aws

Guida per gli sviluppatori

Amazon Cognito



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

Amazon Cognito: Guida per gli sviluppatori

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità 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 Amazon Cognito? 1
Di quali parti di Amazon Cognito hai bisogno? 1
Cosa puoi creare con Amazon Cognito 2
Bacini d'utenza 3
Pool di identità 4
Caratteristiche di Amazon Cognito 5
Bacini d'utenza 5
Pool di identità 8
Confronto tra pool di utenti e pool di identità di Amazon Cognito 10
Nozioni di base su Amazon Cognito 14
Disponibilità regionale 14
Prezzi di Amazon Cognito 15
Termini e concetti 15
Ambito generale 15
Bacini d'utenza 18
Pool di identità 23
Nozioni di base su AWS 24
Registrati per un Account AWS 24
Nozioni di base 25
Il tuo primo accesso 26
La tua prima app e il tuo pool di utenti 27
Altre opzioni di applicazione 30
Esempio di React SPA 31
Esempio di app mobile Flutter 34
Fasi successive 36
Aggiunta di un provider del social 37
Aggiungi un IdP SAML 45
Opzioni introduttive aggiuntive 48
Pianificazione del design 50
Scegli un approccio 50
Scenari comuni di Amazon Cognito 52
Autenticazione con un bacino d'utenza 53
Accedi alle risorse lato server 53
Accesso alle risorse con API Gateway e Lambda 54

Accedi ai AWS servizi con un pool di utenti e un pool di identità 55
Effettua l'autenticazione con una terza parte e accedi AWS servizi con un pool di identità 56
Accesso AWS AppSync risorse con Amazon Cognito 56
Machine-to-machine (M2M) 57
Step-up authentication 57
Configurazione della superficie di accesso 59
Login gestito 59
Managed login localization 61
Documenti relativi ai termini di utilizzo 63
Configurazione dell'accesso gestito con la console Amazon Cognito 64
Configurazione dell'accesso gestito con AWS Amplify 65
Visualizzazione della pagina di accesso 66
Personalizzazione delle pagine di autenticazione 67
Cose da sapere sull'accesso gestito e sull'interfaccia utente ospitata 68
Configurazione di un dominio 71
Branding e personalizzazione 87
Esegui la migrazione all'accesso gestito 100
Autenticazione degli utenti 102
Funzionalità 102
Sign-up 103
Sign-in 104
Accesso gestito 105
Protezione dalle minacce 106
Esperienza utente personalizzata 106
Monitoraggio e analisi 107
Integrazione dei pool di identità di Amazon Cognito 107
Piani di funzionalità del pool di utenti 107
Seleziona un piano di funzionalità 109
Funzionalità suddivise per piano 110
Caratteristiche del piano Essentials 114
Inoltre, le funzionalità del piano 118
Disattiva le funzionalità non idonee 121
Autenticazione 122
Client dell'app 124
Implementa flussi di autenticazione 134
Da sapere 137

Esempio di flusso di autenticazione 140
Autenticazione di accesso gestita 143
Autenticazione SDK 146
Flusso di autenticazione 153
Modelli di autorizzazione SDK 179
Third-party Accesso IdP 195
Accesso alle risorse 271
Accesso alle risorse con autorizzazioni verificate 272
Accesso alle risorse API Gateway 274
Accesso alle AWS risorse utilizzando un pool di identità 275
Token del pool di utenti 281
Token ID 283
Intestazione del token ID 283
Payload predefinito del token ID 284
Firma con token ID 291
Token di accesso 292
Intestazione del token di accesso 293
Payload predefinito del token di accesso 294
Machine-to-machine payload dei token di accesso 296
Firma del token di accesso 297
Analizza i token in modo sicuro 297
Aggiorna i token 298
Aggiornamento di token 299
Aggiorna la rotazione dei token 299
Aggiornamento dell'API e dei token SDK 302
Aggiornamento del token OAuth 303
Revoca dei token di aggiornamento 304
Revoca dei token 305
Abilitazione della revoca dei token 306
Revoca di un token 307
Verifica di un token JSON Web 307
Prerequisiti 308
Convalida dei token con aws-jwt-verify 309
Comprensione e ispezione dei token 310
Caching dei token 315
Caching di token di accesso computer-computer con Gateway Amazon API 316

M2M e ambiti ... 320
Autorizzazione dell'API ... 320
Machine-to-machine Autorizzazione (M2M) ... 321
Informazioni sugli ambiti ... 324
Informazioni sui server di risorse ... 326
Associazione di risorse ... 331
Gestione degli utenti ... 333
Gestione degli utenti ... 333
Consentire la registrazione dell’utente ... 334
Registrazione e conferma degli account utente ... 337
Creazione di utenti come amministratore ... 366
Aggiunta di gruppi a un bacino d'utenza ... 374
Gestione e ricerca degli utenti ... 377
Password ... 385
Importazione di utenti in un bacino d'utenza ... 391
Attributes ... 419
Aggiornamento di un pool di utenti e di un client di app ... 437
Impostazioni che non puoi modificare ... 438
Configurazione SMS ... 439
Aggiornamento di un pool di utenti con un AWS SDK, AWS CDK o API REST ... 440
Personalizzazione dei pool di utenti ... 443
Trigger Lambda ... 443
Da sapere ... 446
Imposta i trigger ... 449
Evento trigger Lambda per il bacino d'utenza ... 449
Parametri comuni del trigger Lambda del bacino d'utenza ... 450
Metadati del client ... 451
Attiva le fonti per operazione ... 454
Sorgenti di attivazione per funzione ... 462
Pre-iscrizione ... 467
Conferma successiva ... 474
Autenticazione preliminare ... 477
Autenticazione successiva ... 481
Federazione in entrata ... 485
Sfida personalizzata ... 495
Generazione pre-token ... 517

Migrazione utente ... 541
Messaggio personalizzato ... 548
Mittenti personalizzati ... 556
Messaggistica e analisi ... 572
Impostazioni e-mail ... 573
Impostazioni per i messaggi SMS ... 588
Utilizzo dell'analisi dei dati di Amazon Pinpoint ... 604
Protezione dei pool di utenti ... 611
Best practice di sicurezza ... 612
Proteggi il tuo pool di utenti a livello di rete ... 613
Proteggiti dall'abuso dei messaggi SMS ... 613
Comprendi l'autenticazione pubblica ... 613
Proteggi i clienti riservati con i segreti dei clienti ... 617
Proteggi altri segreti ... 618
Privilegio minimo di amministrazione del pool di utenti ... 619
Proteggi e verifica i token ... 622
Determina i provider di identità di cui ti vuoi fidare ... 622
Comprendi l'effetto degli ambiti sull'accesso ai profili utente ... 622
Disinfetta gli input per gli attributi utente ... 623
Aggiunta di MFA ... 623
Da sapere ... 624
Preferenze utente MFA ... 628
Logica MFA ... 631
Configurazione MFA ... 636
MFA per SMS ed e-mail ... 638
Autenticazione MFA con token di software TOTP ... 639
Livelli di autenticazione con attestazioni ACR e AMR ... 646
Livelli e metodi di autenticazione ... 647
Configurazione dei nomi dei livelli ACR ... 650
Requisiti del piano di funzionalità ... 651
Le tue responsabilità in materia di autenticazione avanzata ... 652
Utilizzo dei dispositivi ... 652
Configurazione dei dispositivi memorizzati ... 653
Recupero della chiave di un dispositivo ... 655
Accesso con un dispositivo ... 656
Visualizzazione, aggiornamento e annullamento della memorizzazione dei dispositivi ... 658

Protezione dalle minacce ... 659
Considerazioni e limitazioni per la protezione dalle minacce ... 662
Attivazione della protezione dalle minacce nei pool di utenti ... 663
Concetti di applicazione della protezione dalle minacce ... 663
Opzioni di protezione dalle minacce ... 664
Prerequisiti ... 665
Configurazione della protezione dalle minacce ... 665
Credenziali compromesse ... 672
Autenticazione adattiva ... 673
Raccolta dei dati ... 685
AWS WAF ACL Web ... 689
Cose da sapere AWS WAF ACL web e Amazon Cognito ... 689
Content-Security-Policy sulle risposte personalizzate ai blocchi ... 692
Codici di stato nelle risposte ai blocchi personalizzate ... 692
Associazione di una ACL Web al pool di utenti ... 693
Test e registrazione AWS WAF ACL web ... 695
Distinzione tra lettere maiuscole e minuscole ... 696
Protezione da eliminazione ... 698
Gestione delle informazioni utente ... 699
Operazioni di autenticazione e creazione degli utenti ... 700
Operazioni di ripristino della password ... 705
Operazioni di conferma ... 706
Multi-tenancy migliori pratiche ... 707
Per-tenant pool di utenti ... 709
Per-tenant client di app ... 711
Per-tenant gruppi di pool di utenti ... 713
Per-tenant attributi personalizzati ... 715
Per-tenant ambiti personalizzati ... 717
Risorsa di esempio ... 720
Multi-tenancy raccomandazioni sulla sicurezza ... 721
Integrazione con app ... 723
Autenticazione con AWS Amplify ... 724
Creazione di un'interfaccia utente con Amplify ... 725
Autenticazione con AWS SDK ... 726
Come funziona l'autenticazione ... 727
Autenticazione di accesso gestita ... 728

Autenticazione SDK ... 730
Third-party autenticazione tramite provider di identità ... 734
Autenticazione del pool di identità ... 737
Lavorare con gli AWS SDK ... 740
Autorizzazione con Amazon Verified Permissions ... 741
Autorizzazione API con autorizzazioni verificate ... 743
Esempio di policy per un utente Amazon Cognito. ... 746
Documentazione di riferimento degli endpoint del pool di utenti ... 749
endpoint di accesso gestiti ... 750
Endpoint Login ... 753
Endpoint Logout ... 755
Endpoint della federazione ... 759
Emittente OIDC ... 761
Endpoint Authorize ... 763
Endpoint Token ... 777
Endpoint UserInfo ... 789
Endpoint Revoke ... 792
Endpoint SAML idresponse ... 795
Concessioni OAuth 2.0 ... 797
Usare PKCE ... 798
In che modo Amazon Cognito utilizza PKCE ... 799
Risposte gestite agli errori di accesso e federazione ... 801
Implementazione e funzionamento ... 803
Quote ... 803
Informazioni sulle quote di frequenza di richiesta API ... 804
Gestione delle quote di frequenza di richiesta API ... 807
Quote di frequenza di richiesta di pool di utenti ... 816
Quote di frequenza di richiesta di pool di identità ... 830
Quote relative al numero e alla dimensione delle risorse ... 831
Multi-Region replica ... 839
Cose da sapere sulla replica in più regioni ... 840
Limitazioni della replica in più regioni ... 840
Configurazione della replica in più regioni ... 841
Operazioni API supportate nelle regioni secondarie ... 843
Failover in pool di utenti multiregionali ... 846
Applicazione di tag alle risorse ... 849

Risorse supportate ... 849
Limitazioni applicate ai tag ... 850
Gestione dei tag con la console ... 850
AWS CLI esempi ... 851
Operazioni dell'API ... 854
CloudFormation etichettatura ... 855
Risoluzione dei problemi ... 855
Errori di configurazione del dominio personalizzato ... 856
`Errore del token di aggiornamento non valido` ... 858
Errori di risposta SAML non validi nella federazione ... 859
Gli utenti con accesso gestito non possono selezionare un fattore MFA ... 862
Non è possibile ricevere il codice di reimpostazione della password tramite email/SMS ... 863
La reimpostazione della password non riesce con attributi di ripristino non `verificati: impossibile reimpostare la password per l'account, contatta l'assistenza` o riprova ... 864
`Errori SECRET_HASH` ... 865
La console Amazon Cognito sceglie una configurazione predefinita per un nuovo pool di utenti ... 865
Altre risorse per la risoluzione dei problemi ... 866
Esempi di codice ... 867
Amazon Cognito Identity ... 869
Nozioni di base ... 870
Scenari ... 895
Gestore dell'identità per Amazon Cognito ... 897
Nozioni di base ... 899
Scenari ... 1066
Amazon Cognito Sync ... 1254
Nozioni di base ... 1255
Scambio di identità in cambio di credenziali AWS ... 1258
Pool di identità di Amazon Cognito ... 1258
Configurazione dei pool di identità ... 1260
Flusso di autenticazione dei pool di identità ... 1268
Ruoli IAM ... 1279
Best practice di sicurezza ... 1298
Utilizzo di attributi per il controllo degli accessi ... 1303
Utilizzo del controllo degli accessi basato su ruoli ... 1311

Ottenere le credenziali 1318
Uso delle credenziali di 1324
Third-party fornitori di identità 1327
Developer-authenticated identità 1360
Cambio delle identità 1377
Nozioni di base sui pool di identità 1380
Creazione di un pool di identità in Amazon Cognito 1381
Configurazione di un SDK 1383
Integrazione dei provider di identità 1383
Ottenere le credenziali 1384
Applicazione di esempio 1384
Sicurezza 1419
Protezione dei dati 1420
Crittografia dei dati 1421
Crittografia dei dati in transito 1430
Gestione dell’identità e degli accessi 1431
Destinatari 1432
Autenticazione con identità 1432
Gestione dell’accesso tramite policy 1433
Funzionamento di Amazon Cognito con IAM 1435
Identity-based esempi di politiche 1443
Risoluzione dei problemi 1448
Uso di ruoli collegati ai servizi 1451
Registrazione di log e monitoraggio 1455
Monitoraggio dei costi 1456
Esportazione dei log dei pool di utenti 1459
Monitoraggio delle quote e dell'utilizzo 1470
CloudTrail registri 1492
AWS PrivateLink 1520
Flussi di autenticazione per AWS PrivateLink integrazione 1521
Modalità operative per AWS PrivateLink 1522
Considerazioni 1523
Controllo dell'accesso con politiche di controllo delle risorse 1528
Creazione di un endpoint di interfaccia 1529
Creazione di una policy dell'endpoint 1530
Crea una politica basata sull'identità 1532

Convalida della conformità 1533
Resilienza 1534
Considerazioni sui dati regionali 1535
Sicurezza dell'infrastruttura 1535
Analisi della configurazione e delle vulnerabilità 1536
AWS politiche gestite 1536
Aggiornamenti delle policy 1538
Legacy 1542
Amazon Cognito Sync 1542
Modifica della disponibilità di Amazon Cognito Sync 1543
Nozioni di base su Amazon Cognito Sync 1544
Sincronizzazione dei dati tra client 1546
Gestione dei callback degli eventi 1556
Implementazione della sincronizzazione push 1573
Implementazione dei flussi di Amazon Cognito Sync 1583
Personalizzazione dei flussi di lavoro con Amazon Cognito Events 1586
Utilizzo di Amazon Cognito Sync con pool di identità 1591
Gestione di set di dati 1591
Pubblicazione in blocco di dati 1592
Attivazione della sincronizzazione push 1592
Configurazione di Amazon Cognito Streams 1593
Configurazione di Amazon Cognito Events 1593
Branding dell'interfaccia utente ospitata (classica) 1594
Specificare un logo personalizzato nel marchio classico 1594
Specificare le personalizzazioni CSS nel branding classico 1595
Personalizzazione dell'interfaccia utente ospitata con il marchio classico in Console di gestione AWS 1599
Personalizzazione dell'interfaccia utente ospitata con il marchio classico nell'API dei pool di utenti e con AWS CLI 1600
Cronologia dei documenti 1602
..... mdcxxix

Che cos'è Amazon Cognito?

Amazon Cognito è una piattaforma di identità per app web e per dispositivi mobili. È una directory utente, un server di autenticazione e un servizio di autorizzazione per i token di accesso OAuth 2.0 e le credenziali AWS . Con Amazon Cognito, puoi autenticare e autorizzare gli utenti dalla directory utente integrata, dalla directory aziendale e dai provider di identità utente come Google e Facebook.

Argomenti

- Di quali parti di Amazon Cognito hai bisogno?
- Cosa puoi creare con Amazon Cognito
- Bacini d'utenza
- Pool di identità
- Caratteristiche di Amazon Cognito
- Confronto tra pool di utenti e pool di identità di Amazon Cognito
- Nozioni di base su Amazon Cognito
- Disponibilità regionale
- Prezzi di Amazon Cognito
- Termini e concetti comuni di Amazon Cognito
- Nozioni di base su AWS

I due componenti che seguono costituiscono Amazon Cognito. Funzionano in maniera indipendente o in tandem, in base alle esigenze di accesso degli utenti.

Di quali parti di Amazon Cognito hai bisogno?

Amazon Cognito risponde a due domande diverse. Scegli in base a ciò che deve fare la tua applicazione.

Chi è questo utente? (autenticazione)

> Usa un pool di utenti. Un pool di utenti è una directory di utenti che attinge allo standard OpenID Connect (OIDC) per autenticare gli utenti ed emettere token web JSON (JWT) alla tua app o API. Gli utenti possono accedere direttamente, tramite provider social come Google e Apple o

tramite provider SAML e OIDC aziendali. La maggior parte delle applicazioni che richiedono la registrazione e l'accesso iniziano qui.

Quali AWS risorse può raggiungere questo utente? (autorizzazione)

Usa un pool di identità. Un pool di identità scambia un token affidabile (da un pool di utenti, un social provider o un provider SAML o OIDC) con AWS credenziali temporanee con privilegi limitati in modo che l'app possa chiamare servizi come Amazon S3 e Amazon DynamoDB per conto dell'utente. Un pool di identità può anche emettere credenziali con privilegi limitati agli utenti ospiti che non hanno effettuato l'accesso, senza alcun token.

Ti servono entrambi?

Usali insieme quando la tua app effettua l'accesso agli utenti e deve concedere loro l'accesso diretto alle AWS risorse: esegui l'autenticazione con un pool di utenti, quindi scambia il token del pool di utenti con un pool di identità in cambio di AWS credenziali.

Nessuno dei due componenti richiede l'altro. Un pool di utenti può emettere token al proprio backend senza pool di identità. Un pool di identità può mediare le credenziali di un provider che già possiedi, anche senza un pool di utenti. Per ulteriori informazioni sulle differenze tra i due componenti, vedereConfronto tra pool di utenti e pool di identità di Amazon Cognito. Per iniziare a costruire, consultaNozioni di base su Amazon Cognito.

Cosa puoi creare con Amazon Cognito

La maggior parte delle applicazioni Amazon Cognito presenta alcune forme comuni. Usa questo elenco per trovare quella più vicina a ciò che stai creando, quindi segui il link per la procedura dettagliata completa.

Fai accedere gli utenti alla tua app

Un pool di utenti autentica i tuoi utenti ed emette token web JSON (JWT) che l'app o l'API verificano. Questo è il punto di partenza per la maggior parte delle app che richiedono la registrazione e l'accesso. Consulta Autenticazione con un bacino d'utenza.

Proteggi le tue API o le tue risorse di back-end

Usa il token di accesso di un pool di utenti per autorizzare le richieste alle tue risorse lato server oppure consenti ad Amazon API Gateway di convalidare il token e accedere a una funzione Lambda o alla tua API. Consulta Accedi alle risorse di back-end con i token del pool di utenti e Accesso alle risorse con API Gateway e Lambda tramite un bacino d'utenza.

Offri agli utenti un accesso temporaneo ai servizi AWS

Scambia un token del pool di utenti con un pool di identità con AWS credenziali temporanee con privilegi limitati in modo che la tua app possa chiamare servizi come Amazon S3 e Amazon DynamoDB per conto dell'utente. Consulta Accesso AWS servizi con un pool di utenti e un pool di identità.

