in una policy di attendibilità del ruolo per applicare un controllo di accesso granulare in base all'utente o ai suoi attributi quando assume un ruolo. Dopo l'assunzione del ruolo, l'attività viene visualizzata nei AWS CloudTrail log all'interno dell'`AdditionalEventData`attributo, contenenti le coppie chiave-valore del contesto di sessione impostate dal provider di contesto nella richiesta di assunzione del ruolo. Ciò consente agli amministratori di distinguere tra le sessioni di ruolo quando un ruolo viene utilizzato da principali diversi. Le coppie chiave-valore vengono impostate dal provider di contesto specificato, non da o. AWS CloudTrail AWS STS Ciò consente al provider di contesto di controllare quale contesto è incluso nei CloudTrail log e nelle informazioni sulla sessione.

set: RequestContextProviders

Lavora con operatori ARN.

Utilizza questa chiave per confrontare l'ARN del provider di contesto nella richiesta con l'ARN del provider di contesto specificato nella policy di attendibilità del ruolo.

Disponibilità: questa chiave è presente nella richiesta quando viene fornita un'asserzione di contesto nel parametro di `ProvidedContexts` richiesta mentre si assume un ruolo utilizzando l'operazione AWS STS AssumeRole API.

La condizione di esempio seguente verifica che l'ARN del provider di contesto passato nella richiesta corrisponda all'ARN specificato nella condizione della policy di attendibilità del ruolo. Ti consigliamo di aggiungere un controllo di valore null con `ForAllValues` per evitare che le chiavi di contesto mancanti o con valori vuoti vengano valutate come True. Per informazioni dettagliate, vedi Operatore di condizione per verificare la presenza di chiavi di condizione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Action": "sts:SetContext",
    "Effect": "Allow",
    "Resource": "*",
```

```
    "Condition": {
      "ForAllValues:ArnEquals": {
        "sts:RequestContextProviders": [
          "arn:aws:iam::aws:contextProvider/IdentityCenter"
        ]
      },
      "Null": {
        "sts:RequestContextProviders": "false"
      }
    }
  }
}
```

set: RoleAuthorizedByIdp

Lavora con operatori booleani.

Utilizza questa chiave per verificare che il provider di identità (IdP) abbia autorizzato esplicitamente il ruolo richiesto tramite l'`https://aws.amazon.com/roles`attestazione nel token OIDC. L'attestazione dei ruoli contiene gli ARN del ruolo IAM che il token può assumere. AWS STS supporta due formati per il valore del claim:

Array JSON (consigliato):

```
"https://aws.amazon.com/roles": [
  "arn:aws:iam::111122223333:role/RoleA",
  "arn:aws:iam::111122223333:role/RoleB"
]
```

Semicolon-delimited stringa:

```
"https://aws.amazon.com/roles": "arn:aws:iam::111122223333:role/
RoleA;arn:aws:iam::111122223333:role/RoleB"
```

Disponibilità: questa chiave è presente nelle richieste che utilizzano l'`AssumeRoleWithWebIdentity`operazione. Il valore è `true` quando il token include il `https://aws.amazon.com/roles` claim e il ruolo richiesto ARN corrisponde a un valore nel claim. Il valore è `false` quando il token non include il claim. Poiché la chiave è sempre presente nelle `AssumeRoleWithWebIdentity` richieste, utilizza l'operatore di `Bool` condizione anziché`BoolIfExists`.

Quando il token include il claim ma il ruolo richiesto ARN non corrisponde a un valore del claim, la `AssumeRoleWithWebIdentity` chiamata restituisce un `InvalidIdentityToken` errore.

È possibile utilizzare questa chiave condizionale nelle politiche di fiducia dei ruoli, nelle politiche di controllo delle risorse (RCP) e nelle politiche degli endpoint VPC per richiedere che un IdP autorizzi il ruolo prima che possa essere assunto.

La seguente politica di attendibilità dei ruoli è consentita `AssumeRoleWithWebIdentity` solo quando l'IdP ha autorizzato il ruolo richiesto nel token.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:oidc-provider/
idp.example.com"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": { "idp.example.com:aud": "my-app-id" },
                "Bool": { "sts:RoleAuthorizedByIdp": "true" }
            }
        }
    ]
}
```

set: RoleSessionName

Lavora con operatori stringa.

Utilizzare questa chiave per confrontare il nome di sessione specificato da un'entità principale quando si assume un ruolo con il valore specificato nella policy.

Disponibilità: questa chiave è presente nella richiesta quando il principale assume il ruolo utilizzando qualsiasi comando CLI Console di gestione AWS assume-role o qualsiasi operazione API. AWS STS `AssumeRole`

È possibile utilizzare questa chiave in una policy di attendibilità del ruolo per richiedere che gli utenti forniscano un nome di sessione specifico quando assumono un ruolo. Ad esempio, è

possibile richiedere che gli utenti IAM specifichino il proprio nome utente come nome di sessione. Dopo che l'utente IAM assume il ruolo, l'attività viene visualizzata nei log AWS CloudTrail con il nome della sessione corrispondente al nome utente. Ciò consente agli amministratori di distinguere tra le sessioni di ruolo quando un ruolo viene utilizzato da principali diversi.

La seguente policy di attendibilità del ruolo richiede che gli utenti IAM nell'account `111122223333` forniscano il nome utente IAM come nome di sessione quando assumono il ruolo. Questo requisito viene applicato utilizzando la variabile di condizione `aws:username` nella chiave di condizione. Questa policy consente agli utenti IAM di assumere il ruolo a cui è collegata la policy. Questa policy non consente a chiunque utilizzi credenziali temporanee di assumere il ruolo perché la variabile `username` è presente solo per gli utenti IAM.

> **Important**
>
> È possibile utilizzare qualsiasi chiave di condizione a valore singolo come variabile. Non è possibile utilizzare una chiave della condizione multi-valore come variabile.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "RoleTrustPolicyRequireUsernameForSessionName",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
            "Condition": {
                "StringLike": {"sts:RoleSessionName": "prefix-
${aws:username}"}
            }
        }
    ]
}
```

Quando un amministratore visualizza il AWS CloudTrail registro di un'azione, può confrontare il nome della sessione con i nomi utente nel proprio account. Nell'esempio seguente, l'utente denominato `matjac` ha eseguito l'operazione utilizzando il ruolo denominato `MateoRole`. L'amministratore può quindi contattare Mateo Jackson, che ha il nome dell'utente `matjac`.