Consenti agli utenti di accedere con Google, Apple o un elenco aziendale

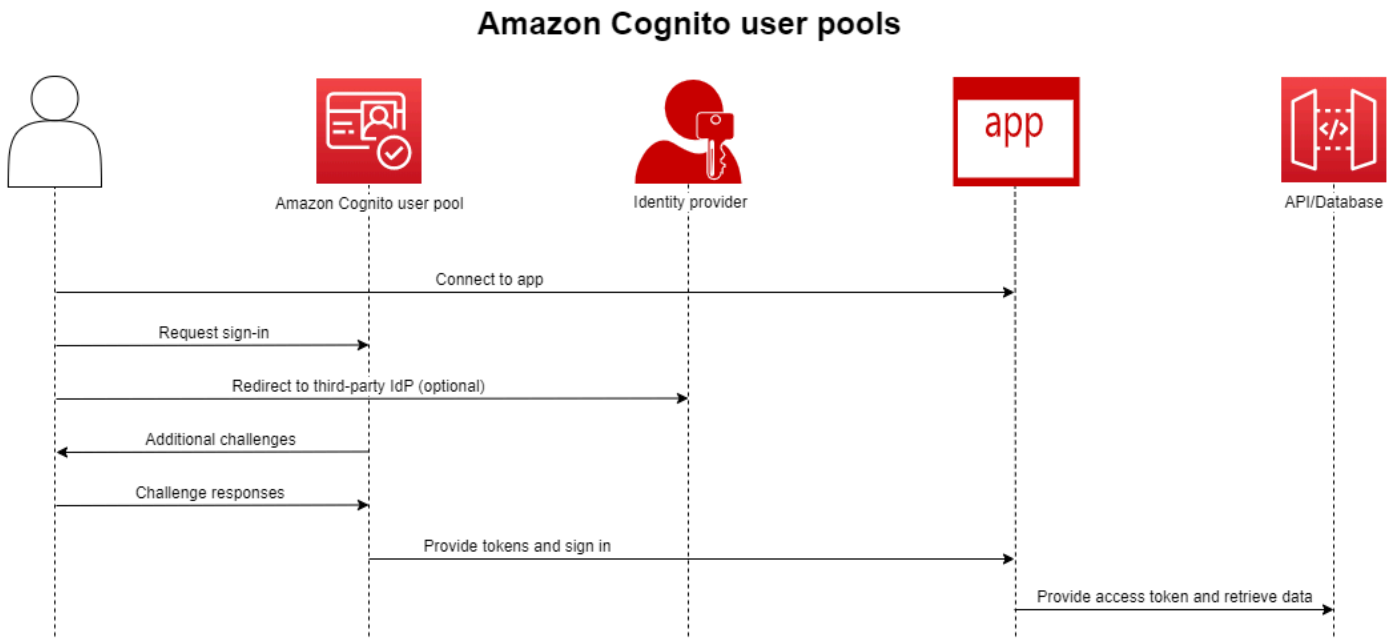
Federati tramite un social o un SAML/OIDC provider. Un pool di utenti gestisce l'accesso e rilascia all'app un set standard di token, oppure un pool di identità scambia il token esterno direttamente in cambio di credenziali. AWS Consulta Effettua l'autenticazione con una terza parte e accedi AWS servizi con un pool di identità.

Autorizza un'applicazione o un dispositivo senza alcun utente umano

Machine-to-machine L'autorizzazione (M2M) rilascia un token di accesso a un'app, servizio o dispositivo che si autentica come se stesso con la `client_credentials` concessione, senza alcun utente umano. Consulta Autorizza le applicazioni da macchina a macchina (M2M).

Per il set completo di scenari con diagrammi di architettura, vedere. Scenari comuni di Amazon Cognito

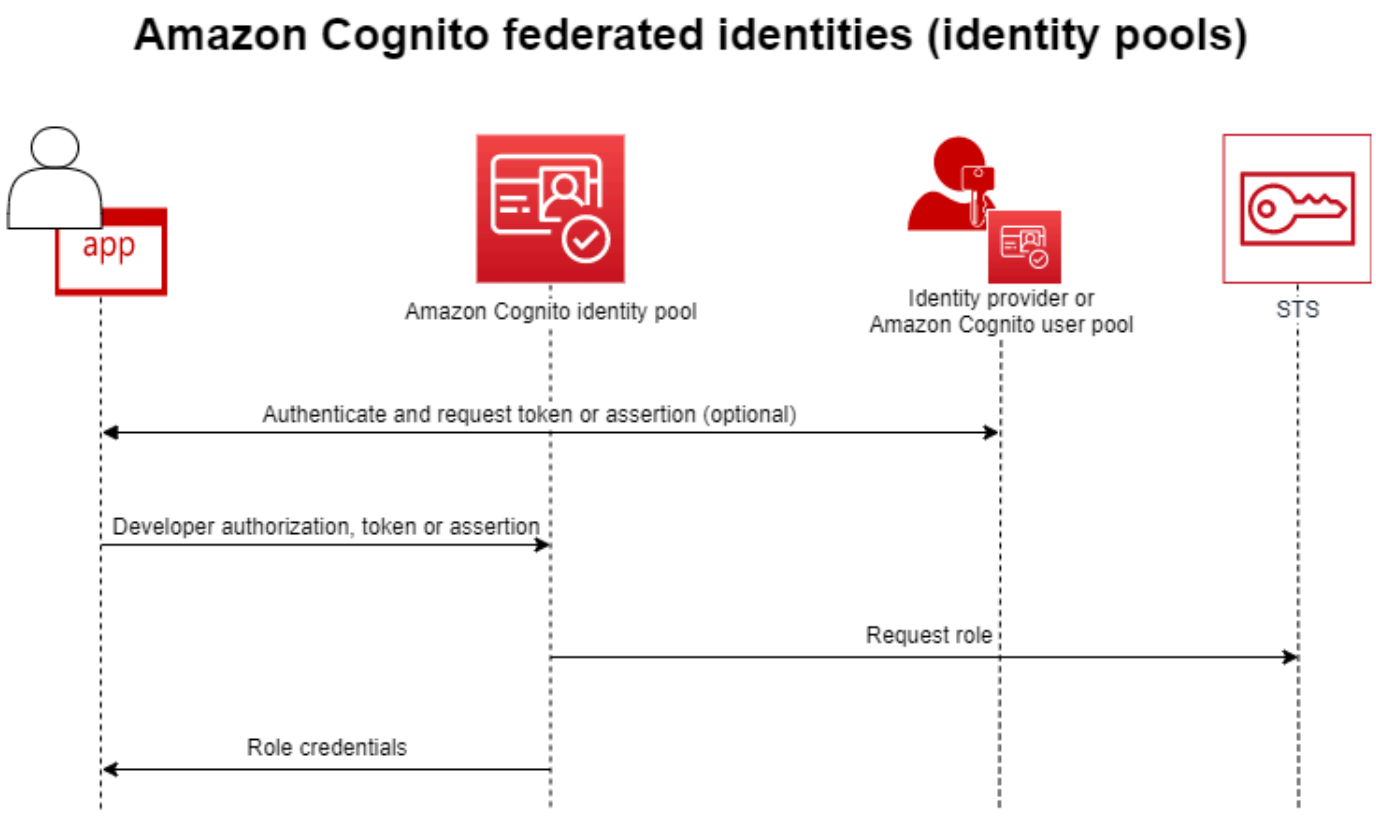
Bacini d'utenza



Crea un pool di utenti quando desideri autenticare e autorizzare gli utenti per l'app o l'API. I pool di utenti sono una directory di utenti con creazione, gestione e autenticazione degli utenti sia in modalità self-service che gestita dagli amministratori. Il pool di utenti può essere una directory indipendente e gestore dell'identità digitale OIDC e un provider di servizi intermedio (SP) per provider di terze parti di identità della forza lavoro e dei clienti. Puoi fornire il Single Sign-On (SSO) nella tua app per le identità della forza lavoro della tua organizzazione in SAML 2.0 e OIDC con pool di utenti. IdPs Puoi anche fornire l'SSO nella tua app per le identità dei clienti dell'organizzazione negli archivi di identità OAuth 2.0 Amazon, Google, Apple e Facebook. Per ulteriori informazioni sulla gestione dell'identità e degli accessi dei clienti (CIAM) consulta Cos'è CIAM?.

I pool di utenti non richiedono l'integrazione con un pool di identità. Da un pool di utenti, puoi emettere token web JSON (JWT) autenticati direttamente in un'app, un server web o un'API.

Pool di identità



Configura un pool di identità Amazon Cognito quando desideri autorizzare utenti autenticati o anonimi ad accedere alle tue risorse. AWS Un pool di identità rilascia AWS le credenziali della tua app per fornire risorse agli utenti. Puoi autenticare gli utenti con un provider di identità attendibile, come un

pool di utenti o un servizio SAML 2.0. Facoltativamente, puoi anche emettere credenziali per gli utenti guest. I pool di identità utilizzano il controllo degli accessi basato sui ruoli e sugli attributi per gestire l'autorizzazione degli utenti ad accedere alle risorse. AWS

I pool di utenti non richiedono l'integrazione con un pool di identità. Un pool di identità può accettare richieste autenticate direttamente dai provider di identità della forza lavoro e degli utenti.

Un pool di utenti e un pool di identità di Amazon Cognito utilizzati insieme

Nel diagramma all'inizio di questo argomento, Amazon Cognito viene utilizzato per autenticare l'utente e quindi concedere l'accesso a un Servizio AWS.

1. L'utente dell'app accede tramite un pool di utenti e riceve token OAuth 2.0.
2. La tua app scambia un token del pool di utenti con un pool di identità per AWS credenziali temporanee che puoi utilizzare con le API e (). AWS AWS Command Line Interface AWS CLI
3. L'app assegna la sessione delle credenziali all'utente e fornisce l'accesso autorizzato ad Servizi AWS Amazon S3 e Amazon DynamoDB.

Per altri esempi che utilizzano pool di identità e pool di utenti, consulta. Scenari comuni di Amazon Cognito

In Amazon Cognito, l'obbligo di sicurezza del cloud del modello di responsabilità condivisa è conforme a SOC 1-3, PCI DSS, ISO 27001 ed è idoneo. HIPAA-BAA Puoi progettare la tua sicurezza nel cloud in Amazon Cognito in modo che sia conforme a SOC 1-3, ISO 27001 e, ma non PCI DSS. HIPAA-BAA Per ulteriori informazioni, consulta Servizi AWS coperti. Consulta anche Considerazioni sui dati regionali.

Caratteristiche di Amazon Cognito

Bacini d'utenza

Un pool di utenti Amazon Cognito è una directory di utenti. Con un pool di utenti, gli utenti possono accedere all'app web o per dispositivi mobili tramite Amazon Cognito o eseguire la federazione tramite un IdP di terze parti. Gli utenti federati e locali hanno un profilo utente nel pool di utenti.

Gli utenti locali sono quelli che hanno effettuato al registrazione o che sono stati creati direttamente nel pool di utenti. Puoi gestire e personalizzare questi profili utente in Console di gestione AWS, un SDK o (). AWS AWS Command Line Interface AWS CLI

I pool di utenti di Amazon Cognito accettano token e asserzioni di terze parti IdPs e raccolgono gli attributi utente in un JWT che rilasciano alla tua app. Puoi standardizzare la tua app su un set di JWT mentre Amazon Cognito gestisce le interazioni con IdPs, mappando le relative affermazioni su un formato di token centrale.

Un pool di utenti Amazon Cognito può essere un IdP autonomo. Amazon Cognito attinge dallo standard OpenID Connect (OIDC) per generare JWT per l'autenticazione e l'autorizzazione. Quando accedi agli utenti locali, il pool di utenti è autorevole per tali utenti. Quando esegui l'autenticazione degli utenti locali, hai accesso alle seguenti funzionalità.

- Implementa il tuo front-end web che chiama l'API dei pool di utenti Amazon Cognito per autenticare, autorizzare e gestire gli utenti.
- Configura l'autenticazione a più fattori (MFA) per gli utenti. Amazon Cognito supporta password monouso e l'autenticazione MFA con messaggio SMS.
- Proteggi dall'accesso da account utente controllati da malintenzionati.
- Crea i tuoi flussi di autenticazione a più fasi personalizzati.
- Cerca gli utenti in un'altra directory ed esegui la migrazione ad Amazon Cognito.

Un pool di utenti di Amazon Cognito può anche svolgere un duplice ruolo di fornitore di servizi (SP) per la tua IdPs app e di IdP per la tua app. I pool di utenti di Amazon Cognito possono connettersi a consumatori IdPs come Facebook e Google o a forza lavoro IdPs come Okta e Active Directory Federation Services (ADFS).

Con i token OAuth 2.0 e OpenID Connect (OIDC) emessi da un pool di utenti Amazon Cognito, puoi

- Accettare un ID token nell'app che autentica un utente e fornisce le informazioni necessarie per configurare il profilo dell'utente
- Accettare un token di accesso nell'API con gli ambiti OIDC che autorizzano le chiamate API degli utenti.
- Recupera AWS le credenziali da un pool di identità di Amazon Cognito.

Funzionalità del pool di utenti Amazon Cognito

Funzionalità	Description
Provider di identità OIDC	Emetti token ID per autenticare gli utenti

Server di autorizzazione	Emetti token di accesso per autorizzare l'accesso degli utenti alle API
Fornitore di servizi SAML 2.0	Trasforma le asserzioni SAML in ID e token di accesso
Parte che fa affidamento sull'OIDC	Trasforma i token OIDC in token ID e di accesso
Fattore affidatario del fornitore di servizi sociali	Trasforma i token ID di Apple, Facebook, Amazon o Google nel tuo ID e accedi ai token
Servizio frontend di autenticazione	Registra, gestisci e autentica gli utenti con accesso gestito
Supporto API per la tua interfaccia utente	Crea, gestisci e autentica gli utenti tramite richieste API di autenticazione negli SDK supportati[1] AWS
Multi-factor authentication	Usa messaggi SMS, TOTP o il dispositivo dell'utente come fattore di autenticazione aggiunto[1]
Monitoraggio e risposta della sicurezza	Proteggiti da attività dannose e password non sicure[1]
Personalizza i flussi di autenticazione	Crea il tuo meccanismo di autenticazione o aggiungi passaggi personalizzati ai flussi esistenti[2]
Gruppi	Crea raggruppamenti logici di utenti e una gerarchia di richieste di ruolo IAM quando passi i token ai pool di identità
Personalizza i token	Personalizza il tuo ID e i token di accesso con reclami nuovi, modificati e soppressi
Personalizza gli attributi utente	Assegna valori agli attributi utente e aggiungi i tuoi attributi personalizzati

[1] La funzionalità non è disponibile per gli utenti federati.

[2] La funzionalità non è disponibile per gli utenti con accesso federato e gestito.

Per ulteriori informazioni sui bacini d'utenza, consulta Crea una nuova applicazione nella console Amazon Cognito e Documentazione di riferimento delle API dei bacini d'utenza di Amazon Cognito.

Pool di identità

Un pool di identità è una raccolta di identificatori univoci, o identità, che assegni ai tuoi utenti o ospiti e autorizzi a ricevere credenziali temporanee. AWS Quando presenti una prova di autenticazione a un pool di identità sotto forma di richieste attendibili da un gestore dell'identità digitale social SAML 2.0, OpenID Connect (OIDC) o OAuth 2.0, associ l'utente a un'identità nel pool di identità. Il token creato dal pool di identità per l'identità può recuperare le credenziali di sessione temporanee da (). Servizio di token di sicurezza AWS AWS STS

A complemento delle identità autenticate, puoi anche configurare un pool di identità per autorizzare l'accesso AWS senza autenticazione IdP. Puoi offrire una prova di autenticazione personalizzata con. Developer-authenticated identità Puoi anche concedere AWS credenziali temporanee agli utenti ospiti, con identità non autenticate.

Con i pool di identità, hai due modi per integrarti con le policy IAM del tuo. Account AWS Puoi utilizzare queste due funzionalità insieme o singolarmente.

Role-based controllo degli accessi

Quando un utente passa le richieste al pool di identità, Amazon Cognito sceglie il ruolo IAM richiesto. Per personalizzare le autorizzazioni del ruolo in base alle esigenze, le policy IAM vengono applicate a ciascun ruolo. Ad esempio, se un utente dimostra di lavorare nel reparto marketing, riceve le credenziali per un ruolo con policy personalizzate in base alle esigenze di accesso del reparto marketing. Amazon Cognito può richiedere un ruolo predefinito, un ruolo basato su regole che eseguono query delle richieste dell'utente o un ruolo basato sull'appartenenza al gruppo dell'utente in un pool di utenti. Puoi anche configurare la policy di attendibilità del ruolo in modo che IAM consideri attendibile solo il tuo pool di identità per generare sessioni temporanee.

Attributi per il controllo degli accessi

Il pool di identità legge gli attributi dalle richieste dell'utente e li associa ai tag principali nella sessione temporanea dell'utente. Puoi quindi configurare le policy basate sulle risorse IAM per consentire

o negare l'accesso alle risorse in base ai principali IAM che contengono i tag di sessione del pool di identità. Ad esempio, se l'utente dimostra di lavorare nel reparto marketing, contrassegna AWS STS la sessione`Department: marketing`. Il tuo bucket Amazon S3 consente operazioni di lettura basate su una PrincipalTag condizione aws: che richiede un valore di `marketing` per il tag. `Department`

Funzionalità del pool di identità di Amazon Cognito

Funzionalità	Description
Party che si affida al pool di utenti di Amazon Cognito	Scambia un token ID dal tuo pool di utenti con credenziali di identità web provenienti da AWS STS
Provider di servizi SAML 2.0	Scambia asserzioni SAML con credenziali di identità web da AWS STS
Parte affidataria OIDC	Scambia token OIDC con credenziali di identità web da AWS STS
Fattore affidatario del fornitore di servizi sociali	Scambia i token OAuth di Amazon, Facebook, Google, Apple e Twitter con credenziali di identità web da AWS STS
Relying party personalizzato	Con AWS le credenziali, puoi scambiare le attestazioni in qualsiasi formato con credenziali di identità web provenienti da AWS STS
Accesso non autenticato	Emetti credenziali di identità web ad accesso limitato senza autenticazione AWS STS
Role-based controllo degli accessi	Scegli un ruolo IAM per il tuo utente autenticato in base alle sue affermazioni e configura i ruoli in modo che vengano assunti solo nel contesto del tuo pool di identità
Attribute-based controllo degli accessi	Converti le attestazioni in tag principali per la tua sessione AWS STS temporanea e utilizza le policy IAM per filtrare l'accesso alle risorse in base ai tag principali

Per ulteriori informazioni sui pool di identità, consulta Guida introduttiva ai pool di identità di Amazon Cognito e Documentazione di riferimento delle API dei pool di identità di Amazon Cognito.

Confronto tra pool di utenti e pool di identità di Amazon Cognito

Funzionalità	Description	Bacini d'utenza	Pool di identità
Provider di identità OIDC	Emetti token ID OIDC per autenticare gli utenti dell'app	✓	
Elenco utenti	Memorizza i profili utente per l'autenti cazione	✓	
Autorizza l'accesso all'API	Emetti token di accesso per autorizza re l'accesso degli utenti alle API (incluse le operazioni API self-service dei profili utente), ai database e ad altre risorse che accettano gli ambiti OAuth	✓	
Autorizzazione dell'identità web IAM	Genera token con AWS STS cui puoi scambiare credenziali temporanee AWS		✓
Provider di servizi SAML 2.0 e provider di identità OIDC	Emetti token OIDC personalizzati in base alle dichiarazioni di un provider di identità SAML 2.0	✓	

Relying party OIDC e provider di identità OIDC	Emetti token OIDC personalizzati in base alle dichiarazioni di un provider di identità OIDC	✓		
Relying party OAuth 2.0 e provider di identità OIDC	Emetti token OIDC personalizzati in base agli ambiti dei provider di social network OAuth 2.0 come Apple e Google	✓		
Provider di servizi SAML 2.0 e broker di credenziali	Emetti AWS credenziali temporanee basate sulle richieste di un provider di identità SAML 2.0			✓
Relying party e broker di credenziali OIDC	Emetti AWS credenziali temporanee in base alle richieste di un provider di identità OIDC			✓
Fornitore di servizi sociali che si affidano a terzi e broker di credenziali	Emetti AWS credenziali temporanee basate su token web JSON provenienti da applicazioni per sviluppatori con provider di social network come Apple e Google			✓

Broker di credenziali e relying party per pool di utenti Amazon Cognito	Emetti AWS credenziali temporanee basate su token web JSON dai pool di utenti di Amazon Cognito		✓
Broker personalizzato per relying party e credenziali	Emetti AWS credenziali temporanee a identità arbitrarie, autorizzate dalle credenziali IAM degli sviluppatori		✓
Servizio frontend di autenticazione	Registra, gestisci e autentica gli utenti con accesso gestito	✓	
Supporto API per la tua interfaccia utente di autenticazione	Crea, gestisci e autentica gli utenti tramite richieste API negli SDK supportati[1] AWS	✓	
MFA	Usa messaggi SMS, TOTP o il dispositivo dell'utente come fattore di autenticazione aggiunto[1]	✓	
Monitoraggio e risposta della sicurezza	Proteggiti da attività dannose e password non sicure[1]	✓	
Personalizza i flussi di autenticazione	Crea il tuo meccanismo di autenticazione o aggiungi passaggi personalizzati ai flussi esistenti[1]	✓	

Gruppi di utenti	Crea raggruppamenti logici di utenti e una gerarchia di richieste di ruolo IAM quando passi i token ai pool di identità	✓	
Personalizza i token	Personalizza il tuo ID e i token di accesso con rivendicazioni e ambiti nuovi, modificati e soppressi	✓	
AWS WAF ACL web	Monitora e controlla le richieste al tuo front-end di autenticazione con AWS WAF	✓	
Personalizza gli attributi utente	Assegna valori agli attributi utente e aggiungi i tuoi attributi personalizzati	✓	
Accesso non autenticato	Emetti credenziali di identità web ad accesso limitato senza autenticazione AWS STS		✓
Role-based controllo degli accessi	Scegli un ruolo IAM per il tuo utente autenticato in base alle sue affermazioni e configura la fiducia dei ruoli per limitare l'accesso agli utenti con identità web		✓

Attribute-based controllo degli accessi	Trasforma le affermazioni degli utenti in tag principali per la sessione AWS STS temporanea e utilizza le policy IAM per filtrare l'accesso alle risorse in base ai tag principali	✓

[1] La funzionalità non è disponibile per gli utenti federati.

Nozioni di base su Amazon Cognito

Ad esempio, applicazioni con pool di utenti, vedereCrea una nuova applicazione nella console Amazon Cognito.

Per un'introduzione ai pool di identità, vedereGuida introduttiva ai pool di identità di Amazon Cognito.

Per i collegamenti alle esperienze di configurazione guidate con pool di utenti e pool di identità, consultaOpzioni di configurazione guidate per Amazon Cognito.

Per iniziare a usare un AWS SDK, consulta AWS Developer Tools. Per le risorse per sviluppatori specifiche per Amazon Cognito, consulta le risorse per sviluppatori di Amazon Cognito.

Per utilizzare Amazon Cognito devi disporre di un Account AWS. Per ulteriori informazioni, consulta Nozioni di base su AWS.

Disponibilità regionale

Amazon Cognito è disponibile in diverse AWS regioni in tutto il mondo. In ciascuna regione, Amazon Cognito viene distribuito su più zone di disponibilità. Queste zone di disponibilità sono fisicamente isolate l'una dall'altra, ma sono unite da connessioni di rete private a bassa latenza, a velocità effettiva elevata e altamente ridondanti. Queste zone di disponibilità consentono AWS di fornire servizi, tra cui Amazon Cognito, con livelli molto elevati di disponibilità e ridondanza, riducendo al contempo al minimo la latenza.

Per vedere se Amazon Cognito è attualmente disponibile in qualsiasi regione Regione AWS, consulta AWS Servizi per regione.

Per ulteriori informazioni sugli endpoint dei servizi API regionali, consulta AWS Regioni ed endpoint nel. Riferimenti generali di Amazon Web Services

Per ulteriori informazioni sul numero di zone di disponibilità presenti in ciascuna regione, consulta Infrastruttura globale AWS.

Prezzi di Amazon Cognito

Per informazioni sui prezzi di Amazon Cognito, consulta Prezzi di Amazon Cognito.

Termini e concetti comuni di Amazon Cognito

Amazon Cognito fornisce credenziali per app Web e mobili. Attinge e si basa su termini comuni nella gestione delle identità e degli accessi. Sono disponibili molte guide all'identità universale e ai termini di accesso. Alcuni esempi sono:

- Terminologia nel Body of Knowledge di IDPro
- AWS Servizi di identità
- Glossario del NIST CSRC

Gli elenchi seguenti descrivono termini che sono esclusivi di Amazon Cognito o che hanno un contesto specifico in Amazon Cognito.

Argomenti

- Ambito generale
- Bacini d'utenza
- Pool di identità

Ambito generale

I termini in questo elenco non sono specifici di Amazon Cognito e sono ampiamente riconosciuti tra i professionisti della gestione delle identità e degli accessi. Quello che segue non è un elenco esaustivo di termini, ma una guida al contesto specifico di Amazon Cognito contenuto in questa guida.

Token di accesso

Un token web JSON (JWT) che contiene informazioni sull'autorizzazione di un'entità ad accedere ai sistemi informativi.

App, applicazione

In genere, un'applicazione mobile. In questa guida, app è spesso un'abbreviazione di un'applicazione Web o di un'app mobile che si connette ad Amazon Cognito.

Attribute-based controllo degli accessi (ABAC)

Un modello in cui un'app determina l'accesso alle risorse in base alle proprietà di un utente, come la qualifica o il reparto. Gli strumenti di Amazon Cognito per applicare l'ABAC includono i token ID nei pool di utenti e i tag principali nei pool di identità.

Autenticazione

Il processo di creazione di un'identità autentica ai fini dell'accesso a un sistema informativo. Amazon Cognito accetta prove di autenticazione da provider di identità di terze parti e funge anche da fornitore di autenticazione per le applicazioni software.

Autorizzazione

Il processo di concessione delle autorizzazioni a una risorsa. I token di accesso al pool di utenti contengono informazioni che le applicazioni possono utilizzare per consentire a utenti e sistemi di accedere alle risorse.

Server di autorizzazione

Un sistema OAuth o OpenID Connect (OIDC) che genera token web JSON. Il server di autorizzazione gestito dei pool di utenti Amazon Cognito è il componente del server di autorizzazione dei due metodi di autenticazione e autorizzazione nei pool di utenti. I pool di utenti supportano anche i flussi di risposta alle sfide delle API nell'autenticazione SDK.

App riservata, app lato server

Un'applicazione a cui gli utenti si connettono in remoto, con codice su un server di applicazioni e accesso ai segreti. Si tratta in genere di un'applicazione Web.

Identity provider (IdP) (Provider di identità)

Un servizio che archivia e verifica le identità degli utenti. Amazon Cognito può richiedere l'autenticazione a provider esterni ed essere un IdP delle app.

Token web JSON (JWT)

Un JSON-formatted documento che contiene affermazioni relative a un utente autenticato. I token ID autenticano gli utenti, i token di accesso autorizzano gli utenti e i token di aggiornamento aggiornano le credenziali. Amazon Cognito riceve token da fornitori esterni ed emette token per app o. AWS STS

Machine-to-machine Autorizzazione (M2M)

Il processo di autorizzazione delle richieste agli endpoint API per entità di macchine non interattive con l'utente, come un livello di applicazione del server web. I pool di utenti forniscono l'autorizzazione M2M nelle concessioni di credenziali del client, mentre gli ambiti OAuth 2.0 nei token di accesso.

Multi-factor autenticazione (MFA)

Il requisito che gli utenti forniscano un'autenticazione aggiuntiva dopo aver fornito nome utente e password. I pool di utenti di Amazon Cognito dispongono di funzionalità MFA per gli utenti locali.

Provider OAuth 2.0 (social)

Un IdP verso un pool di utenti o un pool di identità che fornisce token di accesso e aggiornamento JWT. I pool di utenti di Amazon Cognito automatizzano le interazioni con i social provider dopo l'autenticazione degli utenti.

Provider OpenID Connect (OIDC)

Un IdP a un pool di utenti o a un pool di identità che estende la specifica OAuth per fornire token ID. I pool di utenti di Amazon Cognito automatizzano le interazioni con i provider OIDC dopo l'autenticazione degli utenti.

Chiave di accesso, WebAuthn

Una forma di autenticazione in cui le chiavi crittografiche, o passkey, sul dispositivo di un utente forniscono la prova dell'autenticazione. Gli utenti verificano la presenza di meccanismi biometrici o di codice PIN in un autenticatore hardware o software. Le passkey sono resistenti al phishing e legate a caratteristiche specifiche e offrono un'esperienza sicura senza password. websites/apps I pool di utenti di Amazon Cognito supportano l'accesso con password.

Senza password

Una forma di autenticazione in cui un utente non deve inserire una password. I metodi di accesso senza password includono password monouso (OTP) inviate a indirizzi e-mail e numeri di telefono e passkey. I pool di utenti di Amazon Cognito supportano l'accesso con OTP e passkey.

App pubblica

Un'applicazione autonoma su un dispositivo, con codice archiviato localmente e senza accesso ai segreti. Si tratta in genere di un'app per dispositivi mobili.

Server di risorse

Un'API con controllo degli accessi. I pool di utenti di Amazon Cognito utilizzano anche il server di risorse per descrivere il componente che definisce la configurazione per l'interazione con un'API.

Role-based controllo degli accessi (RBAC)

Un modello che concede l'accesso in base alla designazione funzionale dell'utente. I pool di identità di Amazon Cognito implementano RBAC con differenziazione tra i ruoli IAM.

Fornitore di servizi (SP), relying party (RP)

Un'applicazione che si affida a un IdP per affermare l'affidabilità degli utenti. Amazon Cognito funge da SP per gli SP esterni IdPs e come IdP per gli SP basati su app.

Provider SAML

Un IdP verso un pool di utenti o un pool di identità che genera documenti di asserzione con firma digitale che l'utente trasmette ad Amazon Cognito.

Identificatore univoco universale (UUID)

Un'etichetta a 128 bit applicata a un oggetto. Gli UUID di Amazon Cognito sono unici per pool di utenti o pool di identità, ma non sono conformi a un formato UUID specifico, incluso l'UUID RFC. Non dovresti convalidare rigorosamente il formato degli identificatori di Amazon Cognito.

Elenco utenti

Una raccolta di utenti e relativi attributi che fornisce tali informazioni ad altri sistemi. I pool di utenti di Amazon Cognito sono elenchi di utenti e anche strumenti per il consolidamento degli utenti provenienti da elenchi utenti esterni.

Bacini d'utenza

Quando vedi i termini nell'elenco seguente di questa guida, si riferiscono a una funzionalità o configurazione specifica dei pool di utenti.

Autenticazione adattiva

Una funzionalità di sicurezza avanzata che rileva potenziali attività dannose e applica una protezione aggiuntiva ai profili utente.

Client dell'app

Un componente che definisce le impostazioni per un pool di utenti come IdP per un'app.

URL di callback, URI di reindirizzamento, URL di ritorno

Un'impostazione in un client di app e un parametro nelle richieste al server di autorizzazione del pool di utenti. L'URL di callback è la destinazione iniziale per gli utenti autenticati nell'app.

Choice-based authentication

Una forma di autenticazione API con pool di utenti in cui ogni utente ha a disposizione una serie di scelte per l'accesso. Le loro scelte potrebbero includere nome utente e password con o senza MFA, accesso tramite passkey o accesso senza password con password monouso tramite e-mail o SMS. L'applicazione può modellare il processo di scelta degli utenti richiedendo un elenco di opzioni di autenticazione o dichiarando un'opzione preferita.

Confronta con l'autenticazione basata su client.

Client-based authentication

Una forma di autenticazione con l'API dei pool di utenti e i backend delle applicazioni creati con gli AWS SDK. Nell'autenticazione dichiarativa, l'applicazione determina in modo indipendente il tipo di accesso che un utente deve eseguire e richiede tale tipo in anticipo.

Confronta con l'autenticazione basata sulla scelta.

Credenziali compromesse

Una funzionalità di sicurezza avanzata che rileva le password degli utenti che gli aggressori potrebbero conoscere e applica una protezione aggiuntiva ai profili utente.

Conferma

Processo che determina che sono stati soddisfatti i prerequisiti per consentire a un nuovo utente di accedere. La conferma viene in genere effettuata tramite la verifica dell'indirizzo e-mail o del numero di telefono.

Autenticazione personalizzata

Un'estensione dei processi di autenticazione con trigger Lambda che definiscono sfide e risposte aggiuntive per gli utenti.

Autenticazione del dispositivo

Un processo di autenticazione che sostituisce la MFA con l'accesso che utilizza l'ID di un dispositivo affidabile.

Dominio, dominio del pool di utenti

Un dominio web che ospita le tue pagine di accesso gestite in AWS. Puoi configurare il DNS in un dominio di tua proprietà o utilizzare un prefisso identificativo di sottodominio in un dominio di tua proprietà. AWS

Piano Essentials

Il piano di funzionalità con gli ultimi sviluppi nei pool di utenti. Il piano Essentials non include le funzionalità di sicurezza di apprendimento automatico del piano Plus.

Provider esterno, fornitore terzo

Un IdP che ha una relazione di fiducia con un pool di utenti. I pool di utenti fungono da entità intermedia tra i provider esterni e l'applicazione, gestendo i processi di autenticazione con SAML 2.0, OIDC e i provider social. I pool di utenti consolidano i risultati dell'autenticazione dei provider esterni in un unico IdP in modo che le applicazioni possano elaborare più utenti con un'unica libreria relying-party OIDC.

Piano delle funzionalità

Il gruppo di funzionalità che è possibile selezionare per un pool di utenti. I piani di funzionalità prevedono costi diversi nella AWS fattura. Per impostazione predefinita, i nuovi pool di utenti utilizzano il piano Essentials.

Piani attuali

- Piano Lite
- Piano Essentials
- Piano Plus

Utente federato, utente esterno

Un utente in un pool di utenti che è stato autenticato da un provider esterno.

Interfaccia utente ospitata (classica), pagine dell'interfaccia utente ospitate

La versione iniziale dei servizi di autenticazione front-end, relying party e identity provider sul dominio del pool di utenti. L'interfaccia utente ospitata ha un set di funzionalità di base e un

aspetto semplificato. Puoi applicare il branding dell'interfaccia utente ospitata caricando un file di immagine del logo e un file con un set predeterminato di stili CSS. Confronta con l'accesso gestito.

Trigger Lambda

Una funzione in AWS Lambda cui un pool di utenti può richiamare automaticamente i punti chiave dei processi di autenticazione degli utenti. Puoi utilizzare i trigger Lambda per personalizzare i risultati dell'autenticazione.

Utente locale

Un profilo utente nella directory degli utenti del pool di utenti che non è stato creato mediante l'autenticazione con un provider esterno.

Utente collegato

Un utente di un provider esterno la cui identità viene unita a quella di un utente locale.

Piano Lite

Il piano di funzionalità con le funzionalità originariamente lanciate con i pool di utenti. Il piano Lite non include le nuove funzionalità del piano Essentials o le funzionalità di sicurezza ad apprendimento automatico del piano Plus.

Server di autorizzazione gestito, server di autorizzazione dell'interfaccia utente ospitato, server di autorizzazione

Un componente dell'accesso gestito che ospita servizi di interazione IdPs e app sul dominio del pool di utenti. L'interfaccia utente ospitata si differenzia dall'accesso gestito per le funzionalità interattive con l'utente che offre, ma ha le stesse funzionalità del server di autorizzazione.

Accesso gestito, pagine di accesso gestite

Un insieme di pagine Web sul dominio del pool di utenti che ospitano servizi per l'autenticazione degli utenti. Questi servizi includono funzioni per operare come IdP, relying party per terze parti IdPs e server di un'interfaccia utente di autenticazione interattiva per l'utente. Quando configuri un dominio per il tuo pool di utenti, Amazon Cognito mette online tutte le pagine di accesso gestite.

La tua applicazione importa librerie OIDC che richiamano i browser degli utenti e li indirizzano all'interfaccia utente di accesso gestita per la registrazione, l'accesso, la gestione delle password e altre operazioni di autenticazione. Dopo l'autenticazione, le librerie OIDC possono elaborare l'esito della richiesta di autenticazione.

Autenticazione di accesso gestita

Sign-in con i servizi del dominio del pool di utenti, tramite pagine del browser interattive dall'utente o richieste API HTTPS. Le applicazioni gestiscono l'autenticazione di accesso gestita con le librerie OpenID Connect (OIDC). Questo processo include l'accesso con provider esterni, l'accesso per utenti locali con pagine di accesso gestite interattive e l'autorizzazione M2M. Anche l'autenticazione con la classica interfaccia utente ospitata rientra in questo termine.

Rispetto all'autenticazione AWS SDK.

Piano Plus

Il piano di funzionalità con gli ultimi sviluppi e le funzionalità di sicurezza avanzate nei pool di utenti.

Autenticazione SDK, autenticazione AWS SDK

Una serie di operazioni API di autenticazione e autorizzazione che puoi aggiungere al backend dell'applicazione con un AWS SDK. Questo modello di autenticazione richiede un meccanismo di accesso personalizzato. L'API può accedere agli utenti locali e agli utenti collegati.

Rispetto all'autenticazione di accesso gestita.

Protezione dalle minacce, funzionalità di sicurezza avanzate

Nei pool di utenti, la protezione dalle minacce si riferisce a tecnologie progettate per mitigare le minacce ai meccanismi di autenticazione e autorizzazione. L'autenticazione adattiva, il rilevamento delle credenziali compromesse e gli elenchi di indirizzi IP bloccati rientrano nella categoria della protezione dalle minacce.

Personalizzazione dei token

Il risultato di un trigger Lambda precedente alla generazione del token che modifica l'ID o il token di accesso di un utente in fase di esecuzione.

Pool di utenti, provider di identità Amazon Cognito**`cognito-idp`**, pool di utenti Amazon Cognito

Una AWS risorsa con servizi di autenticazione e autorizzazione per applicazioni che funzionano con OIDC. IdPs

Verifica

Processo di conferma che un utente possiede un indirizzo e-mail o un numero di telefono. Un pool di utenti invia un codice a un utente che ha inserito un nuovo indirizzo e-mail o numero di

telefono. Quando inviano il codice ad Amazon Cognito, verificano la proprietà della destinazione del messaggio e possono ricevere messaggi aggiuntivi dal pool di utenti. Inoltre, vedi conferma.

Profilo utente, account utente

Una voce per un utente nella rubrica degli utenti. Tutti gli utenti, compresi quelli di terze parti IdPs, hanno un profilo nel loro pool di utenti.

Pool di identità

Quando vedi i termini nell'elenco seguente di questa guida, si riferiscono a una funzionalità o configurazione specifica dei pool di identità.

Attributi per il controllo degli accessi

Un'implementazione del controllo degli accessi basato sugli attributi nei pool di identità. I pool di identità applicano gli attributi utente come tag alle credenziali utente.

Autenticazione di base (classica)

Un processo di autenticazione in cui è possibile personalizzare la richiesta di credenziali utente.

Identità autenticate dagli sviluppatori

Un processo di autenticazione che autorizza le credenziali utente del pool di identità con le credenziali dello sviluppatore.

Credenziali dello sviluppatore

Le chiavi API IAM di un amministratore di pool di identità.

Autenticazione avanzata

Un flusso di autenticazione che seleziona un ruolo IAM e applica i tag principali in base alla logica definita nel pool di identità.

Identità

Un UUID che collega un utente dell'app e le relative credenziali utente al relativo profilo in una directory utente esterna che ha una relazione di fiducia con un pool di identità.

Pool di identità, identità federate Amazon Cognito, identità Amazon Cognito, **`cognito-identity`**

Una AWS risorsa con servizi di autenticazione e autorizzazione per applicazioni che utilizzano credenziali temporanee. AWS

Identità non autenticata

Un utente che non ha effettuato l'accesso con un IdP del pool di identità. Puoi consentire agli utenti di generare credenziali utente limitate per un singolo ruolo IAM prima dell'autenticazione.

Credenziali utente

Chiavi AWS API temporanee che gli utenti ricevono dopo l'autenticazione del pool di identità.

Nozioni di base su AWS

Prima di iniziare a lavorare con Amazon Cognito, configurati con alcune risorse necessarie. AWS Se riesci già ad accedere a un account Account AWS, puoi saltare questa sezione. Continua a leggere se stai cercando informazioni sulla registrazione e l'accesso con AWS le credenziali. Dopo aver ottenuto credenziali con autorizzazioni AWS Identity and Access Management (IAM) sufficienti, puoi iniziare a utilizzare i pool di utenti e i pool di identità.