```
    "assumedRoleUser": {
        "assumedRoleId": "AROACQRSTUVWRAOEXAMPLE:matjac",
        "arn": "arn:aws:sts::111122223333:assumed-role/MateoRole/matjac"
    }
```

Se si consente l'accesso tra account mediante i ruoli, gli utenti di un account possono assumere un ruolo in un altro account. L'ARN dell'utente del ruolo assunto elencato in CloudTrail include l'account in cui esiste il ruolo. Non include l'account dell'utente che ha assunto il ruolo. Gli utenti sono univoci solo all'interno di un account. Pertanto, si consiglia di utilizzare questo metodo per controllare CloudTrail i registri solo per i ruoli assunti dagli utenti negli account che amministri. Gli utenti potrebbero utilizzare lo stesso nome utente in più account.

set: SourceIdentity

Lavora con operatori stringa.

Utilizza questa chiave per confrontare l'identità di origine che un principale specifica quando si assume un ruolo con il valore specificato nella policy.

Disponibilità: questa chiave è presente nella richiesta quando il principale fornisce un'identità di origine assumendo un ruolo utilizzando qualsiasi comando CLI AWS STS assume-role o operazione API. AWS STS `AssumeRole`

È possibile utilizzare questa chiave in una policy di attendibilità del ruolo per richiedere che gli utenti forniscano un nome di sessione specifico quando assumono un ruolo. Ad esempio, è possibile richiedere alla forza lavoro o alle identità federate di specificare un valore per l'identità di origine. Puoi configurare il provider di identità (IdP) per utilizzare uno degli attributi associati agli utenti, ad esempio un nome utente o un messaggio di posta elettronica come identità di origine. L'IdP passa quindi l'identità di origine come attributo nelle asserzioni o nei claim a cui invia. AWS Il valore dell'attributo di identità di origine identifica l'utente o l'applicazione che assume il ruolo.

Dopo che l'utente assume il ruolo, l'attività viene visualizzata in Log di AWS CloudTrail con il valore dell'identità di origine impostato. In questo modo è più facile per gli amministratori determinare chi o cosa ha eseguito le azioni con un ruolo. AWS Per consentire a un'identità di impostare un'identità di origine, è necessario concedere le autorizzazioni per l'operazione `sts:SetSourceIdentity`.

A differenza di `sts:RoleSessionName`, dopo aver impostato l'identità di origine, il valore non può essere modificato. È presente nel contesto della richiesta di tutte le operazioni intraprese

con il ruolo dall'identità di origine. Il valore persiste nelle sessioni di ruolo successive quando si utilizzano le credenziali di sessione per assumere un altro ruolo. L'assunzione di un ruolo partendo da un altro si chiama concatenamento del ruolo.

È possibile utilizzare la chiave di condizione `aws:SourceIdentity` globale per controllare ulteriormente l'accesso alle AWS risorse in base al valore dell'identità di origine nelle richieste successive.

La seguente policy di attendibilità del ruolo consente all'utente IAM `AdminUser` di assumere un ruolo nell'account `111122223333`. Inoltre, concede l'autorizzazione all'`AdminUser` per impostare un'identità di origine, purché il set di identità di origine sia `DiegoRamirez`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAdminUserAssumeRole",
            "Effect": "Allow",
            "Principal": {"AWS": " arn:aws:iam::111122223333:user/AdminUser"},
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringEquals": {"sts:SourceIdentity": "DiegoRamirez"}
            }
        }
    ]
}
```

Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

set: TaskPolicyArn

Lavora con operatori ARN.

Usa questa chiave per confrontare l'ARN della policy in una AssumeRoot richiesta sts: con l'ARN della policy specificato nella policy.

Disponibilità: questa chiave è presente nella richiesta quando si effettua una richiesta utilizzando sts:. AssumeRoot

Gli amministratori possono utilizzare questa chiave di condizione nelle policy IAM per impedire a ruoli o utenti specifici all'interno dell'account di gestione o dell'account amministratore delegato di eseguire determinate operazioni quando assumono le credenziali root. Per ulteriori informazioni, consulta Esegui un'attività privilegiata su un AWS Organizations account membro.

sts: TransitiveTagKeys

Lavora con operatori stringa.

Utilizzare questa chiave per confrontare le chiavi dei tag di sessione transitivi nella richiesta con quelle specificate nella policy.

Disponibilità: questa chiave è presente nella richiesta quando si effettua una richiesta utilizzando credenziali di sicurezza temporanee. Queste includono le credenziali create utilizzando qualsiasi operazione di assume-role o l'operazione `GetFederationToken`.

Quando si effettua una richiesta utilizzando credenziali di sicurezza temporanee, il contesto della richiesta include la chiave di contesto `aws:PrincipalTag`. Questa chiave include un elenco di tag di sessione, tag di sessione transitivi e tag di ruolo. I tag di sessione transitivi sono tag che persistono in tutte le sessioni successive quando si utilizzano le credenziali di sessione per assumere un altro ruolo. L'assunzione di un ruolo partendo da un altro si chiama concatenamento del ruolo.

È possibile utilizzare questa chiave di condizione in una policy per richiedere l'impostazione di specifici tag di sessione come transitivi quando si assume un ruolo o si federa un utente.

Azioni, risorse e chiavi di condizione per AWS i servizi

Ogni AWS servizio può definire azioni, risorse e chiavi di contesto di condizione da utilizzare nelle policy IAM. Per un elenco dei AWS servizi e delle relative azioni, risorse e chiavi di contesto delle condizioni, consulta Actions, resources and condition keys nel Service Authorization Reference.

Risorse per ulteriori informazioni su IAM

IAM è un prodotto ricco e troverai molte risorse per aiutarti a saperne di più su come IAM può aiutarti a proteggere Account AWS le tue risorse.

Argomenti

- Identità
- Credenziali (password, chiavi di accesso e dispositivi MFA)
- Autorizzazioni e policy
- Federazione e delega
- IAM e altro AWS prodotti
- Best practice generali relative alla sicurezza
- Risorse generali

Identità

Consulta queste risorse per la creazione, la gestione e l'utilizzo di identità.

- Gestione delle identità nel Centro identità IAM: informazioni procedurali sulla creazione di utenti e gruppi nel Centro identità IAM.
- Identità IAM: una discussione approfondita su utenti, gruppi e ruoli.

Credenziali (password, chiavi di accesso e dispositivi MFA)

Consulta le seguenti guide per gestire password, chiavi di accesso e dispositivi MFA per te Account AWS e per gli utenti IAM.

- Password utente in AWS: descrive le opzioni per gestire le password per gli utenti IAM nel tuo account.
- Gestione delle chiavi di accesso per gli utenti IAM: descrive come funzionano le chiavi di accesso e come è possibile utilizzarle per effettuare chiamate programmatiche a AWS. Tuttavia, ci sono altre alternative più sicure delle chiavi di accesso che ti consigliamo di prendere in considerazione per prime. Per ulteriori informazioni, consulta Considerazioni e alternative per le chiavi di accesso a lungo termine nella guida Riferimenti generali di AWS .

- AWS Multi-factor autenticazione in IAM: illustra come configurare l'account e gli utenti IAM in modo da richiedere una password e un codice una tantum generato su un dispositivo prima che venga consentito l'accesso. (questo metodo viene talvolta denominato autenticazione a due fattori).

Per informazioni generali sui tipi di credenziali utilizzate per accedere ad Amazon Web Services, consulta Credenziali di sicurezza AWS nella guida Riferimenti generali di AWS .