Registrati per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Esistono due modi per iscriversi AWS. Se desideri il controllo completo sulla configurazione dell'account e sulla posizione in cui sono archiviati i tuoi contenuti, o se hai carichi di lavoro regolamentati come HIPAA o FedRAMP, usa Sign up for (avanzato). AWS Se sei un nuovo utente e desideri utilizzare un nome di accesso che già possiedi per accedere AWS e creare un AWS ambiente con impostazioni preconfigurate preconfigurate in select, usa Sign up for. Regioni AWSAWS

Nozioni di base su Amazon Cognito

Questo capitolo ti consente di iniziare con l'autenticazione del pool di utenti, ovvero la creazione del primo pool di utenti e l'accesso del primo utente. Amazon Cognito offre anche pool di identità, che scambiano l'identità di un utente con AWS credenziali temporanee; se è ciò di cui hai bisogno, vedi. Scambio di identità in cambio di credenziali AWS Se non sei sicuro di quale componente ti serve, consulta. the section called “Di quali parti di Amazon Cognito hai bisogno?”

> **Note**
>
> È necessario un AWS account per completare la procedura dettagliata e le guide introduttive in questo capitolo. Se non lo hai, consultare the section called “Nozioni di base su AWS”.

Hai un'applicazione che richiede l'autenticazione e il controllo degli accessi. Puoi lavorare all'interno del framework OpenID Connect (OIDC) per il single sign-on (SSO). Amazon Cognito dispone di strumenti per gestire la logica di autenticazione nel backend dell'applicazione con un AWS SDK e per richiamare un browser nel client per accedere a un server di autorizzazione gestito.

La console Amazon Cognito ti guida nella creazione di un pool di utenti dal punto di vista del tuo framework applicativo preferito. Da lì, puoi continuare ad aggiungere funzionalità come l'accesso federato con social network esterni o provider di identità SAML 2.0 (). IdPs I modelli di applicazione nella console Amazon Cognito si basano sull'aggiunta di librerie OIDC al progetto e sull'utilizzo di un browser.

Mentre lavori per espandere il tuo set di funzionalità e incorporare più componenti di Amazon Cognito, leggi il capitolo sui pool di utenti di Amazon Cognito per le descrizioni complete di tutto ciò che puoi fare con i pool di utenti.

Gli esempi in questo capitolo e nella console Amazon Cognito dimostrano un'integrazione di base delle risorse applicative con i pool di utenti di Amazon Cognito. Successivamente, puoi modificare il tuo pool di utenti per utilizzare più opzioni disponibili. Quindi puoi aggiornare l'applicazione per adottare nuove funzionalità e interagire con IdPs.

Se non desideri utilizzare le pagine di accesso gestite, puoi creare un'applicazione con interfacce di autenticazione personalizzate utilizzando un AWS SDK o. AWS AmplifyLe applicazioni create in questo modo interagiscono con l'API dei pool di utenti e sono adatte solo per l'autenticazione degli

utenti locali. Continua a conoscere questo modello di autenticazione all'indirizzo. Altre opzioni di applicazione

Il tuo primo accesso: una procedura dettagliata

Questa pagina è una mappa, non un sostituto delle procedure dettagliate a cui si collega. Mostra come si collegano le fasi di configurazione in modo da poter vedere l'intero percorso prima di iniziare una singola fase. Ogni passaggio numerato si collega alla procedura autorevole.

> **Note**
>
> Questa procedura dettagliata utilizza un pool di utenti con accesso gestito, il punto di partenza più comune per un'applicazione che consente l'accesso degli utenti. L'accesso gestito è l'attuale interfaccia utente di Cognito-hosted accesso di Amazon, con un editor di branding visivo; la precedente interfaccia utente ospitata classica è un'alternativa più semplice. Scegli l'una o l'altra come versione di branding per il dominio del tuo pool di utenti e tale scelta si applica a tutti i client di app che utilizzano il dominio. L'accesso gestito richiede il piano di funzionalità Essentials o Plus: nel piano Lite, l'interfaccia utente ospitata classica è l'unica opzione ospitata. Per le differenze, vedi e. the section called “Login gestito” the section called “Piani di funzionalità del pool di utenti” Se invece il tuo obiettivo sono AWS le credenziali temporanee per una fonte di identità esistente, inizia conthe section called “Nozioni di base sui pool di identità”.

Da account vuoto a token convalidato

1. Crea un pool di utenti. Un pool di utenti è l'elenco degli utenti e il provider di identità che emette i token. Consulta the section called “La tua prima app e il tuo pool di utenti”.
2. Aggiungi un dominio. Le pagine di accesso gestite vengono servite da un dominio di pool di utenti, un dominio con prefisso Amazon Cognito o il tuo dominio personalizzato. Senza un dominio, non esiste una pagina di accesso ospitata a cui inviare gli utenti. Consulta the section called “Configurazione di un dominio”.
3. Crea un client per l'app. Un client di app rappresenta l'applicazione per il pool di utenti. Contiene le impostazioni che controllano l'accesso, come gli URL di callback consentiti e i metodi di accesso. La versione di branding (accesso gestito o interfaccia utente ospitata classica) è impostata sul dominio, non sul client dell'app. Consulta the section called “Client dell'app”.

4. Invia l'utente all'accesso gestito per accedere. Con un dominio e un client per l'app, l'applicazione reindirizza gli utenti alle pagine di accesso gestite per registrarsi e accedere. Amazon Cognito riporta l'utente all'URL di callback con un codice di autorizzazione. Consulta the section called “Login gestito”.
5. Scambia il codice con dei token. L'applicazione scambia il codice di autorizzazione sull'endpoint del token con un token ID, un token di accesso e un token di aggiornamento. Consulta the section called “Endpoint Token”.
6. Convalida il token. La tua applicazione (o la tua API) verifica la firma e le affermazioni del token prima di considerarlo attendibile. Consulta the section called “Verifica di un token JSON Web”.

A questo punto hai un utente registrato e un token convalidato. Per autorizzare quel token tramite la tua API o il tuo back-end, il passaggio successivo più comune, vedi. the section called “Accesso alle risorse ” Se invece devi concedere all'utente AWS credenziali temporanee per chiamare AWS servizi come Amazon S3 direttamente dalla tua app, continua a farlo. the section called “Pool di identità di Amazon Cognito”

Crea una nuova applicazione nella console Amazon Cognito

I pool di utenti aggiungono opzioni di autenticazione alle applicazioni software. Per iniziare nel modo più semplice, accedi alla console Amazon Cognito e segui le istruzioni ivi riportate. Qui il processo di creazione ti guida non solo nella configurazione delle risorse del pool di utenti, ma anche nella configurazione delle parti iniziali dell'applicazione.

Quando sei pronto per iniziare, accedi alla console Amazon Cognito e seleziona il pulsante per creare un nuovo pool di utenti. Il processo di configurazione ti guida attraverso le opzioni di configurazione e del linguaggio di programmazione.

Risorse aggiuntive sui concetti di autenticazione

- Autenticazione con pool di utenti Amazon Cognito
- Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite
- Come funziona l'autenticazione con Amazon Cognito
- Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili

Per creare risorse Amazon Cognito per la tua applicazione

1. Passa alla console di Amazon Cognito. Per assegnare le autorizzazioni al tuo responsabile IAM in modo che possa creare e gestire risorse Amazon Cognito, consulta. AWS politiche gestite per Amazon Cognito La `AmazonCognitoPowerUser` policy è sufficiente per la creazione di pool di utenti.
2. Seleziona Crea pool di utenti dal menu Pool di utenti oppure seleziona Inizia gratuitamente in meno di cinque minuti.
3. In Definisci la tua applicazione, scegli il tipo di applicazione più adatto allo scenario applicativo per il quale desideri creare servizi di autenticazione e autorizzazione.
4. In Assegna un nome all'applicazione, inserisci un nome descrittivo o procedi con il nome predefinito.
5. È necessario effettuare alcune scelte di base in Configura opzioni che supportano impostazioni che non possono essere modificate dopo aver creato il pool di utenti.

    a. In Opzioni per gli identificatori di accesso, indicaci come desideri identificare gli utenti quando effettuano l'accesso. Puoi preferire nomi utente, indirizzi email o numeri di telefono generati dagli utenti. Puoi anche consentire una combinazione di più opzioni. Amazon Cognito accetta le opzioni che configuri qui nel campo nome utente dei moduli di accesso con accesso gestito.

    b. In Attributi richiesti per la registrazione, indicaci quali informazioni utente desideri raccogliere quando gli utenti si registrano per un nuovo account. Nelle pagine di accesso gestite, Amazon Cognito presenta richieste per tutti gli attributi obbligatori.

    Le opzioni per gli identificatori di accesso influiscono sugli attributi richiesti. Il nome utente richiede gli attributi di e-mail o telefono per ogni utente in modo che possa ricevere un codice di reimpostazione della password in un'e-mail o un messaggio SMS. L'e-mail richiede l'attributo email e il numero di telefono richiede l'attributo del numero di telefono.
6. In Aggiungi un URL di ritorno, inserisci un percorso di reindirizzamento all'applicazione per dopo che gli utenti hanno completato l'autenticazione. Questa posizione dovrebbe essere un percorso nell'applicazione che utilizza le librerie OpenID Connect (OIDC) per elaborare i risultati dell'autenticazione dell'utente. Un esempio di URL di ritorno per un'applicazione di test è. `https://localhost:3000/callback` Nell'applicazione NodeJS di esempio nella console Amazon Cognito, questa route utilizza openid-client per raccogliere il token di accesso e convertirlo in informazioni sull'utente. Potrai consultare gli esempi per la tua piattaforma di sviluppo dopo aver creato le tue risorse.

7. Scegli Crea la tua applicazione. Amazon Cognito crea un pool di utenti e un client di app con impostazioni predefinite per il tipo di applicazione. Puoi configurare opzioni aggiuntive come provider di identità esterni e autenticazione a più fattori (MFA) dopo aver creato le risorse iniziali.
8. Nella pagina Configura l'applicazione, puoi ottenere immediatamente esempi di codice per la tua applicazione. Per esplorare il nuovo pool di utenti, scorri verso il basso e seleziona Vai alla panoramica.
9. Per aggiungere altre applicazioni nello stesso pool di utenti, vai al menu App client e aggiungi un nuovo client per l'app. In questo modo si ripete il processo di creazione incentrato sulle applicazioni, ma si aggiunge solo un nuovo client di app al pool di utenti esistente.

Dopo aver creato un pool di utenti e uno o più client di app con questo processo, puoi iniziare a testare le operazioni di autenticazione con l'accesso gestito. Queste opzioni di avvio rapido sono aperte all'iscrizione automatica pubblica. Ti consigliamo di creare un ambiente di test con la procedura della console, quindi passare alla produzione del progetto definitivo. Dedica del tempo a familiarizzare con le funzionalità di Amazon Cognito. Quindi, per passare ai carichi di lavoro di produzione, crea configurazioni personalizzate e distribuiscile con strumenti di automazione come e the. AWS CloudFormation AWS Cloud Development Kit (AWS CDK)

Amazon Cognito crea alcune configurazioni predefinite in questo processo che non è possibile annullare. Per ulteriori informazioni sulle impostazioni del pool di utenti che non puoi modificare e sulle opzioni che puoi scegliere nella console, consulta. Aggiornamento della configurazione del pool di utenti e del client dell'app

Impostazione	Effetto	Come modificare	Ulteriori informazioni
Client secret	Richiede un hash segreto del client nelle richieste di autentica zione.	Crea un nuovo client per l'app con un'applicazione Web tradizionale o un profilo di Machine-to-machine applicazione.	Application-specific impostazioni con i client dell'app
Il nome utente preferito	Il pool di utenti non accetta l'`preferred_username` attributo come alias.	Crea un pool di utenti a livello di codice con un SDK. AWS	Personalizzazione degli attributi di accesso

Impostazione	Effetto	Come modificare	Ulteriori informazioni
Distinzione tra lettere maiuscole e minuscole	I nomi utente del pool di utenti non fanno distinzione tra maiuscole e minuscole, ad esempio `JohnD` viene considerato lo stesso utente di. `johnd`	Crea un pool di utenti a livello di codice con un SDK. AWS	Distinzione tra maiuscole e minuscole del bacino d'utenza

Altre opzioni di applicazione

Potresti avere un'interfaccia utente dell'applicazione esistente che desideri integrare con l'autenticazione Amazon Cognito. Potresti anche avere le tue pagine di autenticazione esistenti con una configurazione di directory meno funzionale rispetto ai pool di utenti di Amazon Cognito. Puoi aggiungere o sostituire un componente di autenticazione a un'applicazione di questo tipo con le integrazioni di Amazon Cognito negli AWS SDK per una varietà di linguaggi di programmazione. Di seguito sono riportati alcuni esempi.

Se crei un pool di utenti per questo scopo nella console Amazon Cognito, potrebbe non essere necessario disporre di un dominio del pool di utenti che ospita pagine di accesso interattive e servizi OpenID Connect (OIDC). Il processo di creazione del pool di utenti nella console genera automaticamente un dominio per te. Puoi eliminare questo dominio dalla scheda Dominio del tuo pool di utenti. Altre opzioni includono la creazione programmatica di risorse Amazon Cognito per la tua applicazione con richieste API AWS negli SDK e con le opzioni di configurazione automatizzate nella CLI. AWS Amplify Per ulteriori informazioni, consulta Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili.

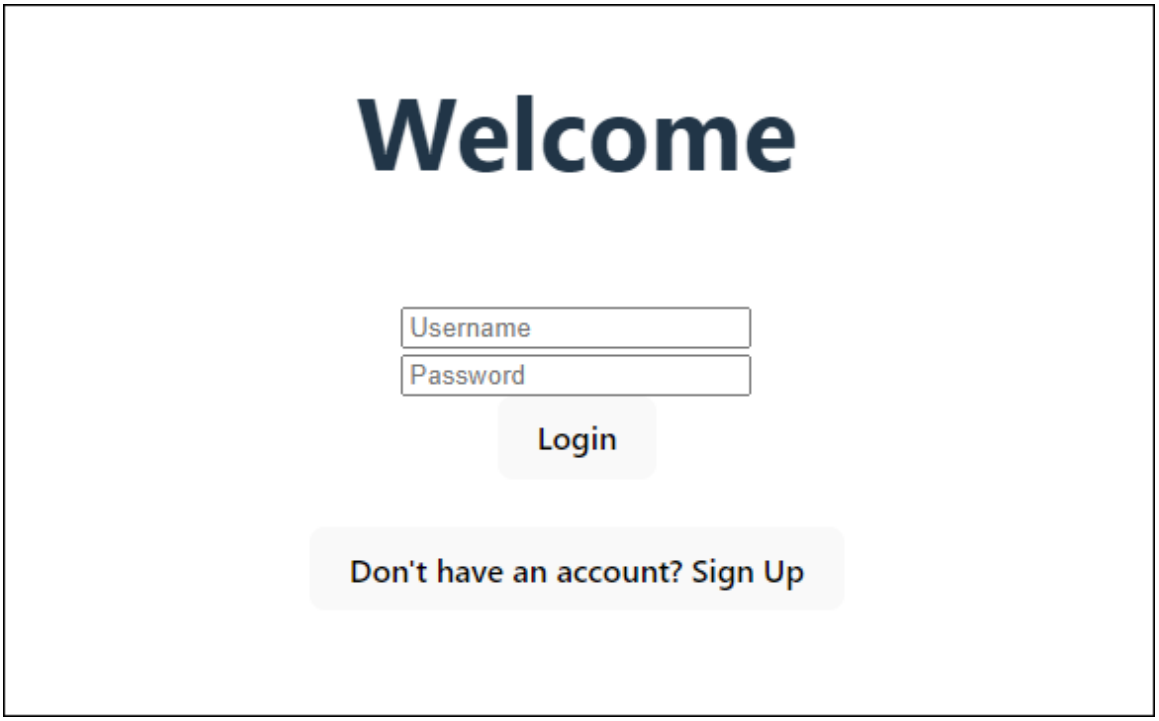
Argomenti

- Configura un esempio di applicazione React a pagina singola
- Configura un'app Android di esempio con Flutter

Configura un esempio di applicazione React a pagina singola

In questo tutorial, creerai un'applicazione React a pagina singola in cui puoi testare la registrazione, la conferma e l'accesso degli utenti. React è una JavaScript-based libreria per app web e mobili, con particolare attenzione all'interfaccia utente (UI). Questa applicazione di esempio illustra alcune funzioni di base dei pool di utenti di Amazon Cognito. Se hai già esperienza nello sviluppo di app web con React, scarica l'app di esempio da. GitHub

La schermata seguente mostra la pagina di autenticazione iniziale dell'applicazione che creerai.



Per configurare questa applicazione, il pool di utenti deve soddisfare i seguenti requisiti:

- Gli utenti possono accedere con il proprio indirizzo e-mail. Opzioni di accesso al pool di utenti di Cognito: e-mail.
- I nomi utente non fanno differenza tra maiuscole e minuscole. Requisiti relativi al nome utente: l'opzione Fai distinzione tra maiuscole e minuscole non è selezionata.
- Multi-factor l'autenticazione (MFA) non è richiesta. Applicazione della MFA: MFA opzionale.
- Il pool di utenti verifica gli attributi per la conferma del profilo utente con un messaggio e-mail. Attributi da verificare: inviare un messaggio e-mail, verificare l'indirizzo e-mail.
- L'email è l'unico attributo obbligatorio. Attributi obbligatori: email.
- Gli utenti possono registrarsi nel tuo pool di utenti. Self-registration: è selezionata l'opzione Abilita l'autoregistrazione.

- Il client iniziale dell'app è un client pubblico che consente l'accesso con nome utente e password. Tipo di app: Client pubblico, Flussi di autenticazione:. `ALLOW_USER_PASSWORD_AUTH`

Creazione di un'applicazione

Per creare questa applicazione, devi configurare un ambiente di sviluppo. I requisiti dell'ambiente di sviluppo sono:

1. Node.js è installato e aggiornato.
2. Node package manager (npm) è installato e aggiornato almeno alla versione 10.2.3.
3. L'ambiente è accessibile sulla porta TCP 5173 in un browser web.

Per creare un esempio di applicazione web React

1. Accedi al tuo ambiente di sviluppo e vai alla directory principale dell'applicazione.

```
cd ~/path/to/project/folder/
```

2. Crea un nuovo servizio React.

```
npm create vite@latest frontend-client -- --template react-ts
```

3. Clona la cartella del `cognito-developer-guide-react-example` progetto dal repository degli esempi di AWS codice in poi. GitHub

```
cd ~/some/other/path
```

```
git clone https://github.com/awsdocs/aws-doc-sdk-examples.git
```

```
cp -r ./aws-doc-sdk-examples/javascriptv3/example_code/cognito-identity-provider/
scenarios/cognito-developer-guide-react-example/frontend-client ~/path/to/project/
folder/
```

4. Vai alla `src` directory del tuo progetto.

```
cd ~/path/to/project/folder/frontend-client/src
```

5. Modifica `config.json` e sostituisci i seguenti valori:

a. Sostituisci `YOUR_AWS_REGION` con un Regione AWS codice. Ad esempio: `us-east-1`.

b. Sostituiscilo `YOUR_COGNITO_USER_POOL_ID` con l'ID del pool di utenti che hai designato per il test. Ad esempio: `us-east-1_EXAMPLE`. Il pool di utenti deve appartenere a Regione AWS quello inserito nel passaggio precedente.

c. Sostituiscilo `YOUR_COGNITO_APP_CLIENT_ID` con l'ID del client dell'app che hai designato per il test. Ad esempio: `1example23456789`. Il client dell'app deve essere incluso nel pool di utenti del passaggio precedente.

6. Se desideri accedere all'applicazione di esempio da un IP diverso da`localhost`, modifica `package.json` e modifica la riga `"dev": "vite",` in`"dev": "vite --host 0.0.0.0",`.
7. Installa la tua applicazione.

```
npm install
```

8. Avvia l'applicazione.

```
npm run dev
```

9. Accedere all'applicazione in un browser Web all'indirizzo `http://localhost:5173` o`http://[IP address]:5173`.
10. Registra un nuovo utente con un indirizzo email valido.
11. Recupera il codice di conferma dal tuo messaggio e-mail. Inserisci il codice di conferma nell'applicazione.
12. Accedi con il tuo nome utente e la tua password.

Creazione di un ambiente di sviluppo React con Amazon Lightsail

Un modo rapido per iniziare a usare questa applicazione è creare un server cloud virtuale con Amazon Lightsail.

Con Lightsail, puoi creare rapidamente una piccola istanza di server preconfigurata con i prerequisiti per questa applicazione di esempio. Puoi accedere tramite SSH alla tua istanza con un client basato su browser e connetterti al server web con un indirizzo IP pubblico o privato.

Per creare un'istanza Lightsail per questa applicazione di esempio

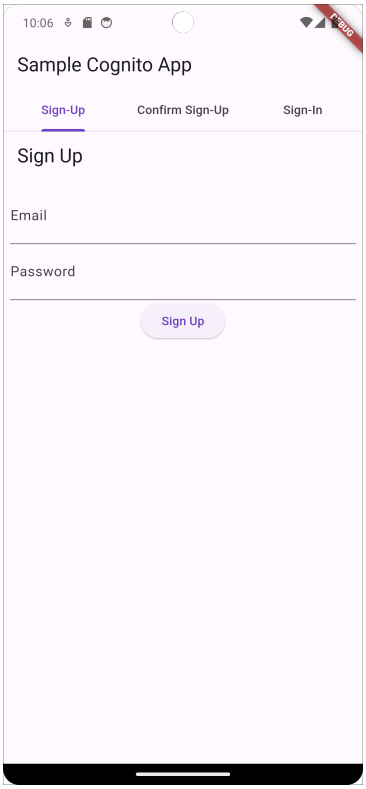
1. Vai alla console Lightsail. Se richiesto, inserisci le tue credenziali. AWS

2. Seleziona Crea istanza.
3. Per Seleziona una piattaforma, scegli. Linux/Unix
4. Per Seleziona un progetto, scegli Node.js.
5. In Identifica la tua istanza, assegna un nome descrittivo al tuo ambiente di sviluppo.
6. Seleziona Crea istanza.
7. Dopo che Lightsail ha creato l'istanza, selezionala e nella scheda Connetti scegli Connetti tramite SSH.
8. Una sessione SSH si apre in una finestra del browser. Esegui `node -v` e `npm -v` conferma che la tua istanza sia stata fornita con Node.js la versione minima di npm 10.2.3.
9. Procedi con la configurazione dell'applicazione React.

Configura un'app Android di esempio con Flutter

In questo tutorial, creerai un'applicazione mobile in Android Studio in cui potrai emulare un dispositivo e testare la registrazione, la conferma e l'accesso degli utenti. Questa applicazione di esempio crea un client mobile di base per pool di utenti Amazon Cognito per Android in Flutter. Se hai già esperienza nello sviluppo di app per dispositivi mobili con Flutter, scarica l'app di esempio da. GitHub

La schermata seguente mostra l'app in esecuzione su un dispositivo Android virtuale.



Per configurare questa applicazione, il pool di utenti deve soddisfare i seguenti requisiti:

- Gli utenti possono accedere con il proprio indirizzo e-mail. Opzioni di accesso al pool di utenti di Cognito: e-mail.
- I nomi utente non fanno differenza tra maiuscole e minuscole. Requisiti relativi al nome utente: l'opzione Fai distinzione tra maiuscole e minuscole non è selezionata.
- Multi-factor l'autenticazione (MFA) non è richiesta. Applicazione della MFA: MFA opzionale.
- Il pool di utenti verifica gli attributi per la conferma del profilo utente con un messaggio e-mail. Attributi da verificare: inviare un messaggio e-mail, verificare l'indirizzo e-mail.
- L'email è l'unico attributo obbligatorio. Attributi obbligatori: email.
- Gli utenti possono registrarsi nel tuo pool di utenti. Self-registration: è selezionata l'opzione Abilita l'autoregistrazione.
- Il client iniziale dell'app è un client pubblico che consente l'accesso con nome utente e password. Tipo di app: Client pubblico, Flussi di autenticazione:. `ALLOW_USER_PASSWORD_AUTH`

Creazione di un'applicazione

Per creare un'app Android di esempio

1. Installa Android Studio e gli strumenti da riga di comando.
2. In Android Studio, installa il plug-in Flutter.
3. Crea un nuovo progetto Android Studio dal contenuto della `cognito_flutter_mobile_app` directory in questa app di esempio.
   - Modifica `assets/config.json` e sostituisci `<<YOUR USER POOL ID>>` e `<< YOUR CLIENT ID>>` con gli ID del tuo pool di utenti e del client dell'app.
4. Installa Flutter.
   a. Aggiungi Flutter alla tua variabile PATH.
   b. Accetta le licenze con il seguente comando.

      ```
      flutter doctor --android-licenses
      ```

   c. Verifica il tuo ambiente Flutter e installa i componenti mancanti.

      ```
      flutter doctor
      ```

- Se mancano dei componenti, esegui `flutter doctor -v` per scoprire come risolvere il problema.

d. Passa alla directory del tuo nuovo progetto Flutter e installa le dipendenze.

- Esegui `flutter pub add amazon_cognito_identity_dart_2`.

e. Esegui `flutter pub add flutter_secure_storage`.

5. Crea un dispositivo Android virtuale.

1. Nella GUI di Android Studio, crea un nuovo dispositivo con il gestore dispositivi.
2. Nella CLI, esegui. `flutter emulators --create --name android-device`

6. Avvia il tuo dispositivo Android virtuale.

1. Nella GUI di Android Studio, seleziona l' icona di avvio accanto al tuo dispositivo virtuale.
2. Nella CLI, esegui. `flutter emulators --launch android-device`

7. Avvia l'app sul tuo dispositivo virtuale.

1. Nella GUI di Android Studio, seleziona l'icona di distribuzione.



2. Nella CLI, esegui. `flutter run`

8. Accedi al tuo dispositivo virtuale in esecuzione in Android Studio.
9. Registra un nuovo utente con un indirizzo email valido.
10. Recupera il codice di conferma dal tuo messaggio e-mail. Inserisci il codice di conferma nell'applicazione.
11. Accedi con il tuo nome utente e la tua password.

Aggiungi altre funzionalità e opzioni di sicurezza al tuo pool di utenti

Dopo aver seguito i tutorial per completare esempi di applicazioni, è possibile ampliare l'ambito di implementazione del pool di utenti. Oppure, se non hai creato un'applicazione di prova, crea un nuovo pool di utenti in base alle tue preferenze. Puoi personalizzare le funzionalità del pool di utenti

per altre applicazioni o aggiungere provider di identità esterni. Mentre pianifichi il passaggio ai pool di utenti di Amazon Cognito nelle applicazioni di produzione, puoi valutare esempi e tutorial aggiuntivi.

Se la tua prossima priorità è esaminare e applicare le opzioni di sicurezza delle applicazioni nei tuoi pool di utenti, consulta. Best practice di sicurezza per i bacini d'utenza di Amazon Cognito

Amazon Cognito offre piani di funzionalità che aggiungono opzioni funzionali e di sicurezza quando opti per livelli superiori. Puoi iniziare con il piano Lite, aggiungere opzioni avanzate di autenticazione e autorizzazione con il piano Essentials e aggiungere barriere di sicurezza con ragionamento automatico con il piano Plus. Per ulteriori informazioni, consulta Piani di funzionalità del pool di utenti.

Di seguito sono riportate alcune funzionalità aggiuntive dei pool di utenti di Amazon Cognito:

- Applica il branding alle pagine di accesso gestite
- Aggiunta dell'autenticazione MFA a un bacino d'utenza
- Sicurezza avanzata con protezione dalle minacce
- Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda
- Utilizzo di Amazon Pinpoint per l'analisi dei pool di utenti

Per una panoramica dei modelli di autenticazione e autorizzazione di Amazon Cognito, consulta. Come funziona l'autenticazione con Amazon Cognito

Per accedere ad altri Servizi AWS dopo una corretta autenticazione del pool di utenti, consultaAccesso Servizi AWS utilizzo di un pool di identità dopo l'accesso.

Oltre a utilizzare gli SDK Console di gestione AWS e il pool di utenti, puoi anche gestire i tuoi pool di utenti utilizzando il AWS Command Line Interface.

Argomenti

- Aggiungi l'accesso tramite social al tuo pool di utenti
- Aggiungi un provider di identità SAML 2.0

Aggiungi l'accesso tramite social al tuo pool di utenti

Offrire agli utenti la possibilità di accedere alla tua applicazione tramite i loro provider di identità pubblici o social esistenti può migliorare la loro esperienza di autenticazione. I pool di utenti di

Amazon Cognito si integrano con i più diffusi provider di identità social (IdPs) come Facebook, Google, Amazon e Apple, offrendo agli utenti comode opzioni di accesso che già conoscono.

Quando configuri l'accesso tramite social, offri ai tuoi utenti un'alternativa alla creazione di un account dedicato solo per la tua applicazione. Ciò può migliorare i tassi di conversione e rendere più semplice la procedura di registrazione. Dal punto di vista dell'utente, può applicare le credenziali social esistenti per autenticarsi rapidamente, senza doversi preoccupare di ricordare un altro nome utente e una password.

La configurazione di un IdP social nel tuo pool di utenti richiede alcuni passaggi chiave. È necessario registrare l'applicazione presso il provider social per ottenere un ID cliente e un segreto. Quindi puoi aggiungere la configurazione social IdP al tuo pool di utenti, specificando gli ambiti che desideri richiedere e gli attributi del pool di utenti che desideri mappare dagli attributi IdP. In fase di esecuzione, Amazon Cognito gestisce lo scambio di token con il provider, mappa gli attributi utente ed emette token all'applicazione nel formato del pool di utenti condiviso.

Registrazione con un IdP social

Prima di creare un IdP social con Amazon Cognito, è necessario registrare l'applicazione con l'IdP social in modo da ricevere un ID client e un segreto client.

Registrazione di un'app con Facebook

1. Creazione di un account sviluppatore con Facebook.
2. Accedi con le tue credenziali di Facebook.
3. Nel menu My Apps (Le mie app), scegli Create New App (Crea nuova app).

   Se non disponi di un'app Facebook esistente, vedrai un'opzione diversa. Scegli Create App (Crea app).
4. Nella pagina Crea un'app, scegli un caso d'uso per l'app, quindi scegli Avanti.
5. Inserisci un nome per la app Facebook, quindi scegli Crea l'app.
6. Nella barra di navigazione a sinistra, scegli Impostazioni app e quindi Base.
7. Prendi nota di ID app e Segreto app. Li utilizzerai nella sezione successiva.
8. Nella parte inferiore della pagina, scegli + Aggiungi piattaforma.
9. Nella schermata Seleziona piattaforma, seleziona le tue piattaforme, quindi scegli Avanti.
10. Scegli Save changes (Salva modifiche).
11. Per i domini dell'app, inserisci il dominio del bacino d'utenza.

```
https://your_user_pool_domain
```

12. Scegli Save changes (Salva modifiche).
13. Dalla barra di navigazione, scegli Prodotti, quindi scegli Configura da Facebook Login.
14. Nel menu Facebook Login Configura, scegli Impostazioni.

    Inserisci il tuo URL di reindirizzamento nel campo Valid OAuth Redirect URIs (URI di reindirizzamento OAuth validi). L'URL di reindirizzamento è costituito dal dominio del pool di utenti con l'`/oauth2/idpresponse`endpoint.

    ```
    https://your_user_pool_domain/oauth2/idpresponse
    ```

15. Scegli Save changes (Salva modifiche).

Registrazione di un'app con Amazon

1. Creazione di un account sviluppatore con Amazon.
2. Accedi con le tue credenziali di Amazon.
3. È necessario creare un profilo di sicurezza Amazon per ricevere l'ID client di Amazon e il segreto client.

   Scegli App e servizi dalla barra di navigazione nella parte superiore della pagina, quindi scegli Login with Amazon.
4. Scegli Create a Security Profile (Crea un profilo di sicurezza).
5. Digita un Security Profile Name (nome profilo sicurezza), una Security Profile Description (descrizione profilo sicurezza) e un Consent Privacy Notice URL (URL consenso informativa privacy).
6. Scegli Save (Salva).
7. Scegli Client ID (ID client) e Client Secret (Segreto client) per mostrare i dati relativi. Li utilizzerai nella sezione successiva.
8. Passa il mouse sull'icona a forma di ingranaggio e scegli Web Settings (Impostazioni Web), quindi Scegli Edit (Modifica).
9. Inserisci il dominio del bacino d'utenza nel campo Allowed Origins (origini consentite).

   ```
   https://<your-user-pool-domain>
   ```

10. Digita il dominio del bacino d'utenza con l'endpoint `/oauth2/idpresponse` in Allowed Return URLs (URL restituiti consentiti).

    ```
    https://<your-user-pool-domain>/oauth2/idpresponse
    ```

11. Scegli Save Salva.

Registrazione di un'app con Google

Per ulteriori informazioni su OAuth 2.0 nella piattaforma Google Cloud, consulta la sezione relativa all'autenticazione e autorizzazione nella documentazione disponibile nella pagina Google Workspace for Developers.

1. Creazione di un account sviluppatore con Google.
2. Accedi alla console Piattaforma Google Cloud.
3. Nella barra di navigazione in alto, scegli Select a project (Seleziona un progetto). Se hai già un progetto nella piattaforma Google, nel menu viene visualizzato il tuo progetto predefinito.
4. Scegli Nuovo progetto.
5. Immetti un nome per il progetto e scegli Crea.
6. Nella barra di navigazione a sinistra, scegli API e servizi, quindi scegli la schermata di consenso Oauth.
7. Inserisci le informazioni sull'app, un dominio dell'app, i domini autorizzati e le informazioni di contatto dello sviluppatore. I domini autorizzati devono includere `amazoncognito.com` e la radice del dominio personalizzato. Ad esempio: `example.com`. Seleziona Salva e continua.
8. 1. In Ambiti, scegli Aggiungi o rimuovi ambiti, quindi scegli almeno i seguenti ambiti OAuth.

    1. `.../auth/userinfo.email`
    2. `.../auth/userinfo.profile`
    3. openid
9. In Test users (Utenti di test), scegli Add users (Aggiungi utenti). Inserisci il tuo indirizzo e-mail e tutti gli altri utenti autorizzati al test, quindi scegli SALVA E CONTINUA.
10. Espandi nuovamente la barra di navigazione a sinistra, scegli API e servizi, quindi scegli Credenziali.
11. Scegli CREATE CREDENTIALS, quindi scegli ID client OAuth.
12. Scegli un Application type (Tipo di applicazione) e assegna un Name (Nome) al client.

13. In JavaScript Origini autorizzate, scegli AGGIUNGI URI. Immetti il dominio del pool di utenti.

    ```
    https://<your-user-pool-domain>
    ```

14. In Authorized redirect URIs (URI di reindirizzamento autorizzati), scegli ADD URI (Aggiungi URI). Inserisci il percorso dell'endpoint `/oauth2/idpresponse` del dominio del pool di utenti.

    ```
    https://<your-user-pool-domain>/oauth2/idpresponse
    ```

15. Seleziona CREATE.
16. Memorizza in modo sicuro i valori visualizzati da Google in Il tuo ID client e Il tuo segreto del client. Fornisci questi valori ad Amazon Cognito quando aggiungi un gestore dell'identità digitale (IdP) Google.

Come registrare un'app con Apple

Per ulteriori informazioni sulla configurazione della funzionalità di accesso con Apple, consulta la sezione relativa alla configurazione dell'ambiente per la funzionalità di accesso con Apple nella documentazione per gli sviluppatori Apple.

1. Creazione di un account sviluppatore con Apple.
2. Accedi con le tue credenziali Apple.
3. Sulla barra di navigazione a sinistra, scegli Certificates, Identifiers & Profiles (Certificati, identificatori e profili).
4. Nel riquadro di navigazione a sinistra, scegli Identifiers (Identificatori).
5. Nella pagina Identifiers (Identificatori), scegli l'icona +.
6. Nella pagina Register a New Identifier (Registra un nuovo identificatore), scegli App IDs (ID app), quindi scegli Continue (Continua).
7. Nella pagina Seleziona un tipo, scegli App, quindi scegli Continua.
8. Nella pagina Register an App ID (Registra un'ID app), procedi come indicato di seguito:

    1. In Description (Descrizione), inserisci una descrizione.
    2. In App ID Prefix (Prefisso ID app), inserisci un Bundle ID ID bundle. Prendi nota del valore alla voce Prefisso ID app. Ne avrai bisogno per configurare Apple come provider di identità in Utilizzo di provider di identità social con un pool di utenti.
    3. In Funzionalità, scegli Accedi con Apple, quindi scegli Modifica.

4. Nella pagina Sign in with Apple: App ID Configuration (Accedi con Apple: configurazione ID app) scegli di configurare l'app come app primaria o raggruppata con altri ID app, quindi scegli Save (Salva).
5. Scegli Continua.

9. Nella pagina Confirm your App ID (Conferma l'ID app), scegli Register (Registra).
10. Nella pagina Identifiers (Identificatori), scegli l'icona +.
11. Nella pagina Register a New Identifier (Registra un nuovo identificatore), seleziona Services IDs (ID servizi), quindi scegli Continue (Continua).
12. Nella pagina Register a Services ID (Registra un ID servizi), procedi nel modo seguente:

    1. In Description (Descrizione), inserisci una descrizione.
    2. Alla voce Identifier (Identificatore), inserisci un identificatore. Prendi nota di questo ID dei servizi perché avrai bisogno di questo valore dopo aver scelto Apple come provider di identità inUtilizzo di provider di identità social con un pool di utenti.
    3. Scegli Continua, quindi scegli Registra.
13. Scegli l'ID dei servizi che hai appena creato dalla pagina Identificatori.

    1. Seleziona Accedi con Apple, quindi scegli Configura.
    2. Nella pagina Web Authentication Configuration (Configurazione autenticazione Web), seleziona l'ID dell'app creato in precedenza come Primary App ID (ID app principale).
    3. Scegli l'icona + accanto a Website URLs (URL siti Web).
    4. In Domains and subdomains (Domini e sottodomini), inserisci il dominio del pool di utenti senza prefisso `https://`.

       ```
       <your-user-pool-domain>
       ```

    5. In Return URLs (URL restituiti), inserisci il percorso dell'endpoint `/oauth2/idpresponse` del dominio del pool di utenti.

       ```
       https://<your-user-pool-domain>/oauth2/idpresponse
       ```

    6. Scegli Avanti, quindi scegli Fine. Non è necessario verificare il dominio.
    7. Scegli Continua, quindi Salva.
14. Nel riquadro di navigazione a sinistra, scegli Keys (Chiavi).
15. Nella pagina Keys (Chiavi), scegli l'icona +.

16. Nella pagina Register a New Key, (Registra una nuova chiave) procedi nel modo seguente:

    1. Alla voce Key Name (nome chiave), digita un nome della chiave.
    2. Scegli Accedi con Apple, quindi scegli Configura.
    3. Nella pagina Configure Key, seleziona l'ID dell'app che hai creato in precedenza come ID app principale. Selezionare Salva.
    4. Scegli Continue (Continua), quindi scegli Register (Registra).

17. Nella pagina Scarica la tua chiave, scegli Scarica per scaricare la chiave privata, prendi nota dell'ID chiave mostrato, quindi scegli Fine. Avrai bisogno di questa chiave privata e del valore di Key ID (ID chiave) visualizzato in questa pagina dopo aver scelto Apple come provider di identità in Utilizzo di provider di identità social con un pool di utenti.

Aggiunta di un IdP social al bacino d'utenza

In questa sezione viene configurato un IdP social nel bacino d'utenza utilizzando l'ID client e il segreto client della sezione precedente.