Autorizzazioni e policy

Scopri i meccanismi interni delle policy IAM e trova i suggerimenti sui metodi migliori per assegnare le autorizzazioni:

- Politiche e autorizzazioni in AWS Identity and Access Management: introduce il linguaggio della policy utilizzata per definire le autorizzazioni. Illustra il modo in cui collegare le autorizzazioni a utenti o gruppi oppure, per alcuni prodotti AWS , alle risorse stesse.
- Documentazione di riferimento degli elementi delle policy JSON IAM: fornisce descrizioni ed esempi per ciascun elemento del linguaggio delle policy.
- Convalida delle policy IAM: trova le risorse per la convalida delle policy JSON.
- Esempi di policy basate su identità IAM— Mostra esempi di politiche per attività comuni in vari AWS prodotti.
- Generatore di policy AWS: crea policy personalizzate tramite la scelta di prodotti e operazioni da un elenco.
- IAM Policy Simulator : verifica se una policy consentirebbe o rifiuterebbe una richiesta specifica a. AWS

Federazione e delega

Puoi concedere l'accesso alle risorse presenti nel tuo account Account AWS agli utenti autenticati (che hanno effettuato l'accesso) altrove. Questi possono essere utenti IAM appartenenti a un altro provider Account AWS (noto come delega), utenti autenticati tramite il processo di accesso della tua organizzazione o utenti di un provider di identità Internet come Login with Amazon, Facebook, Google o qualsiasi altro provider di identità compatibile con OpenID Connect (OIDC). In questi casi, gli utenti ottengono credenziali di sicurezza temporanee per accedere alle risorse. AWS

- Tutorial IAM: Delega l'accesso attraverso AWS account che utilizzano ruoli IAM: descrive in dettaglio la procedura per concedere l'accesso multi-account a un utente IAM di un altro Account AWS.
- Scenari comuni per le credenziali temporanee— Descrive i modi in cui gli utenti possono essere federati AWS dopo essere stati autenticati all'esterno. AWS

IAM e altro AWS prodotti

La maggior parte dei AWS prodotti è integrata con IAM in modo da poter utilizzare le funzionalità di IAM per proteggere l'accesso alle risorse di tali prodotti. Le seguenti risorse illustrano l'IAM e la sicurezza di alcuni dei AWS prodotti più diffusi. Per un elenco completo dei prodotti che funzionano con IAM, inclusi i collegamenti per ulteriori informazioni su ciascuno, consulta AWS servizi supportati da IAM.

Uso di IAM con Amazon EC2

- Controllo dell'accesso alle risorse Amazon EC2: descrive come utilizzare le funzionalità IAM per consentire agli utenti di amministrare istanze, volumi e altri elementi di Amazon EC2.
- Usare profili dell'istanza— Descrive come utilizzare i ruoli IAM per fornire in modo sicuro le credenziali per le applicazioni che vengono eseguite su istanze Amazon EC2 e che richiedono l'accesso ad altri prodotti. AWS

Uso di IAM con Amazon S3

- Gestione delle autorizzazioni di accesso alle risorse Amazon S3: illustra il modello di sicurezza Amazon S3 per bucket e oggetti, che include le policy IAM.
- Scrittura di politiche IAM: concedere l'accesso alle User-Specific cartelle in un bucket Amazon S3: spiega come consentire agli utenti di proteggere le proprie cartelle in Amazon S3. Per ulteriori post su Amazon S3 e IAM, seleziona il tag S3 sotto il titolo del post del blog.

Utilizzo di IAM con Amazon RDS

- Utilizzo di AWS Identity and Access Management (IAM) per gestire l'accesso alle risorse di Amazon RDS: descrive come utilizzare IAM per controllare l'accesso alle istanze del database, agli snapshot del database e altro ancora.

- Introduzione alle Resource-Level autorizzazioni RDS: descrive come utilizzare IAM per controllare l'accesso a specifiche istanze di Amazon RDS.

Uso di IAM con Amazon DynamoDB

- Utilizzo di IAM per controllare l'accesso alle risorse DynamoDB: descrive come utilizzare IAM per consentire agli utenti di amministrare tabelle e indici DynamoDB.
- Il video seguente (8:55) spiega come fornire il controllo dell'accesso a singoli elementi o attributi (o entrambi) del database DynamoDB.

  Guida introduttiva al controllo degli accessi per Fine-Grained DynamoDB

Best practice generali relative alla sicurezza

Trova i consigli e le indicazioni degli esperti sui modi migliori per proteggere Account AWS le tue risorse:

- Best practice per la sicurezza, l'identità e la conformità : trova risorse su come gestire la sicurezza di tutti Account AWS i prodotti, inclusi suggerimenti per l'architettura di sicurezza, l'uso di IAM, la crittografia e la sicurezza dei dati e altro ancora.
- Gestione delle identità e degli accessi : il AWS Well-Architected Framework ti aiuta a comprendere i concetti chiave, i principi di progettazione e le best practice architetturali per la progettazione e l'esecuzione di carichi di lavoro nel cloud.
- Best practice per la sicurezza in IAM: fornisce suggerimenti sui modi di utilizzare IAM per proteggere l' Account AWS e le risorse.
- AWS CloudTrail Guida per l'utente : consente di AWS CloudTrail tenere traccia di una cronologia delle chiamate API effettuate AWS e di archiviare tali informazioni nei file di registro. Ciò consente di determinare quali utenti e account hanno effettuato l'accesso alle risorse nell'account, quando sono state effettuate le chiamate, quali operazioni sono state richieste e altro ancora.

Risorse generali

Esplora le seguenti risorse per saperne di più su IAM e AWS.

- Informazioni sul prodotto per IAM: informazioni generali su AWS Identity and Access Management .

- AWS re:Post per AWS Identity and Access Management — Visita il sito AWS re:Post per discutere con la AWS community di domande tecniche relative a IAM.

- Corsi e workshop: collegamenti a corsi specialistici e basati sui ruoli, oltre a laboratori di autoapprendimento per affinare le tue competenze e acquisire esperienza pratica. AWS
- AWS Developer Center: esplora i tutorial, scarica gli strumenti e scopri gli eventi per sviluppatori. AWS
- AWS Strumenti per sviluppatori: collegamenti a strumenti per sviluppatori, SDK, toolkit IDE e strumenti da riga di comando per lo sviluppo e la gestione di applicazioni. AWS
- Centro risorse introduttive: scopri come configurare la tua Account AWS, entrare a far parte della AWS community e lanciare la tua prima applicazione.
- Hands-On Tutorial: segui i tutorial dettagliati per avviare la tua prima applicazione su. AWS
- AWS Whitepaper: collegamenti a un elenco completo di AWS white paper tecnici, che trattano argomenti come architettura, sicurezza ed economia e scritti da Solutions Architects o altri esperti tecnici. AWS
- Supporto AWS Center: l'hub per la creazione e la gestione dei tuoi casi. Supporto AWS Include anche collegamenti ad altre risorse utili, come forum, domande frequenti tecniche, stato di salute del servizio e AWS Trusted Advisor.
- Supporto— La pagina web principale per informazioni su Supporto un canale di supporto individuale e a risposta rapida per aiutarti a creare ed eseguire applicazioni nel cloud.
- Contattaci: un punto di contatto centrale per richieste relative a fatturazione, account, eventi, uso illecito e altre questioni relative ad AWS .
- AWS Termini del sito: informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito dell'utente e su altri argomenti.