Per configurare un provider di identità social per un pool di utenti con Console di gestione AWS

1. Passa alla console Amazon Cognito. È possibile che ti vengano richieste le AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Social e provider esterni. Individua Federated sign-in (accesso federato) e seleziona Add an identity provider (aggiungi un provider di identità).
5. Scegli un provider di identità social: Facebook, Google, Login with Amazon o Accedi con Apple.
6. Scegli uno tra i seguenti passaggi, in base alla tua scelta del provider di identità social:

    - Google e Login with Amazon: inserisci l'ID client dell'app e il segreto del client dell'app generati nella sezione precedente.
    - Facebook: inserisci l'ID client dell'app e il segreto del client dell'app generati nella sezione precedente, quindi scegli una versione dell'API (ad esempio, la versione 2.12). Ti consigliamo di scegliere la versione più recente possibile: ogni API di Facebook ha un ciclo di vita e una data di obsolescenza. Gli ambiti e gli attributi di Facebook possono variare a seconda delle versioni dell'API. Ti consigliamo di testare l'accesso alla tua identità social con Facebook per assicurarti che la federazione funzioni come previsto.

- Accedi con Apple: inserisci l'ID dei servizi, l'ID del team, l'ID chiave e la chiave privata generati nella sezione precedente.

7. Inserisci i nomi degli ambiti autorizzati che desideri utilizzare. Gli ambiti definiscono a quali attributi utente (ad esempio `name` e `email`) intendi accedere con l'app. Per Facebook, devono essere separati da virgole. Per Google e Login with Amazon, devono essere separati da spazi. Per Accedi con Apple, seleziona le caselle di controllo relative agli ambiti a cui desideri accedere.

Provider di identità social	Ambiti di esempio
Facebook	`public_profile, email`
Google	`profile email openid`
Login with Amazon	`profile postal_code`
Accedi con Apple	`email name`

   All'utente dell'app viene richiesto il consenso a fornire questi attributi all'app. Per ulteriori informazioni sugli ambiti dei provider di social, consulta la documentazione di Google, Facebook, Login with Amazon e Accedi con Apple.

   Nel caso di Accedi con Apple, di seguito sono riportati scenari utente in cui gli ambiti potrebbero non essere restituiti:

   - Un utente finale riscontra degli errori dopo aver lasciato la pagina di accesso di Apple (possono essere dovuti a errori interni di Amazon Cognito o a qualsiasi altro errore scritto dallo sviluppatore).
   - L'identificatore ID del servizio viene utilizzato tra pool di utenti e altri servizi di autenticazione. and/or
   - Uno sviluppatore aggiunge altri ambiti dopo che l'utente ha effettuato l'accesso. Gli utenti recuperano nuove informazioni solo quando si autenticano e quando aggiornano i token.
   - Uno sviluppatore elimina l'utente e quindi l'utente accede nuovamente senza rimuovere l'app dal proprio profilo ID Apple.

8. Mappa gli attributi dal provider di identità al bacino d'utenza. Per ulteriori informazioni, consulta Cose da sapere sulle mappature.
9. Scegli Crea.

10. Dal menu App client, scegli uno dei client dell'app nell'elenco e Modifica le impostazioni dell'interfaccia utente ospitata. Aggiungi il nuovo provider di identità social al client dell'App alla voce Identity providers (Provider di identità).
11. Scegli Save changes (Salva modifiche).

Test della configurazione dell'IdP social

Puoi creare un URL di accesso utilizzando gli elementi delle due sezioni precedenti. Utilizzalo per testare la tua configurazione IdP social.

```
https://mydomain.auth.us-east-1.amazoncognito.com/login?
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

Per individuare il dominio, vai alla pagina della console Domain name (Nome dominio) del bacino d'utenza. Il valore di client_id è disponibile nella pagina App client settings Impostazioni client di applicazioni. Utilizza l'URL di callback per il parametro redirect_uri. Questo è l'URL della pagina a cui l'utente verrà reindirizzato dopo aver completato la procedura di autenticazione.

> **Note**
>
> Amazon Cognito annulla le richieste di autenticazione che non vengono completate entro 5 minuti e reindirizza l'utente all'accesso gestito. Viene visualizzato il messaggio di errore `Something went wrong` nella pagina.

Aggiungi un provider di identità SAML 2.0

Gli utenti della tua app possono accedere con un provider di identità SAML 2.0 (IdP). Potresti scegliere SAML 2.0 IdPs rispetto ai social IdPs quando i tuoi clienti sono clienti interni o aziende collegate alla tua organizzazione. Laddove un IdP social consente a tutti gli utenti di registrare un account, è più probabile che un IdP SAML si abbini a una directory di utenti controllata dalla tua organizzazione. Sia che effettuino l'accesso direttamente o attraverso terze parti, tutti gli utenti hanno un profilo nel bacino d'utenza. Salta questa fase se non desideri aggiungere l'accesso attraverso un provider di identità SAML.

Per ulteriori informazioni, consulta Utilizzo di provider di identità SAML con un pool di utenti.

Devi aggiornare il tuo provider di identità SAML e configurare il tuo pool di utenti. Per informazioni su come aggiungere il tuo pool di utenti come relying party o applicazione per il tuo provider di identità SAML 2.0, consulta la documentazione del tuo provider di identità SAML.

È inoltre necessario fornire un endpoint ACS (Assertion Consumer Service) al provider di identità SAML. Configura il seguente endpoint nel dominio del pool di utenti per il binding SAML 2.0 POST nel gestore dell'identità digitale SAML. Per ulteriori informazioni sui domini del pool di utenti, consulta. Configurazione di un dominio di bacino d'utenza

```
https://Your user pool domain/saml2/idpresponse
With an Amazon Cognito domain:
https://<yourDomainPrefix>.auth.<region>.amazoncognito.com/saml2/idpresponse
With a custom domain:
https://Your custom domain/saml2/idpresponse
```

Puoi trovare il prefisso del dominio e il valore della regione per il tuo pool di utenti nel menu Dominio della console Amazon Cognito.

Per alcuni provider di identità SAML, devi anche fornire il service provider (SP)`urn`, chiamato anche URI del pubblico o ID dell'entità SP, nel formato:

```
urn:amazon:cognito:sp:<yourUserPoolID>
```

Puoi trovare l'ID del tuo pool di utenti nella dashboard Panoramica del tuo pool di utenti nella console Amazon Cognito.

È inoltre necessario configurare il provider di identità SAML per fornire i valori di attributo per tutti gli attributi obbligatori nel bacino d'utenza. Di solito, `email` è un attributo obbligatorio per i bacini d'utenza. In questo caso, il provider di identità SAML deve fornire un valore `email` (attestazione) nell'asserzione SAML.

I bacini d'utenza di Amazon Cognito supportano la federazione SAML 2.0 con endpoint post-binding. Ciò elimina la necessità per l'app di recuperare o analizzare le risposte alle asserzioni SAML, poiché il pool di utenti riceve direttamente la risposta SAML dal tuo provider di identità tramite uno user agent.

Per configurare un provider di identità SAML 2.0 nel bacino d'utenza

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Scegli User Pools (Pool di utenti).

3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Social e provider esterni. Individua Federated sign-in (accesso federato) e seleziona Add an identity provider (aggiungi un provider di identità).
5. Scegli un provider SAML di identità social.
6. Inserisci gli Identifiers (identificatori) separati da virgole. Un identificatore indica ad Amazon Cognito che deve controllare l'indirizzo e-mail che un utente inserisce quando effettua l'accesso. Quindi li indirizza al provider che corrisponde al loro dominio.
7. Scegli Add sign-out flow (Aggiungi flusso di disconnessione) se desideri che Amazon Cognito invii richieste di disconnessione firmate al tuo provider quando un utente si disconnette. Devi configurare il tuo provider di identità SAML 2.0 per inviare risposte di disconnessione all'`https://<your Amazon Cognito domain>/saml2/logout`endpoint creato quando configuri l'accesso gestito. L'`saml2/logout`endpoint utilizza l'associazione POST.

   > **Note**
   >
   > Se questa opzione è selezionata e il tuo provider di identità SAML prevede una richiesta di disconnessione firmata, dovrai anche configurare il certificato di firma fornito da Amazon Cognito con il tuo IdP SAML.
   > L'IdP SAML elabora la richiesta di disconnessione firmata e disconnette l'utente dalla sessione di Amazon Cognito.

8. Scegli una Metadata document source (Fonte del documento di metadati). Se il tuo provider di identità fornisce metadati SAML a un URL pubblico, puoi scegliere l'opzione Metadata document URL (URL del documento di metadati) e inserire l'URL pubblico. In caso contrario, seleziona Upload metadata document (Carica documento di metadati) e seleziona un file di metadati scaricato dal tuo provider in precedenza.

   > **Note**
   >
   > Ti consigliamo di inserire l'URL di un documento di metadati se il tuo provider ha un endpoint pubblico, anziché caricare un file. Ciò consente ad Amazon Cognito di aggiornare automaticamente i metadati. In genere, l'aggiornamento dei metadati avviene ogni 6 ore oppure prima della scadenza dei metadati, in base a ciò che avviene prima.

9. Seleziona Map attributes between your SAML provider and your app (mappare gli attributi tra il provider SAML e la tua app) per mappare gli attributi del provider SAML al profilo utente nel bacino d'utenza. Includi gli attributi richiesti del bacino d'utenza nella mappa degli attributi.

Ad esempio, quando si scegli l'User pool attribute (attributo del bacino d'utenza) `email`, inserisci il nome dell'attributo SAML come compare nell'asserzione SAML dal provider di identità. Il tuo provider di identità potrebbe offrire esempi di asserzioni SAML come riferimento. Alcuni provider di identità utilizzano nomi semplici, ad esempio`email`, mentre altri utilizzano nomi di URL-formatted attributi, come l'esempio seguente:

```
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
```

10. Scegli Crea.

Opzioni di configurazione guidate per Amazon Cognito

Potresti voler valutare le funzionalità di Amazon Cognito in un'esperienza strutturata e guidata. Ecco alcune risorse esterne che forniscono esperienze personalizzate con pool di utenti e pool di identità.

Completa un workshop

AWS workshop studio ospita un workshop che illustra la configurazione della maggior parte delle funzionalità di Amazon Cognito. Queste funzionalità includono l'API dei pool di utenti, l'interfaccia utente ospitata dai pool di utenti, i pool di identità e la configurazione di sicurezza.

Aggiungi codice applicativo tratto da esempi

Il capitolo sugli esempi di codice di questa guida contiene codice applicativo che è possibile utilizzare con pool di utenti e pool di identità. La sezione pool di utenti del capitolo sugli esempi di codice contiene brevi frammenti che coprono singole operazioni ed esempi più lunghi per applicazioni di esempio end-to-end in una varietà di linguaggi di programmazione.

Crea un'applicazione full stack con AWS Amplify

AWS Amplifyè Servizio AWS destinato agli sviluppatori che desiderano sviluppare e ospitare un'applicazione e un'interfaccia utente. Amazon Cognito è il componente di autenticazione di Amplify. Quando aggiungi l'autenticazione alla tua applicazione, Amplify può automatizzare la distribuzione del pool di utenti e delle risorse del pool di identità di Amazon Cognito. Consulta anche Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili.

Altre risorse sulle applicazioni Amazon Cognito su GitHub

- Esempi di flussi di autenticazione con.NET per Amazon Cognito
- Autenticazione senza password di Amazon Cognito
- PetStoreesempio con Amazon Verified Permissions
- Esempio di app React che utilizza ABAC + Identity Pools per accedere alle risorse AWS
- Autorizzazione da macchina a macchina basata su Amazon Cognito e API Gateway tramite CDK AWS
- Creazione di autorizzazioni granulari utilizzando Amazon Cognito, API Gateway e IAM
- CloudFrontautorizzazione @edge

Altri workshop

- Implementa l'autenticazione senza password con Amazon Cognito e WebAuthn
- Workshop Amazon Cognito
- Workshop sulla risoluzione dei problemi di Amazon Cognito
- Autenticazione e autorizzazione con Amazon Cognito e Autorizzazioni verificate
- Approfondimento su Amazon Cognito JWT

Post sul blog

- Proteggi i client pubblici per Amazon Cognito utilizzando un proxy Amazon CloudFront
- Come configurare Amazon Cognito per l'autenticazione federata usando Azure AD
- Semplifica l'autenticazione delle app Web: una guida alla federazione AD FS con i pool di utenti di Amazon Cognito

Pianificazione del design di Amazon Cognito

Le scelte in questo capitolo sono più facili da fare in anticipo che da modificare in seguito. Questo capitolo tratta come:

- Scelta di un approccio di autenticazione: il modo in cui gli utenti accedono.
- Scenari comuni: modelli di architettura da abbinare alla tua applicazione.
- Multi-tenancy—se servite più di un'organizzazione cliente, vengono illustrate le strategie per isolare gli utenti e i dati di ciascun tenant. Creazione di applicazioni multi-tenant B2B

Scegli il tuo approccio di autenticazione

Quando autentichi gli utenti con un pool di utenti di Amazon Cognito, scegli come presentare l'esperienza di accesso. I tre approcci seguenti differiscono nella quantità di interfaccia fornita da Amazon Cognito rispetto a quella creata, e non si escludono a vicenda: è possibile combinarli. Si autenticano tutti sullo stesso pool di utenti e producono gli stessi token. La differenza è dove si trovano le pagine di accesso e quanto controllo hai.

Accesso gestito (Amazon Cognito ospita le pagine)

> Amazon Cognito fornisce le pagine di registrazione, accesso, autenticazione a più fattori e reimpostazione della password del dominio del tuo pool di utenti. Puoi personalizzarne l'aspetto con l'editor di branding. Scegli questa opzione quando desideri un'esperienza di accesso completa e ospitata senza creare un'interfaccia utente e quando desideri che i flussi OAuth 2.0 e OpenID Connect basati su standard vengano gestiti per te. L'accesso gestito è la versione attuale di queste Cognito-hosted pagine Amazon; la precedente interfaccia utente ospitata classica è un'alternativa più semplice di prima generazione che puoi comunque selezionare per client di app. Consulta the section called “Login gestito”.

API dei pool di utenti (l'interfaccia è creata dall'utente)

> L'applicazione raccoglie le credenziali e chiama l'API dei pool di utenti di Amazon Cognito direttamente tramite un SDK. AWS Crei e controlli l'intera interfaccia. Scegli questa opzione quando hai bisogno di un'esperienza di accesso completamente personalizzata o quando accedi utenti da un servizio di back-end. Consulta the section called “Autenticazione”.

Third-party provider di identità (federazione)

I tuoi utenti accedono tramite un provider esterno. Può trattarsi di un provider di servizi social come Google o Apple o di un provider SAML o OIDC aziendale. Amazon Cognito federa la loro identità nel tuo pool di utenti. L'accesso federato viene eseguito tramite gli endpoint di accesso ospitati del dominio del pool di utenti: accesso gestito o endpoint di autorizzazione e accesso a cui reindirizzi da un'interfaccia personalizzata. Il funzionamento dell'API dei pool di utenti nativi `InitiateAuth` non consente l'accesso di un utente tramite un provider federato, quindi la federazione utilizza sempre gli endpoint ospitati anziché il flusso di autenticazione SDK diretto. Scegli questa opzione quando i tuoi utenti hanno già identità altrove. Consulta the section called “Third-party Accesso IdP”.

La tabella seguente riassume quando adottare ciascun approccio.

Se vuoi	Inizia con
Effettua rapidamente l'accesso senza alcuna interfaccia utente da creare e mantenere	Login gestito
Controlla completamente l'aspetto e il comportamento di ogni schermata di accesso	L'API dei pool di utenti
Autentica le richieste provenienti da un chiamante back-end o da macchina a macchina	L'API dei pool di utenti (e, per M2M, le credenziali del client)
Consenti agli utenti di accedere con Google, Apple o una directory aziendale	Third-party provider di identità (federazione), tramite accesso gestito o endpoint di autorizzazione e accesso ospitati

> **Note**
>
> Indipendentemente dall'approccio scelto, il piano di funzionalità del pool di utenti determina quali funzionalità sono disponibili. Il piano Lite prevede l'iscrizione e l'accesso di base;

Essentials aggiunge opzioni come l'accesso senza password e con chiave di accesso e l'accesso gestito; e Plus aggiunge una protezione avanzata dalle minacce. Seleziona un piano quando crei un pool di utenti e puoi modificarlo in un secondo momento, in modo da non dover decidere prima di iniziare. Se una funzionalità che ti aspetti non è disponibile, controlla prima il tuo piano. Per un confronto completo, consultathe section called “Piani di funzionalità del pool di utenti”.

> **Note**
>
> Tutti questi approcci rispondono chi è questo utente? (autenticazione). Se devi anche concedere agli utenti AWS credenziali temporanee per accedere a servizi come Amazon S3, questa è una domanda a parte, l'autorizzazione, a cui rispondono i pool di identità. Consulta the section called “Di quali parti di Amazon Cognito hai bisogno?”.

Scenari comuni di Amazon Cognito

Questo argomento descrive sette scenari comuni per l'utilizzo di Amazon Cognito.

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 Web e per dispositivi mobili. I pool di identità forniscono AWS credenziali temporanee per consentire agli utenti di accedere ad altri utenti. Servizi AWS

Un bacino d'utenza è una directory di utenti in Amazon Cognito. Gli utenti della tua app possono accedere direttamente tramite un pool di utenti oppure possono federarsi tramite un provider di identità (IdP) di terze parti. Il pool di utenti gestisce il sovraccarico di gestione dei token restituiti dall'accesso social tramite Facebook, Google, Amazon e Apple e da OpenID Connect (OIDC) e SAML. IdPs Sia se gli utenti effettuano l'accesso direttamente o tramite terze parti, tutti i membri del bacino d'utenza dispongono di un profilo di directory a cui è possibile accedere tramite un SDK.

Con un pool di identità, i tuoi utenti possono ottenere AWS credenziali temporanee per accedere a AWS servizi come Amazon S3 e DynamoDB. I pool di identità supportano utenti ospiti anonimi e la federazione tramite terze parti. IdPs

Argomenti

- Autenticazione con un bacino d'utenza

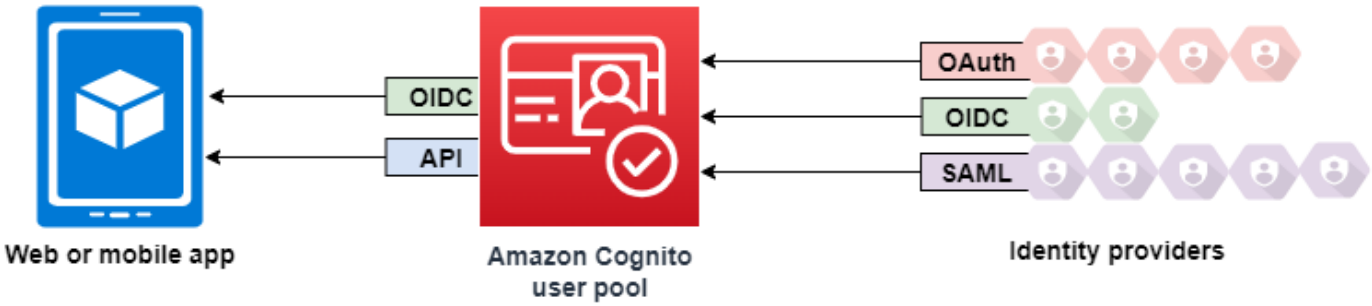
- Accedi alle risorse di back-end con i token del pool di utenti
- Accesso alle risorse con API Gateway e Lambda tramite un bacino d'utenza
- Accesso AWS servizi con un pool di utenti e un pool di identità
- Effettua l'autenticazione con una terza parte e accedi AWS servizi con un pool di identità
- Accesso AWS AppSync risorse con Amazon Cognito
- Autorizza le applicazioni da macchina a macchina (M2M)
- Step-up authentication

Autenticazione con un bacino d'utenza

Puoi consentire ai tuoi utenti di autenticarsi con un bacino d'utenza. Gli utenti della tua app possono accedere direttamente tramite un pool di utenti oppure possono federarsi tramite un provider di identità (IdP) di terze parti. Il pool di utenti gestisce il sovraccarico di gestione dei token restituiti dall'accesso social tramite Facebook, Google, Amazon e Apple e da OpenID Connect (OIDC) e SAML. IdPs

Dopo un'autenticazione riuscita, la tua app web o mobile riceve i token del pool di utenti da Amazon Cognito. Puoi utilizzare questi token per recuperare AWS le credenziali che consentono alla tua app di accedere ad altri AWS servizi oppure puoi scegliere di usarli per controllare l'accesso alle tue risorse lato server o ad Amazon API Gateway.

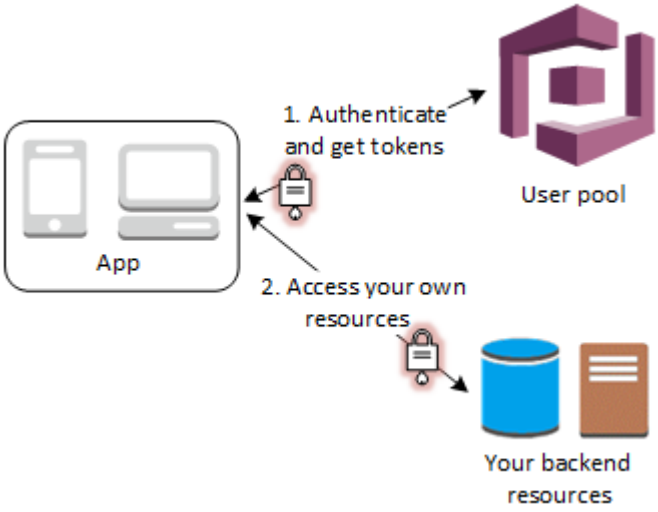
Per ulteriori informazioni, consulta Un esempio di sessione di autenticazione e Comprensione dei token web JSON (JWT) del pool di utenti.



Accedi alle risorse di back-end con i token del pool di utenti

Dopo aver effettuato correttamente l'accesso al pool di utenti, la tua app Web o mobile riceve i token del pool di utenti da Amazon Cognito. È possibile utilizzare questi token per controllare l'accesso

alle risorse lato server. Puoi anche creare gruppi di bacini d'utenza per gestire le autorizzazioni o rappresentare diversi tipi di utenti. Per ulteriori informazioni sull'utilizzo dei gruppi per controllare l'accesso alle risorse, consulta Aggiunta di gruppi a un bacino d'utenza.



Dopo aver configurato un dominio per il pool di utenti, Amazon Cognito effettua il provisioning di un'interfaccia utente Web ospitata che consente di aggiungere pagine di registrazione e di accesso all'app. Tramite questa base OAuth 2.0 è possibile creare il server di risorse per consentire agli utenti di accedere alle risorse protette. Per ulteriori informazioni, consulta Ambiti, M2M e server di risorse.

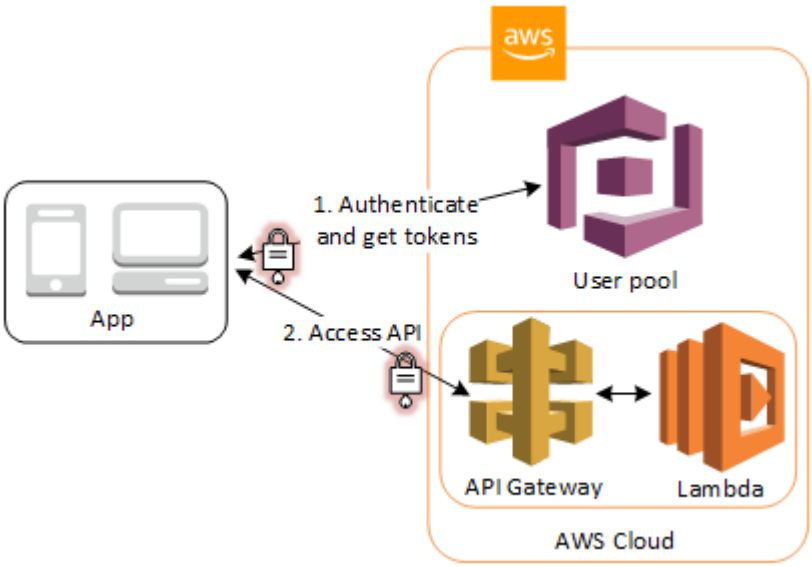
Per ulteriori informazioni sull'autenticazione dei pool di utenti, consulta Un esempio di sessione di autenticazione e Comprensione dei token web JSON (JWT) del pool di utenti.

Accesso alle risorse con API Gateway e Lambda tramite un bacino d'utenza

È possibile consentire agli utenti di accedere all'API tramite API Gateway. API Gateway convalida i token da una corretta autenticazione del bacino d'utenza e li utilizza per concedere agli utenti l'accesso a risorse, tra cui le funzioni Lambda o la tua API.

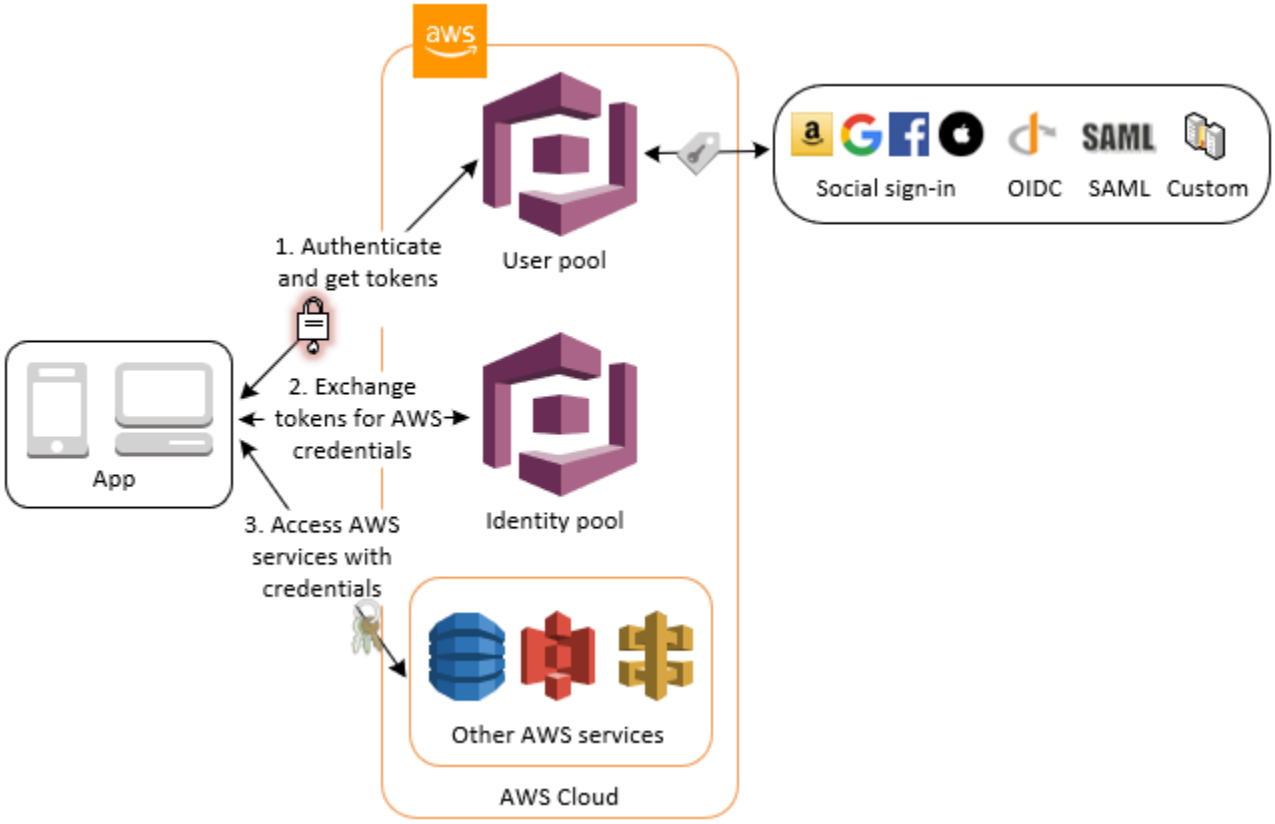
È possibile utilizzare i gruppi in un bacino d'utenza per controllare le autorizzazioni con API Gateway mappando l'appartenenza al gruppo ai ruoli IAM. I gruppi di cui un utente è membro sono inclusi nel token ID fornito da un bacino d'utenza quando l'utente dell'app effettua l'accesso. Per ulteriori informazioni sui gruppi di pool di utenti, consulta. Aggiunta di gruppi a un bacino d'utenza

È possibile inviare i token del bacino d'utenza con una richiesta ad API Gateway di verifica mediante una funzione Lambda di autorizzazione di Amazon Cognito. Per ulteriori informazioni su API Gateway, consulta Utilizzo di API Gateway con bacini d'utenza di Amazon Cognito.



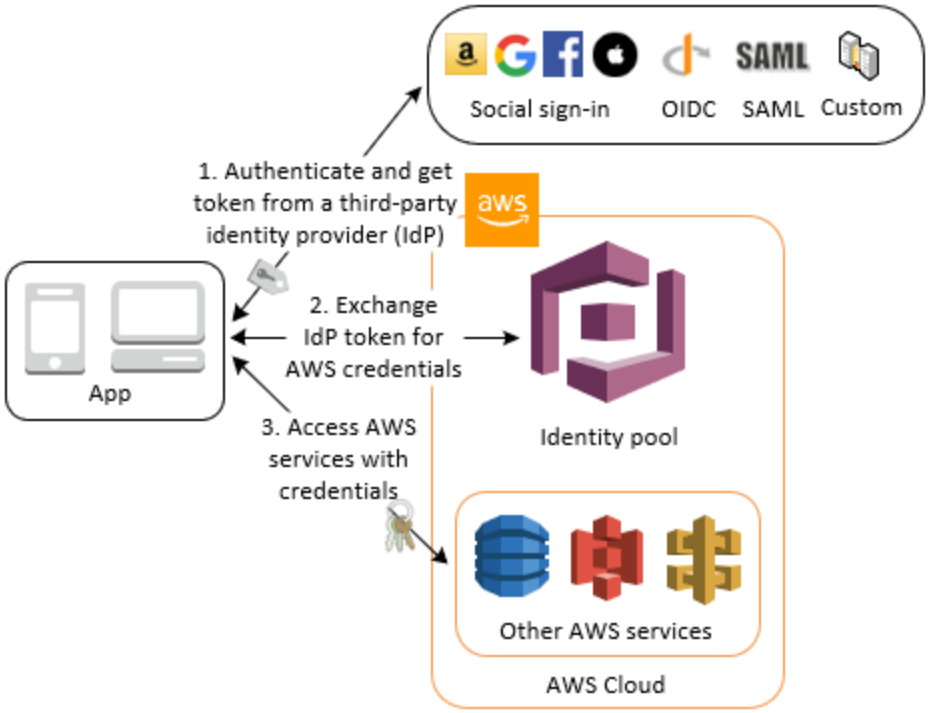
Accesso AWS servizi con un pool di utenti e un pool di identità

Dopo una corretta autenticazione del pool di utenti, l'app riceve i token del pool di utenti da Amazon Cognito. Puoi scambiarli con l'accesso temporaneo ad altri AWS servizi con un pool di identità. Per ulteriori informazioni, consulta Accesso Servizi AWS utilizzo di un pool di identità dopo l'accesso e Guida introduttiva ai pool di identità di Amazon Cognito.



Effettua l'autenticazione con una terza parte e accedi AWS servizi con un pool di identità

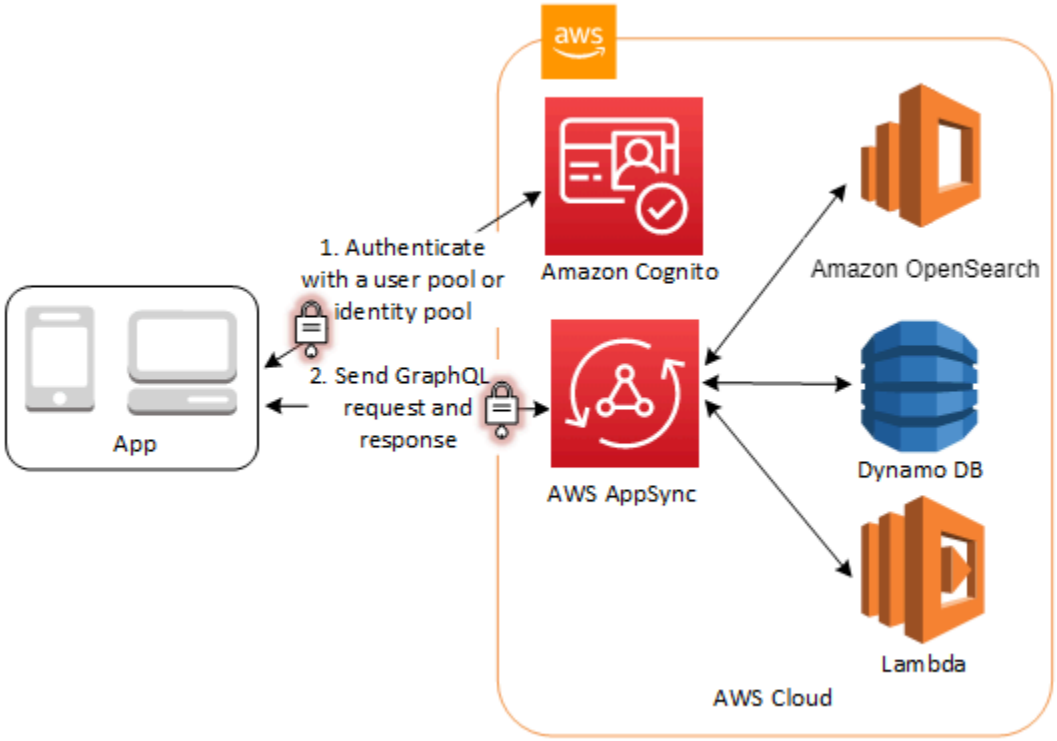
È possibile consentire agli utenti di accedere ai AWS servizi tramite un pool di identità. Un pool di identità richiede un token IdP da un utente autenticato da un provider di identità di terze parti (o niente se è un utente guest anonimo). In cambio, il pool di identità concede AWS credenziali temporanee che è possibile utilizzare per accedere ad altri AWS servizi. Per ulteriori informazioni, consulta Guida introduttiva ai pool di identità di Amazon Cognito.



Accesso AWS AppSync risorse con Amazon Cognito

Puoi concedere ai tuoi utenti l'accesso alle AWS AppSync risorse con i token di un'autenticazione riuscita del pool di utenti di Amazon Cognito. Per ulteriori informazioni, consultare Autorizzazione AMAZON_COGNITO_USER_POOLS nella Guida per sviluppatori di AWS AppSync .

Puoi anche firmare le richieste all'API AWS AppSync GraphQL con le credenziali IAM che ricevi da un pool di identità. Consultare Autorizzazione AWS_IAM.



Autorizza le applicazioni da macchina a macchina (M2M)

Con l'autorizzazione da macchina a macchina (M2M), un'applicazione, un servizio o un dispositivo possono autenticarsi e ottenere l'accesso autorizzato alle API, senza alcun utente connesso. Amazon Cognito autorizza un'applicazione anziché un utente ed emette solo un token di accesso, nessun token ID e nessun token di aggiornamento. Questo è il modello per carichi di lavoro come un'applicazione che raccoglie in modo autonomo i dati da una flotta di dispositivi Internet of Things (IoT), in cui ogni dispositivo si connette all'API e invia dati senza alcun intervento umano.

Per abilitare l'autorizzazione M2M, crea un client di app con un client secret e abilitalo per la concessione. `client_credentials` L'identità della macchina richiede i token dal server di autorizzazione e li presenta al server di risorse. Per ulteriori informazioni, consulta Machine-to-machine Autorizzazione (M2M) e Endpoint Token.

Step-up authentication

Prendi in considerazione un'applicazione bancaria che consenta agli utenti di accedere con nome utente e password. Per effettuare un bonifico bancario, l'applicazione richiede l'autenticazione a più fattori (MFA). Quando un utente tenta un trasferimento, l'applicazione esamina la `acr` dichiarazione

nei token di accesso e ID dell'utente, determina che il livello di autenticazione è insufficiente e reindirizza l'utente ad Amazon Cognito per completare la MFA. Amazon Cognito emette quindi un nuovo token con un livello di autenticazione elevato e l'applicazione consente il trasferimento.

Prima di reindirizzare un utente ad Amazon Cognito per l'autenticazione avanzata, valuta i token già presenti nell'applicazione. Confronta la `auth_time` dichiarazione con l'età massima consentita dall'operazione e confronta il `acr` livello del token con il livello richiesto dall'operazione. Reindirizza l'utente ad Amazon Cognito per completare il flusso di autenticazione solo quando il livello di autenticazione o il livello di aggiornamento richiesti non sono già stati raggiunti.

Per ulteriori informazioni sui livelli ACR, i valori AMR e altri concetti di autenticazione avanzata, consulta. the section called “Livelli di autenticazione con attestazioni ACR e AMR” Per i meccanismi di richiesta dell'autenticazione step-up, consulta For the `USER_AUTH` flow e Step-up autenticazione con il flusso USER_AUTH for managed login. Endpoint Authorize Endpoint Login

Configurazione della superficie di accesso

Prima che gli utenti possano autenticarsi, devi configurare la superficie tramite cui accedono. Per quanto riguarda l'accesso gestito, ciò significa un dominio con pool di utenti per gestire le pagine e un client dell'app che indirizza a esse, oltre al marchio che applichi. Questo capitolo tratta come:

- Accesso gestito: le pagine di accesso ospitate, il dominio del pool di utenti che le serve e l'editor di branding.
- Migrazione dalla classica interfaccia utente ospitata: cosa cambia quando si sposta un pool di utenti esistente all'accesso gestito.

Per decidere innanzitutto tra accesso gestito, API dei pool di utenti e federazione, consulta. the section called “Scegli un approccio” Per i flussi di autenticazione stessi, vedithe section called “Autenticazione”.

Accesso gestito dal pool di utenti

Puoi scegliere un dominio web per ospitare i servizi per il tuo pool di utenti. Un pool di utenti di Amazon Cognito acquisisce le seguenti funzioni quando aggiungi un dominio, denominato collettivamente accesso gestito.

- Un server di autorizzazione che funge da provider di identità (IdP) per le applicazioni che funzionano con OAuth 2.0 e OpenID Connect (OIDC). Il server di autorizzazione indirizza le richieste, emette e gestisce i token web JSON (JWT) e fornisce informazioni sugli attributi utente.
- Un'interfaccia utente (UI) pronta all'uso per operazioni di autenticazione come accesso, disconnessione e gestione delle password. Le pagine di accesso gestite fungono da front-end web per i servizi di autenticazione.
- Un service provider (SP) o relying party (RP) per SAML 2.0, OIDC IdPs, Facebook IdPs, Login with Amazon, Accedi con Apple e Google.

L'accesso gestito ha un'alternativa: la classica interfaccia utente ospitata. Scegli l'uno o l'altro per un dominio con pool di utenti: l'accesso gestito e la classica interfaccia utente ospitata non vengono utilizzati insieme. L'interfaccia utente ospitata classica è una versione di prima generazione dei servizi di accesso gestito. I servizi IdP e RP dell'interfaccia utente ospitata in genere hanno le stesse caratteristiche dell'accesso gestito, ma le pagine di accesso hanno un design più semplice e meno

funzionalità. Ad esempio, l'accesso tramite passkey non è disponibile nella classica interfaccia utente ospitata. Nel piano di funzionalità Lite, l'interfaccia utente ospitata classica è l'unica opzione per i servizi di dominio in pool di utenti.

Le pagine di accesso gestito sono una raccolta di interfacce Web per le attività di registrazione, accesso, autenticazione a più fattori e reimpostazione della password nel pool di utenti. Inoltre, collegano gli utenti a uno o più provider di identità di terze parti (IdPs) quando si desidera offrire agli utenti una scelta di opzioni di accesso. L'app può richiamare le pagine di accesso gestite nei browser degli utenti quando si desidera autenticare e autorizzare gli utenti.

Puoi adattare l'esperienza utente con accesso gestito al tuo marchio con loghi, sfondi e stili personalizzati. Hai due opzioni per il branding da applicare all'interfaccia utente di accesso gestito: l'editor di branding per l'accesso gestito e il branding dell'interfaccia utente ospitata (classica) per l'interfaccia utente ospitata.

Editor di branding

> Un'esperienza utente aggiornata con le opzioni di autenticazione più aggiornate e un editor visivo nella console Amazon Cognito.

Branding dell'interfaccia utente ospitata (classica)

> Un'esperienza utente familiare per chi ha già utilizzato i pool di utenti di Amazon Cognito. Il branding per l'interfaccia utente ospitata è un sistema basato su file. Per applicare il branding alle pagine dell'interfaccia utente ospitate, carichi un file di immagine del logo e un file che imposta i valori per diverse opzioni di stile CSS predeterminate.

L'editor di branding non è disponibile in tutti i piani di funzionalità per i pool di utenti. Per ulteriori informazioni, consulta Piani di funzionalità del pool di utenti.

Per ulteriori informazioni sulla creazione di richieste per l'accesso gestito e i servizi di interfaccia utente ospitati, consulta. Endpoint del pool di utenti e riferimento per l'accesso gestito

L'accesso gestito supporta l'autenticazione avanzata con i `max_age` parametri `acr_values` e sugli endpoint di autorizzazione e accesso. Per ulteriori informazioni, consulta l'endpoint di autorizzazione e l'endpoint di accesso.

> **Note**
>
> L'accesso gestito di Amazon Cognito non supporta l'autenticazione personalizzata con trigger Lambda della sfida di autenticazione personalizzata.

Argomenti

- Managed login localization
- Documenti relativi ai termini di utilizzo
- Configurazione dell'accesso gestito con la console Amazon Cognito
- Configurazione dell'accesso gestito con AWS Amplify
- Visualizzazione della pagina di accesso
- Personalizzazione delle pagine di autenticazione
- Cose da sapere sull'accesso gestito e sull'interfaccia utente ospitata
- Configurazione di un dominio di bacino d'utenza
- Applica il branding alle pagine di accesso gestite

Managed login localization

L'accesso gestito utilizza per impostazione predefinita la lingua inglese nelle pagine interattive con l'utente. È possibile visualizzare le pagine di accesso gestito localizzate per la lingua desiderata. Le lingue disponibili sono quelle disponibili in. Console di gestione AWS Nel link che distribuite agli utenti, aggiungete un parametro di `lang` query, come illustrato nell'esempio seguente.