Chiamata all'API IAM utilizzando le richieste di query HTTP

Indice

- Endpoints
- HTTPS obbligatorio
- Firma delle richieste API IAM

È possibile accedere a IAM e ai AWS STS servizi in modo programmatico utilizzando l'API Query. Le richieste dell'API Query sono richieste HTTPS che devono contenere un parametro `Action` per indicare l'operazione da eseguire. IAM e AWS STS supporta le richieste GET e POST per tutte le azioni. Questo significa che l'API non richiede l'uso di GET per alcune operazioni e di POST per altre. Tuttavia, le richieste GET sono soggette ai limiti di dimensione di un URL. Anche può variare a seconda del browser, il limite tipico è di 2048 byte. Di conseguenza, per le richieste API Query che richiedono dimensioni maggiori, devi usare una richiesta POST.

La risposta è un documento XML. Per maggiori dettagli sulla risposta, consulta le pagine delle singole operazioni nella Documentazione di riferimento dell'API IAM o nella Documentazione di riferimento dell'API Servizio di token di sicurezza AWS.

> **Tip**
>
> Invece di effettuare chiamate dirette alle operazioni IAM o AWS STS API, puoi utilizzare uno degli AWS SDK. Gli AWS SDK sono costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione (Java, Ruby, .NET, iOS, Android, ecc.). Gli SDK forniscono un modo conveniente per creare un accesso programmatico a IAM e. AWS Ad esempio, gli SDK si occupano di attività quali la firma crittografica delle richieste (vedi di seguito), la gestione degli errori e la ripetizione automatica delle richieste. Per informazioni sugli AWS SDK, incluso come scaricarli e installarli, consulta la pagina Strumenti per Amazon Web Services.

Per ulteriori informazioni sulle operazioni delle API e sugli errori, consulta la Documentazione di riferimento dell'API IAM o la Documentazione di riferimento dell'API Servizio di token di sicurezza AWS.

Endpoints

IAM e AWS STS ognuno di essi ha un unico endpoint globale:

- (IAM) https://iam.amazonaws.com
- (AWS STS) https://sts.amazonaws.com

> **Important**
>
> AWS STS supporta anche l'invio di richieste agli endpoint regionali oltre all'endpoint globale. AWS consiglia di utilizzare endpoint regionali anziché endpoint globali per ridurre la latenza, aumentare la ridondanza e aumentare la validità del token di sessione. Prima di utilizzare AWS STS in una regione, devi attivare STS in quella regione per il tuo Account AWS. Per ulteriori informazioni sull'attivazione di regioni aggiuntive per, consulta. AWS STSManage (Gestione) AWS STS in un Regione AWS

Per ulteriori informazioni sugli AWS endpoint e le regioni per tutti i servizi, consulta Endpoint e quote di servizio nella. Riferimenti generali di AWS

HTTPS obbligatorio

L'API Query restituisce informazioni sensibili, come ad esempio le credenziali di sicurezza. Per tale ragione devi usare HTTPS per crittografare tutte le richieste API.

Firma delle richieste API IAM

Le richieste devono essere firmate usando un ID chiave di accesso e una Secret Access Key. Ti consigliamo vivamente di non utilizzare le tue Utente root dell'account AWS credenziali per il lavoro quotidiano con IAM. Puoi utilizzare le credenziali di un utente IAM oppure puoi utilizzarle per generare credenziali AWS STS di sicurezza temporanee.

Per firmare le tue richieste API, ti consigliamo di utilizzare la versione 4 di AWS Signature. Per informazioni sull'uso di Signature Version 4, consulta la pagina relativa al processo di firma con Signature Version 4 nelle AWS General Reference.

In Riferimento generale AWS sono disponibili anche le informazioni sull'utilizzo di Signature Version 2.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- AWS Credenziali di sicurezza. Fornisce informazioni generali sui tipi di credenziali utilizzate per l'accesso. AWS
- Best practice per la sicurezza in IAM. Presenta un elenco di suggerimenti per l'utilizzo del servizio IAM per proteggere AWS le risorse.
- Credenziali di sicurezza temporanee in IAM. Descrive come creare e usare credenziali di sicurezza temporanee.

Cronologia dei documenti per IAM

La tabella seguente descrive i principali aggiornamenti della documentazione IAM.

Modifica	Descrizione	Data
job_workflow_ref è accettato come claim di scoping alternativo per il provider OIDC di Actions GitHub	Per il provider GitHub Actions OIDC, IAM ora accetta il `sub` claim o il claim come controllo del provider di identità nelle politiche di fiducia dei `job_workflow_ref` ruoli.	12 maggio 2026
Chiavi di contesto delle condizioni IAM per API di credenziali specifiche del servizio	IAM ora supporta due nuove chiavi di contesto delle condizioni per controllare l'accesso alle API di credenziali specifiche del servizio: `iam:ServiceSpecificCredentialAgeDays` e `iam:ServiceSpecificCredentialServiceName` . Queste chiavi consentono di limitare la creazione e la gestione di credenziali specifiche del servizio in base alle impostazioni di scadenza e ai servizi consentiti. AWS • `iam:ServiceSpecificCredentialAgeDays` • `iam:ServiceSpecificCredentialServiceName`	4 settembre 2025

Chiavi di condizione degli endpoint VPC per i controlli perimetrali della rete	IAM ora supporta tre nuove chiavi di contesto condizionali AWS globali per l'implementazione di controlli perimetrali di rete scalabili:, e. `aws:VpceAccount` `aws:VpceOrgID` `aws:VpceOrgPaths` Queste chiavi aiutano a garantire che le richieste arrivino attraverso endpoint VPC di proprietà di specifici account, organizzazioni o unità organizzative, scalando automaticamente in base all'utilizzo degli endpoint VPC senza richiedere aggiornamenti delle policy quando si creano nuovi endpoint. Per ulteriori informazioni, consulta Establish permissions guardrails using data perimeters.	28 agosto 2025

Tutorial sulla federazione IAM SAML CloudFormation	IAM ha aggiunto nuovi tutorial per la creazione di provider di identità SAML () IdPs e ruoli federati utilizzando. CloudFormation • Creazione di un IdP SAML con CloudFormation • Crea un ruolo federato SAML con CloudFormation • Crea IdP SAML e ruolo federato con CloudFormation	25 agosto 2025
Controlli aggiuntivi dei provider OIDC condivisi per i provider OIDC condivisi	IAM ora include Amazon Cognito, Azure Sentinel, Pulumi Cloud e gli endpoint globali Vercel nell'elenco dei gestori dell'identità digitale OIDC condivisi che richiedono una valutazione esplicita di affermazioni specifiche in JSON Web Tokens (JWT).	1 agosto 2025

Sistema di analisi degli accessi IAM ha aggiunto gli analizzatori degli accessi interni	Sistema di analisi degli accessi IAM ti aiuta a identific are quali principali all'interno della tua organizzazione o del tuo account hanno accesso a risorse aziendali critiche selezionate. Gli analizzatori degli accessi interni supportan o l'implementazione del principio del privilegio minimo assicurando che le risorse specificate siano accessibi li solo ai principali designati all'interno dell'organizzazione o dell'account AWS .	16 giugno 2025
Controlli tramite provider di identità per i provider OIDC condivisi	IAM ora richiede una valutazio ne esplicita di affermazioni specifiche in JSON Web Tokens (JWT) per i gestori dell'identità digitale OIDC condivisi riconosciuti. Questo controllo di sicurezza garantisc e che solo le identità autorizza te dell'organizzazione prevista possano assumere ruoli e accedere alle risorse. AWS	6 giugno 2025
`IAMUserChangePassw ord` : autorizzazioni aggiunte	IAM ha aggiunto autorizza zioni a `IAMUserChangePassw ord` per autorizzare gli utenti specificati all'interno di un percorso.	28 maggio 2025

AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	Strumento di analisi degli accessi IAM ha aggiunto `iam:GetAccountAuthorizationDetails` alle autorizzazioni a livello di servizio di AccessAnalyzerServiceRolePolicy.	12 maggio 2025
AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto i punti di accesso ai bucket di directory di Amazon S3 alle autorizzazioni a livello di servizio di. AccessAnalyzerServiceRolePolicy	31 marzo 2025
Supporto per endpoint dual-stack con IAM	IAM ora fornisce un supporto per endpoint dual-stack migliorato che consente ai client di comunicare con IAM utilizzando indirizzi IPv4 o IPv6.	20 marzo 2025
Aggiornamenti alla logica di valutazione delle policy	Documentazione migliorata per la logica di valutazione delle politiche, inclusi diagrammi di flusso migliorati e spiegazioni più chiare su come AWS valuta le politiche per determinare se consentire o rifiutare una richiesta.	30 gennaio 2025
IAMDeleteRootUserCredentials : autorizzazioni rimosse	IAM ha rimosso l'autorizzazione `iam:DeleteVirtualMFADevice` dalla policy gestita.	7 gennaio 2025

AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	Sistema di analisi degli accessi IAM supporta l'autorizzazione per recuperare informazioni sulle impostazioni dell'account Amazon ECR e sulle policy di registro alle autorizzazioni a livello di servizio di AccessAnalyzerServiceRolePolicy.	10 dicembre 2024
Sistema di analisi degli accessi IAM ha aggiunto la configurazione degli accessi	Sistema di analisi degli accessi IAM ha aggiunto il supporto per configurare i sistemi di analisi per modificare l'ambito dei quali gli Account AWS, gli utenti IAM e i ruoli generano i risultati.	14 novembre 2024
Gestire centralmente l'accesso root per gli account membri	Ora puoi gestire le credenziali degli utenti root con privilegi tra gli account membri in AWS Organizations con l'accesso root centralizzato. Proteggi centralmente le credenziali dell'utente root del tuo account Account AWS gestito AWS Organizations per rimuovere e impedire il ripristino e l'accesso delle credenziali degli utenti root su larga scala. • Best practice per gli utenti root	14 novembre 2024

AWS aggiornamento gestito delle politiche - Nuove politiche	IAM ha aggiunto due nuove policy per limitare l'ambito delle autorizzazioni per le sessioni di utenti root con privilegi, che è possibile avviare dopo aver centraliz zato l'accesso degli utenti root per gli account membri dell'organizzazione. • IAMAuditRootUserCr edentials • IAMCreateRootUserP assword • IAMDeleteRootUserC redentials • S3UnlockBucketPolicy • SQSUnlockQueuePolicy	14 novembre 2024
Supporto per le politiche di controllo AWS Organizations delle risorse (RCP)	Utilizza una politica di controllo AWS Organizations delle risorse (RCP) per definire le autorizzazioni massime per le risorse all'interno degli account della tua organizzazione o unità organizzativa (OU). Le RCP limitano le autorizza zioni che le policy basate su identità e le policy basate sulle risorse possono concedere alle risorse degli account all'interno dell'organizzazione.	13 novembre 2024

AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto il supporto per l'autorizzazione per recuperare informazioni sui tag di ruoli e utenti IAM alle autorizzazioni a livello di servizio di AccessAnalyzerServiceRolePolicy.	29 ottobre 2024
Supporto migliorato per la crittografia SAML	È stata migliorata la documentazione sul supporto della crittografia SAML nei provider di identità IAM SAML, tra cui una migliore guida alla risoluzione dei problemi e chiarimenti sulla compatibilità dei servizi.	5 giugno 2024
AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto il supporto per l'autorizzazione per recuperare informazioni sulle policy di ruoli e utenti IAM alle autorizzazioni a livello di servizio di AccessAnalyzerServiceRolePolicy.	30 maggio 2024
Supporto della crittografia per i provider di identità SAML	Ora i provider IAM SAML supportano le asserzioni crittografate nella risposta SAML dal tuo IdP esterno. Per capire come funziona la crittografia con la federazione IAM SAML, consulta Usare la federazione per l'accesso alle SAML-based API.	4 febbraio 2024

AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto il supporto per l'autorizzazione per recuperare lo stato corrente dell'accesso pubblico al blocco per gli snapshot di Amazon EC2 alle autorizzazioni a livello di servizio di AccessAnalyzerServiceRolePolicy.	23 gennaio 2024
AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto stream e tabelle DynamoDB alle autorizzazioni a livello di servizio di. AccessAnalyzerServiceRolePolicy	11 gennaio 2024
AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto bucket di directory Amazon S3 alle autorizzazioni a livello di servizio di. AccessAnalyzerServiceRolePolicy	1 dicembre 2023

IAMAccessAnalyzerReadOnlyAccess : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto delle autorizzazioni per consentirti di verificare se IAMAccessAnalyzerReadOnlyAccess gli aggiornamenti alle tue policy concedono un accesso aggiuntivo. Questa autorizzazione è richiesta da Sistema di analisi degli accessi IAM per eseguire i controlli delle policy sulla policy.	26 novembre 2023
Analizzatori degli accessi inutilizzati aggiunti per Sistema di analisi degli accessi IAM	Sistema di analisi degli accessi IAM semplifica l'ispezione degli accessi inutilizzati per guidarti verso il privilegio minimo. Sistema di analisi degli accessi AWS IAM analizza continuamente i tuoi account per identificare gli accessi inutilizzati e crea una dashboard centralizzata con i risultati.	26 novembre 2023
Controlli delle policy personalizzati aggiunti per Sistema di analisi degli accessi IAM	Sistema di analisi degli accessi AWS IAM ora fornisce controlli delle policy personalizzati per verificare che le policy IAM aderiscano agli standard di sicurezza prima dell'implementazione.	26 novembre 2023

AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	Sistema di analisi degli accessi IAM ha aggiunto le operazioni IAM alle autorizzazioni a livello di servizio di AccessAnalyzerServiceRolePolicy per supportare le seguenti operazioni: • Elencare le entità per una policy • Generare dettagli sull'ultimo accesso al servizio • Elencare le informazioni sulla chiave di accesso	26 novembre 2023
Informazioni relative all'ultimo accesso a un'operazione e supporto alla generazione di policy per oltre 60 servizi e operazioni aggiuntivi	Ora IAM supporta le informazioni relative all'ultimo accesso a un'operazione e genera policy con informazioni a livello di operazione per oltre 60 servizi aggiuntivi, insieme a un elenco delle operazioni per cui sono disponibili le informazioni relative all'ultimo accesso.	1° novembre 2023
Supporto per le informazioni relative all'ultimo accesso a un'operazione per più di 140 servizi	Ora IAM fornisce le informazioni relative all'ultimo accesso a un'operazione per oltre 140 servizi, insieme a un elenco delle operazioni per cui sono disponibili le informazioni relative all'ultimo accesso.	14 settembre 2023

Supporto per più dispositivi con autenticazione a più fattori (MFA) per utenti root e utenti IAM	Ora puoi aggiungere fino a otto dispositivi MFA per utente, tra cui le chiavi di sicurezza FIDO, password monouso (TOTP) di software con applicazioni di autentica zione virtuale o token TOTP hardware.	16 novembre 2022
Supporto di Sistema di analisi degli accessi IAM per i nuovi tipi di risorse	Sistema di analisi degli accessi IAM ha aggiunto il supporto per i seguenti tipi di risorse: • Snapshot del volume Amazon EBS • Repository di Amazon ECR • File system di Amazon EFS • Snapshot del database Amazon RDS • Snapshot del cluster database Amazon RDS • Argomenti di Amazon SNS	25 ottobre 2022
Deprecazione e aggiornam ento U2F WebAuthn/FIDO	Sono state rimosse le menzioni di U2F come opzione MFA e sono state aggiunte informazioni sulle WebAuthn chiavi di sicurezza FIDO2 e FIDO.	31 maggio 2022

Aggiornamenti alla resilienza in IAM	Sono state aggiunte informazioni sul mantenimento dell'accesso alle credenziali IAM quando un evento interrompe la comunicazione tra Regioni AWS.	16 maggio 2022
Nuove chiavi della condizione globale per le risorse	Ora puoi controllare l'accesso alle risorse in base all'account, all'unità organizzativa (OU) o all'organizzazione in cui sono contenute le tue risorse. AWS Organizations In una policy IAM puoi utilizzare le chiavi della condizione globale `aws:ResourceAccount` , `aws:ResourceOrgID` e `aws:ResourceOrgPaths` .	27 aprile 2022
Esempi di codice per IAM che utilizzano gli AWS SDK	Sono stati aggiunti esempi di codice che mostrano come utilizzare IAM con un kit di sviluppo AWS software (SDK). Gli esempi sono suddivisi in estratti di codice che mostrano come richiamare le singole funzioni di servizio ed esempi che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.	7 aprile 2022

Aggiornamenti al diagramma di flusso della logica della valutazione delle policy	Aggiornamenti al diagramma di flusso della logica di valutazione della policy e al testo correlato nella sezione Determinare se una richiesta è consentita o rifiutata in un account.	17 novembre 2021
Aggiornamenti alle best practice di sicurezza	Sono state aggiunte informazioni sulla creazione di utenti amministrativi invece di utilizzare le credenziali dell'utente root, sono state rimosse le best practice di utilizzo dei gruppi IAM per assegnare le autorizzazioni agli utenti IAM ed è stato chiarito quando utilizzare le policy gestite anziché le policy in linea.	5 ottobre 2021
Aggiornamenti all'argomento della logica di valutazione delle policy per le policy basate sulle risorse	Sono state aggiunte informazioni sull'impatto delle policy basate sulle risorse e dei diversi tipi pincipali nello stesso account.	5 ottobre 2021
Aggiornamenti alle chiavi a valore singolo e alle chiavi di condizione multivalore	Ora le differenze tra le chiavi a valore singolo e di condizione multivalore sono illustrate in modo più dettagliato. Il tipo di valore è stato aggiunto a ogni chiave di contesto della condizione globale AWS.	30 settembre 2021

IAM Access Analyzer supporta i punti di accesso Amazon S3 Multi-Region	IAM Access Analyzer identifica i bucket Amazon S3 che consentono l'accesso pubblico e tra account, compresi quelli che utilizzano punti di accesso Amazon S3. Multi-Region	2 settembre 2021
AWS aggiornamenti gestiti delle policy: aggiornamento a una policy esistente	IAM Access Analyzer ha aggiornato una policy AWS gestita esistente.	2 settembre 2021
Più servizi supportati per la generazione di policy a livello di operazione	IAM Access Analyzer può generare policy IAM con informazioni sulle attività di accesso a livello di azione per servizi aggiuntivi. AWS	24 agosto 2021
Generazione di policy IAM per percorsi tra account	Ora puoi utilizzare IAM Access Analyzer per generare policy dettagliate in base alla tua attività di accesso utilizzando un AWS CloudTrail percorso in un altro account, ad esempio un percorso centralizzato. AWS Organizations	18 agosto 2021

Controlli delle policy aggiuntivi per Sistema di analisi degli accessi IAM	Sistema di analisi degli accessi IAM ha esteso la convalida delle policy aggiungendo nuovi controlli delle policy che convalidano le condizioni incluse nelle policy IAM. Questi controlli analizzano il blocco delle condizioni nell'istruzione della policy e riportano avvisi di sicurezza, errori e suggerimenti insieme a consigli attuabili. Sistema di analisi degli accessi IAM ha aggiunto i seguenti controlli delle policy: • Errore: formato principale del servizio non valido • Errore: chiave di tag mancante nella condizione • Avviso di sicurezza: negazione NotAction con tag non supportato (chiave di condizione per il servizio) • Avviso di sicurezza: Rifiuta con chiave di condizione tag non supportata per il servizio • Avviso di sicurezza: chiavi di condizione abbinate mancanti • Suggerimento: consentire l'utilizzo di una chiave di NotAction condizione del tag	29 giugno 2021

	non supportata per l'assistenza • Suggerimento: consentire con chiave di condizione e tag non supportata per il servizio	
Supporto per l'ultima operazione eseguita per più servizi	Ora è possibile visualizzare le informazioni dell'ultimo accesso dell'operazione alla console IAM relative all'ultima volta che un principale IAM ha utilizzato un'operazione per i seguenti servizi: Amazon EC2, IAM, Lambda e operazioni di gestione di Amazon S3. Puoi anche utilizzare l' AWS API AWS CLI or per recuperare un report sui dati. È possibile utilizzare queste informazioni per identificare le autorizzazioni non necessarie, in modo da perfezionare le policy IAM e aderire meglio al principio del privilegio minimo.	19 aprile 2021

Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti	Gli amministratori possono configurare i ruoli IAM per richiedere che le identità passino un'identità di origine che viene registrata in AWS CloudTrail. La revisione delle informazioni sull'iden tità di origine consente agli amministratori di determina re chi o cosa ha eseguito le operazioni con le sessioni del ruolo assunto.	13 aprile 2021
Generazione di policy IAM basate sull'attività di accesso	È ora possibile utilizzar e Sistema di analisi degli accessi AWS IAM per generare policy granulari in base all'attività di accesso rilevata in AWS CloudTrail.	7 Aprile 2021
Controlli delle policy per Sistema di analisi degli accessi IAM	Sistema di analisi degli accessi AWS IAM fornisce ora oltre 100 controlli delle policy con suggerimenti utili durante la creazione delle policy.	16 marzo 2021
Opzioni di convalida delle policy estese	La convalida estesa delle policy è disponibile nella console IAM e nell' AWS API e AWS CLI utilizza i controlli delle policy in IAM Access Analyzer per aiutarti a creare policy JSON sicure e funzional i.	15 marzo 2021

Tagging delle risorse IAM	È ora possibile taggare altre risorse IAM utilizzando una coppia di tag chiave-valore.	11 febbraio 2021
Policy delle password di default per gli utenti IAM	Se non imposti una policy personalizzata per le tue password Account AWS, le password degli utenti IAM devono ora soddisfar e la politica delle password predefinita AWS .	18 novembre 2020
Le pagine delle azioni, delle risorse e delle chiavi di condizione per AWS i servizi sono state spostate	Ogni AWS servizio può definire azioni, risorse e chiavi di contesto delle condizion i da utilizzare nelle policy IAM. È ora possibile trovare l'elenco dei AWS servizi e le relative azioni, risorse e chiavi di contesto delle condizion i nel Service Authorization Reference.	16 Novembre 2020
Durata della sessione dei ruoli più lunga per gli utenti IAM	Gli utenti IAM possono ora avere una durata più lunga della sessione di ruolo quando cambiano ruolo nel Console di gestione AWS, riducendo le interruzioni dovute alla scadenza della sessione. Agli utenti viene concessa la durata massima della sessione impostata per il ruolo o il tempo rimanente nella sessione dell'utente IAM, a seconda di quale sia minore.	24 luglio 2020

Utilizzo di Service Quotas per richiedere aumenti rapidi per le entità IAM	Puoi richiedere aumenti di quota per quote IAM regolabili utilizzando la console Service Quotas. Ora, alcuni aumenti vengono approvati automaticamente in Service Quotas e sono disponibili nel tuo account in pochi minuti. Vengono inviate richieste di dimensioni maggiori a. Supporto AWS	25 giugno 2020
Le ultime informazioni di accesso a IAM ora includono le operazioni di gestione di Amazon S3	Oltre alle informazioni sull'ultimo accesso al servizio, è ora possibile visualizzare nella console IAM le informazioni sull'ultima volta che un principale IAM ha utilizzato un'operazione Amazon S3. Puoi anche utilizzare l' AWS API AWS CLI or per recuperare il report sui dati. Il report include informazioni sui servizi e le azioni consentiti a cui i principali hanno tentato di accedere e quando. È possibile utilizzare queste informazioni per identificare le autorizzazioni non necessari e, in modo da perfezionare le policy IAM e aderire meglio al principio del privilegio minimo.	3 giugno 2020

Aggiunta del capitolo sulla sicurezza	Il capitolo sulla sicurezza ti aiuta a capire come configura re IAM e AWS STS raggiunge re i tuoi obiettivi di sicurezza e conformità. Scoprirai anche come utilizzare altri servizi AWS per monitorare e proteggere le risorse IAM.	29 aprile 2020
set: RoleSessionName	È ora possibile scrivere una policy che concede le autorizzazioni in base al nome di sessione specificato da un'entità principale quando si assume un ruolo.	21 aprile 2020
AWS aggiornamento della pagina di accesso	Quando accedi alla pagina di AWS accesso principale, ora puoi scegliere di accedere come utente IAM Utente root dell'account AWS o come utente IAM. In questo caso, l'etichetta sulla pagina indica se è necessario fornire il proprio indirizzo e-mail dell'utente root o le informazi oni sull'utente IAM. Questa documentazione include acquisizioni dello schermo aggiornate per comprendere meglio le pagine di accesso AWS .	4 marzo 2020

AWS: viaAWSService e aws: condition keys CalledVia	È ora possibile scrivere una policy per limitare se i servizi possono effettuare richieste per conto di un principale IAM (utente o ruolo). Quando un principale effettua una richiesta a un servizio AWS , tale servizio potrebbe utilizzare le credenziali del principale per effettuare richieste successiv e ad altri servizi. Utilizzar e la chiave di condizione `aws:ViaAWSService` per stabilire se un servizio effettua una richiesta utilizzando le credenziali di un principale. Utilizzare le chiavi di condizion e `aws:CalledVia` per stabilire se servizi specifici fanno una richiesta utilizzando le credenziali di un principale.	20 febbraio 2020
Policy Simulator aggiunge il supporto per i limiti delle autorizzazioni	È ora possibile verificar e l'effetto dei limiti delle autorizzazioni sulle entità IAM con il simulatore di policy IAM.	23 gennaio 2020

Cross-account valutazione delle politiche	Ora puoi scoprire come AWS valuta le politiche per l'accesso tra account. Ciò si verifica quando una risorsa in un account che concede fiducia include una policy basata sulle risorse che consente al principale in un altro account di accedere alla risorsa. La richiesta deve essere consentit a in entrambi gli account.	2 gennaio 2020
Tag di sessione	Puoi ora includere tag quando assumi un ruolo o esegui la federazione di un utente in AWS STS. Quando esegui l'operazione `AssumeRole` o `GetFederationToken` , puoi passare i tag di sessione come attributi. Quando esegui le `AssumeRoleWithWebI dentity` operazioni `AssumeRoleWithSAML` o, puoi trasferire gli attributi delle tue identità aziendali a. AWS	22 novembre 2019

Controlla l'accesso per gruppi di utenti Account AWSAWS Organizations	Ora puoi fare riferimento alle unità organizzative (OU) AWS Organizations nelle policy IAM. Se lo utilizzi AWS Organizations per organizzare i tuoi account in unità organizzative, puoi richiedere che i responsabili appartengano a una specifica unità organizzativa prima di concedere l'accesso alle tue risorse. I principali includono Utente root dell'account AWS utenti IAM e ruoli IAM. A tale scopo, specifica il percorso dell'unità organizzativa nella chiave di condizione `aws:PrincipalOrgPaths` nelle tue policy.	20 novembre 2019
Ultimo ruolo utilizzato	Puoi ora visualizzare la data, l'ora e la regione in cui è stato utilizzato l'ultimo ruolo. Queste informazioni consentono inoltre di identificare i ruoli inutilizzati nel tuo account. Puoi utilizzare l' Console di gestione AWS AWS API AWS CLI e per visualizzare le informazioni sull'ultima volta che un ruolo è stato utilizzato.	19 novembre 2019

Aggiornamento della pagina Chiavi di contesto delle condizioni globali	Puoi ora scoprire quando ciascuna delle chiavi di condizione globali è inclusa nel contesto di una richiesta . Puoi inoltre spostarti tra le chiavi più facilmente utilizzando il sommario della pagina. Le informazioni contenute nella pagina consentono di scrivere policy più accurate. Ad esempio, se i tuoi dipendenti utilizzano la federazione con ruoli IAM, devi utilizzare la chiave `aws:userId` e non la chiave `aws:userName` . La chiave `aws:userName` si applica solo agli utenti IAM e non ai ruoli.	6 ottobre 2019

ABAC in AWS	Scopri come funziona il controllo degli accessi basato sugli attributi (ABAC) nell' AWS uso dei tag e come si confronta con il modello di autorizzazione tradizionale. AWS Utilizza il tutorial ABAC per informazioni su come creare e testare una policy che consenta ai ruoli IAM con tag del principale di accedere alle risorse con i tag corrispon denti. Questa strategia consente alle persone di visualizzare o modificare solo le AWS risorse necessarie per il proprio lavoro.	3 ottobre 2019
AWS STS GetAccessKeyInfo operazione	Puoi esaminare le chiavi di AWS accesso nel tuo codice per determinare se provengono da un account di tua proprietà. Puoi passare l'ID di una chiave di accesso utilizzando il `aws sts get-access-key-info` AWS CLI comando o l'operazione `GetAccessKeyInfo` AWS API.	24 luglio 2019

Visualizzazione delle informazioni sull'ultimo accesso al AWS Organizations servizio in IAM	Ora puoi visualizzare le informazioni sull'ultimo accesso al servizio per un' AWS Organizations entità o una policy nella AWS Organizations sezione della console IAM. Puoi anche utilizzare l' AWS API AWS CLI or per recuperare il report sui dati. Questi dati includono informazioni sui servizi consentiti a cui i titolari di un AWS Organizations account hanno tentato l'ultima volta di accedere e quando. Puoi utilizzare queste informazioni per identificare le autorizzazioni non necessarie in modo da poter perfezionare AWS Organizations le tue politiche per aderire meglio al principio del privilegio minimo.	20 giugno 2019
Utilizzo di una policy gestita in una policy di sessione	Ora puoi trasferire fino a 10 ARN di policy gestite quando assumi un ruolo. Questo consente di limitare le autorizzazioni delle credenziali temporanee del ruolo.	7 maggio 2019

AWS STS Compatibilità regionale dei token di sessione per l'endpoint globale	Ora puoi scegliere se utilizzare i token degli endpoint globali versione 1 o versione 2. I token della versione 1 sono validi solo nelle AWS regioni disponibili per impostazione predefinita. Questi token non funzionano nelle regioni abilitate manualmente, ad esempio Asia Pacifico (Hong Kong). I token Versione 2 sono validi in tutte le regioni. Tuttavia, i token versione 2 sono più lunghi e potrebbe influenzare i sistemi utilizzati per archiviare temporaneamente i token.	26 aprile 2019
Consenti l'attivazione e la disabilitazione delle regioni AWS	Ora puoi creare una policy che consenta a un amministratore di abilitare e disabilitare la regione Asia Pacifico (Hong Kong) (ap-east-1).	24 aprile 2019

Pagina Le mie credenziali di sicurezza dell'utente IAM	Gli utenti IAM possono ora gestire le proprie credenziali nella pagina Le mie credenziali di sicurezza. Questa Console di gestione AWS pagina mostra informazioni sull'account come l'ID dell'account e l'ID utente canonico. Gli utenti possono anche visualizzare e modificare le proprie password, chiavi di accesso, X.509 certificati, chiavi SSH e credenziali Git.	24 gennaio 2019
API di Sistema di analisi degli accessi	È ora possibile utilizzare l' AWS API AWS CLI and per visualizzare le informazioni relative all'ultimo accesso al servizio.	7 dicembre 2018
Tagging di utenti e ruoli IAM	È ora possibile utilizzare i tag IAM per aggiungere attributi personalizzati a un'identità (utente o ruolo IAM) utilizzando una coppia chiave-valore di tag. È anche possibile usare i tag per controllare l'accesso di un'identità alle risorse o per controllare quali tag possono essere collegati a un'identità.	14 novembre 2018
Chiavi di sicurezza U2F	È ora possibile utilizzare chiavi di sicurezza U2F come opzione per l'autenticazione a più fattori (MFA) per l'accesso alla Console di gestione AWS.	25 settembre 2018

Supporto per gli endpoint Amazon VPC	Ora puoi stabilire una connessione privata tra il tuo VPC e la AWS STS regione Stati Uniti occidentali (Oregon).	31 luglio 2018
Limiti delle autorizzazioni	La nuova funzionalità permette di concedere più facilmente a dipendenti affidabili la possibili tà di gestire le autorizzazioni IAM senza concedere anche l'accesso amministrativo completo a IAM.	12 luglio 2018
aws: ID PrincipalOrg	La nuova chiave di condizione fornisce un modo più semplice per controllare l'accesso alle AWS risorse specificando l' AWS organizzazione dei responsabili IAM.	17 maggio 2018
leggi: RequestedRegion	La nuova chiave di condizione fornisce un modo più semplice per utilizzare le policy IAM per controllare l'accesso alle AWS regioni.	25 aprile 2018
Maggiore durata della sessione per i ruoli IAM	Ora un ruolo IAM può avere una sessione della durata di 12 ore.	28 marzo 2018
Flusso di lavoro aggiornato per la creazione di ruoli	Il nuovo flusso di lavoro migliora il processo di creazione di relazioni di trust e di collegamento delle autorizza zioni ai ruoli.	8 settembre 2017

Account AWS processo di accesso	L'esperienza di AWS accesso aggiornata consente sia all'utente root che agli utenti IAM di utilizzare il link Accedi alla console nella home page Console di gestione AWS dell'app.	25 agosto 2017
Policy IAM di esempio	La documentazione aggiornata include oltre 30 esempi di policy.	2 agosto 2017
Best practice di IAM	Le informazioni aggiunte alla sezione Utenti della console IAM semplificano l'adozione delle best practice di IAM.	5 luglio 2017
Dimensionamento automatico delle risorse	Resource-level le autorizzazioni possono controllare l'accesso e le autorizzazioni per le risorse Auto Scaling.	16 maggio 2017
Database Amazon RDS for MySQL e Amazon Aurora	Gli amministratori del database possono associare gli utenti del database agli utenti e ai ruoli IAM e quindi gestire l'accesso degli utenti a tutte le AWS risorse da un'unica posizione.	24 Aprile 2017
Service-linked ruoli	Service-linked i ruoli forniscono un modo più semplice e sicuro per delegare le autorizzazioni ai AWS servizi.	19 aprile 2017

Riepiloghi delle policy	I nuovi riepiloghi delle policy semplificano la comprensione delle autorizzazioni nelle policy IAM.	23 marzo 2017

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.