```
https://<your domain>/oauth2/authorize?lang=es&response_type=code&client_id=<your app
 client id>&redirect_uri=<your relying-party url>
```

Amazon Cognito imposta un cookie nel browser degli utenti con le loro preferenze di lingua dopo la richiesta iniziale con un parametro. `lang` Dopo l'impostazione del cookie, la selezione della lingua persiste senza visualizzare o richiedere l'inclusione del parametro nelle richieste. Ad esempio, dopo che un utente effettua una richiesta di accesso con un `lang=de` parametro, le pagine di accesso gestite vengono visualizzate in tedesco finché non cancella i cookie o non effettua una nuova richiesta con un nuovo parametro di localizzazione, ad esempio. `lang=en`

La localizzazione è disponibile solo per l'accesso gestito. Devi avere sottoscritto il piano di funzionalità Essentials o Plus e aver assegnato il tuo dominio per utilizzare il marchio di accesso gestito.

La selezione effettuata dall'utente nell'accesso gestito non è disponibile per i trigger personalizzati dei mittenti di e-mail o SMS. Quando si implementano questi trigger, è necessario utilizzare altri meccanismi per determinare la lingua preferita dell'utente. Nei flussi di accesso, l'`locale`attributo potrebbe indicare la lingua preferita dell'utente in base alla posizione. Nei flussi di registrazione, la regione o l'ID client dell'app del tuo pool di utenti potrebbero indicare una preferenza di lingua.

Sono disponibili le seguenti lingue.

Lingue di accesso gestite

Lingua	Codice
Tedesco	de
Inglese	en
Spagnolo	es
Francese	fr
Bahasa Indonesia	id
Olandese	nl
Italiano	it
Giapponese	ja
Coreano	ko
Portoghese (Brasile)	pt-BR
Cinese (semplificato)	zh-CN
Cinese (tradizionale)	zh-TW

Documenti relativi ai termini di utilizzo

Puoi configurare le tue pagine di accesso gestito in modo da visualizzare i link ai documenti delle Condizioni d'uso e dell'Informativa sulla privacy quando gli utenti si iscrivono. Quando configuri entrambi i documenti terminologici nel client dell'app, gli utenti visualizzano il seguente testo durante la registrazione: Registrandoti, accetti le nostre Condizioni d'uso e la nostra Informativa sulla privacy. Le frasi Termini di utilizzo e Informativa sulla privacy vengono visualizzate nella tua pagina di accesso gestita, con collegamento ipertestuale ai tuoi documenti.

I documenti Terms supportano URL specifici per lingua che si allineano alla localizzazione degli accessi gestiti. Quando gli utenti selezionano una lingua con il parametro di `lang` query, Amazon Cognito visualizza i link ai tuoi documenti terminologici in quella lingua. Se non hai configurato un URL per una lingua specifica, Amazon Cognito utilizza l'URL predefinito che hai configurato per il client dell'app.

Per configurare i termini e i documenti per il client dell'app, accedi al menu Accesso gestito nel tuo pool di utenti. In Documenti relativi ai termini, scegli Crea documento con i termini.

Amazon Cognito console

Per creare un documento relativo ai termini

1. Accedi al tuo pool di utenti e scegli il menu Accesso gestito. Individua i documenti Terms.
2. Scegli Crea documento con i termini.
3. Seleziona il client dell'app a cui desideri assegnare il documento relativo ai termini.
4. Inserisci il nome di un termine. Questo identifica il documento nella console.
5. In Collegamenti, scegli una lingua e inserisci l'URL in cui archiviare il documento terminologico in quella lingua.
6. Per aggiungere URL per altre lingue, scegli Aggiungi un'altra.
7. Scegli Crea.

Amazon Cognito user pools API

Di seguito è riportato un esempio di corpo della CreateTermsrichiesta. `1example23456789Fa` sì che la pagina di registrazione per il client dell'app mostri i collegamenti per una French-language e una versione Portuguese-language (Brasile) dell'informativa sulla privacy quando l'accesso

gestito è localizzato in quella lingua. È necessaria una richiesta separata per impostare gli URL da utilizzare `terms-of-use` prima che l'accesso gestito visualizzi i link nella pagina di registrazione.

```
{
   "ClientId": "1example23456789",
   "Enforcement": "NONE",
   "Links": {
      "cognito:default" : "https://example.com/privacy/",
      "cognito:french" : "https://example.com/fr/privacy/",
      "cognito:portuguese-brazil" : "https://example.com/pt/privacy/"
   },
   "TermsName": "privacy-policy",
   "TermsSource": "LINK",
   "UserPoolId": "us-east-1_EXAMPLE"
}
```

> **Note**
>
> È necessario creare sia le condizioni d'uso che un documento di informativa sulla privacy per il client dell'app prima che Amazon Cognito visualizzi i documenti relativi ai termini nelle pagine di accesso gestite.

Configurazione dell'accesso gestito con la console Amazon Cognito

Il primo requisito per l'accesso gestito e l'interfaccia utente ospitata è un dominio con pool di utenti. Nella console dei pool di utenti, vai alla scheda Dominio del tuo pool di utenti e aggiungi un dominio Cognito o un dominio personalizzato. Puoi anche scegliere un dominio durante il processo di creazione di un nuovo pool di utenti. Per ulteriori informazioni, consulta Configurazione di un dominio di bacino d'utenza. Quando un dominio è attivo nel tuo pool di utenti, tutti i client dell'app forniscono pagine di autenticazione pubbliche su quel dominio.

Quando crei o modifichi un dominio con pool di utenti, imposti la versione di branding per il tuo dominio. Questa versione di branding è una scelta tra accesso gestito o interfaccia utente ospitata (classica). La versione di branding scelta si applica a tutti i client dell'app che utilizzano i servizi di accesso del tuo dominio.

Il passaggio successivo consiste nel creare un client per l'app dalla scheda App client del pool di utenti. Durante il processo di creazione di un client per l'app, Amazon Cognito ti chiede informazioni

sulla tua applicazione, quindi ti chiede di selezionare un URL di ritorno. L'URL di ritorno viene anche chiamato URL del relying party (RP), URI di reindirizzamento e URL di callback. Questo è l'URL da cui viene eseguita l'applicazione, ad esempio o. `https://www.example.com` `myapp://example`

La versione di branding che imposti sul dominio determina lo stile delle tue pagine di accesso: scegli Accesso gestito per progettare pagine nell'editor di branding o Hosted UI (classica) per applicare il branding dell'interfaccia utente ospitata basato su file. Per ulteriori informazioni sull'editor di branding e sui piani di funzionalità che lo supportano, consulta. Piani di funzionalità del pool di utenti

Dopo aver configurato un dominio e un client per l'app e impostato una versione di branding nel tuo pool di utenti, le pagine di accesso gestite diventano disponibili su Internet.

Per configurare l'accesso gestito nella console Amazon Cognito

1. Aggiungi un dominio del pool di utenti. Nella console Amazon Cognito, scegli il tuo pool di utenti, apri la scheda Dominio e aggiungi un dominio Cognito (dominio con prefisso) o un dominio personalizzato. Per ulteriori informazioni, consulta Configurazione di un dominio di bacino d'utenza.
2. Imposta la versione di branding per il dominio. Quando crei o modifichi il dominio, scegli Accesso gestito o Interfaccia utente ospitata (classica). Questa scelta imposta lo stile delle pagine di accesso e si applica a tutti i client di app che utilizzano il dominio.
3. Crea un client per l'app. Nella scheda App client, create un client per l'app e impostate il relativo URL di ritorno (chiamato anche URL di callback o URI di reindirizzamento) sull'URL da cui viene eseguita l'applicazione.
4. (Solo accesso gestito) Personalizza lo stile. Apri l'editor di branding per il client dell'app per progettare le tue pagine di accesso. L'editor di branding è disponibile in un sottoinsieme di piani di funzionalità; per ulteriori informazioni, consulta. Piani di funzionalità del pool di utenti
5. Visualizza la tua pagina di accesso. Nella scheda Pagine di accesso per il client dell'app, scegli Visualizza pagine di accesso per aprire le pagine di accesso gestite.

Configurazione dell'accesso gestito con AWS Amplify

Se lo utilizzi AWS Amplify per aggiungere l'autenticazione alla tua app web o mobile, puoi configurare le pagine di accesso gestite nell'interfaccia a riga di comando (CLI) di Amplify e le librerie nel framework Amplify. Per aggiungere l'autenticazione alla tua app, aggiungi la categoria al `Auth` tuo progetto. Quindi, nella tua applicazione, autentica gli utenti del pool di utenti con le librerie client Amplify.

Puoi richiamare pagine di accesso gestite per l'autenticazione oppure puoi federare gli utenti tramite un endpoint di autorizzazione che reindirizza a un IdP. Dopo che un utente si è autenticato con successo con il provider, Amplify crea un nuovo utente nel tuo pool di utenti e passa i token dell'utente alla tua app.

Gli esempi seguenti mostrano come configurare l' AWS Amplify accesso gestito con i social provider nella tua app.

- React
- Swift
- Flutter
- Android

Visualizzazione della pagina di accesso

Nella console Amazon Cognito, scegli il pulsante Visualizza pagine di accesso nella scheda Pagine di accesso per il client dell'app nel menu App client. Questo pulsante ti porta a una pagina di accesso nel dominio del tuo pool di utenti con i seguenti parametri di base.

- ID del client dell'app
- Richiesta di concessione del codice di autorizzazione
- Richiesta per tutti gli ambiti attivati per il client dell'app corrente
- Primo URL di callback nell'elenco per il client dell'app corrente

Il pulsante Visualizza pagina di accesso è utile quando si desidera testare le funzioni di base delle pagine di accesso gestite. Le tue pagine di accesso corrispondono alla versione di branding che hai assegnato al dominio del tuo pool di utenti. Puoi personalizzare l'URL di accesso con parametri aggiuntivi e modificati. Nella maggior parte dei casi, i parametri generati automaticamente del link Visualizza la pagina di accesso non soddisfano completamente le esigenze dell'app. In questi casi, devi personalizzare l'URL richiamato dall'app in fase di accesso degli utenti. Per ulteriori informazioni sui parametri di accesso e sui relativi valori, consulta Endpoint del pool di utenti e riferimento per l'accesso gestito.

La pagina web di accesso utilizza il seguente formato URL. In questo esempio viene richiesta una concessione del codice di autorizzazione con il parametro `response_type=code`.

```
https://<your domain>/oauth2/authorize?response_type=code&client_id=<your app client
 id>&redirect_uri=<your relying-party url>
```

È possibile cercare la stringa del dominio del pool di utenti dal menu Dominio del pool di utenti. Nel menu App client, puoi identificare gli ID dei client dell'app, i relativi URL di callback, gli ambiti consentiti e altre configurazioni.

Quando si accede all'endpoint `/oauth2/authorize` con i parametri personalizzati, Amazon Cognito ti reindirizza all'endpoint `/oauth2/login` o, se è presente un parametro `identity_provider` o `idp_identifier`, ti reindirizza in modalità invisibile all'utente alla pagina di accesso del gestore dell'identità digitale (IdP).

Esempio di richiesta di concessione implicita

Puoi visualizzare la tua pagina web di accesso con il seguente URL per il codice implicito grant where. `response_type=token` Dopo aver eseguito correttamente l'accesso, Amazon Cognito restituisce i token del bacino d'utenza alla barra degli indirizzi del browser Web.

```
            https://mydomain.auth.us-east-1.amazoncognito.com/authorize?
response_type=token&client_id=1example23456789&redirect_uri=https://
mydomain.example.com
```

I token di identità e accesso vengono visualizzati come parametri aggiunti all'URL di reindirizzamento.

Di seguito è riportato un esempio di risposta da una richiesta di concessione implicita.

```
            https://auth.example.com/
#id_token=eyJraaBcDeF1234567890&access_token=eyJraGhIjKlM1112131415&expires_in=3600&token_type=
```

Personalizzazione delle pagine di autenticazione

In passato, Amazon Cognito ospitava solo pagine di accesso con la classica interfaccia utente ospitata, un design semplice che conferiva un aspetto universale alle pagine Web di autenticazione. Puoi personalizzare i pool di utenti di Amazon Cognito con un'immagine del logo e modificare alcuni stili con un file che specifichi alcuni valori di stile CSS. Successivamente, Amazon Cognito ha

introdotto l'accesso gestito, un servizio di autenticazione ospitato aggiornato. L'accesso gestito ha un aspetto aggiornato con l'editor di branding. L'editor di branding è un editor visivo senza codice e offre una suite di opzioni più ampia rispetto all'esperienza di personalizzazione dell'interfaccia utente ospitata. L'accesso gestito ha inoltre introdotto immagini di sfondo personalizzate e un tema in modalità scura.

Puoi passare dall'accesso gestito all'esperienza di branding dell'interfaccia utente ospitata nei pool di utenti. Per ulteriori informazioni sulla personalizzazione delle pagine di accesso gestito, consulta. Applica il branding alle pagine di accesso gestite

Cose da sapere sull'accesso gestito e sull'interfaccia utente ospitata

Il cookie di accesso gestito della durata di un'ora e della sessione dell'interfaccia utente ospitata

Quando un utente accede con le tue pagine di accesso o con un provider terzo, Amazon Cognito imposta un cookie nel suo browser. Con questo cookie, gli utenti possono accedere nuovamente con lo stesso metodo di autenticazione per un'ora. Quando accedono con il cookie del browser, ottengono nuovi token che durano la durata specificata nella configurazione del client dell'app. Le modifiche agli attributi utente o ai fattori di autenticazione non hanno alcun effetto sulla loro capacità di accedere nuovamente con il cookie del browser.

L'autenticazione con il cookie di sessione non reimposta la durata del cookie a un'ora aggiuntiva. Gli utenti devono effettuare nuovamente l'accesso se tentano di accedere alle tue pagine di accesso più di un'ora dopo l'ultima autenticazione interattiva riuscita.

Conferma degli account utente e verifica degli attributi utente

Per gli utenti locali del pool di utenti, l'accesso gestito e l'interfaccia utente ospitata funzionano meglio quando configuri il pool di utenti per consentire a Cognito di inviare automaticamente messaggi per la verifica e la conferma. Quando abiliti questa impostazione, Amazon Cognito invia un messaggio con un codice di conferma agli utenti che effettuano la registrazione. Quando invece confermi gli utenti come amministratori del pool di utenti, le pagine di accesso visualizzano un messaggio di errore dopo la registrazione. In questo stato, Amazon Cognito ha creato il nuovo utente, ma non è stato in grado di inviare un messaggio di verifica. Puoi comunque confermare gli utenti come un amministratore, ma potrebbero contattare il supporto tecnico dopo che hanno rilevato un errore. Per ulteriori informazioni sulla conferma amministrativa, consulta Permettere agli utenti di registrarsi ma confermarli come un amministratore del pool di utenti.

Ambito operativo del login gestito

L'accesso gestito e la classica interfaccia utente ospitata supportano la registrazione, l'accesso e la gestione delle password. Ciò include il completamento dell'accesso con l'autenticazione a più fattori (MFA) e la registrazione degli autenticatori WebAuthn. L'accesso gestito non supporta la gestione dei profili self-service degli utenti, come la modifica degli attributi e l'impostazione delle preferenze MFA. È necessario implementare la gestione dei profili nel codice dell'applicazione. Inoltre, l'accesso gestito non offre la capacità di confermare le modifiche agli attributi quando aggiorni gli indirizzi e-mail e i numeri di telefono in qualità di amministratore con il funzionamento dell'AdminUpdateUserAttributesAPI.

Visualizzazione delle modifiche alla configurazione

Se apporti modifiche di stile alle pagine e queste non vengono visualizzate immediatamente, attendi qualche minuto, quindi aggiorna la pagina.

Decodifica dei token del pool di utenti

I token dei pool di utenti di Amazon Cognito vengono firmati utilizzando un algoritmo RS256. È possibile decodificare e verificare i token del pool di utenti utilizzando. AWS Lambda Vedi Decodifica e verifica dei token Amazon Cognito JWT su. GitHub

Versione TLS

L'accesso gestito e le pagine dell'interfaccia utente ospitate richiedono la crittografia in transito. I domini del pool di utenti forniti da Amazon Cognito richiedono che i browser degli utenti negoziino una versione TLS minima di 1.2. I domini personalizzati supportano le connessioni tramite browser con la versione TLS 1.2. L'interfaccia utente ospitata (classica) non richiede TLS 1.2 per i domini personalizzati, ma il nuovo accesso gestito richiede la versione TLS 1.2 sia per i domini personalizzati che per quelli con prefisso. Poiché Amazon Cognito gestisce la configurazione dei servizi del tuo dominio, non puoi modificare i requisiti TLS del dominio del tuo pool di utenti.

Policy CORS

Né l'accesso gestito né l'interfaccia utente ospitata supportano politiche di origine CORS (Cross-Origin Resource Sharing) personalizzate. Una policy CORS impedirebbe agli utenti di passare i parametri di autenticazione nelle loro richieste. Implementate invece una policy CORS nel front-end dell'applicazione. Amazon Cognito restituisce un'intestazione di `Access-Control-Allow-Origin: *` risposta alle richieste ai seguenti endpoint.

1. Endpoint Token

2. Endpoint Revoke
3. Endpoint UserInfo

Cookie

L'accesso gestito e l'interfaccia utente ospitata impostano i cookie nei browser degli utenti. I cookie sono conformi ai requisiti di alcuni browser secondo cui i siti non impostano cookie di terze parti. Sono limitati solo agli endpoint del pool di utenti e includono quanto segue:

- Un `XSRF-TOKEN` cookie per ogni richiesta.
- Un `csrf-state` cookie per la coerenza della sessione quando un utente viene reindirizzato.
- Un `csrf-state-legacy` cookie per la coerenza della sessione, letto da Amazon Cognito come alternativa quando il browser non supporta l'attributo. `SameSite`
- Un cookie `cognito` di sessione che conserva i tentativi di accesso riusciti per un'ora.
- Un `lang` cookie che mantiene la localizzazione della lingua scelta dall'utente nell'accesso gestito.
- Un `page-data` cookie che memorizza i dati richiesti mentre un utente naviga tra le pagine di accesso gestite.

In iOS, puoi bloccare tutti i cookie. Questa impostazione non è compatibile con l'accesso gestito o l'interfaccia utente ospitata. Per lavorare con utenti che potrebbero abilitare questa impostazione, crea l'autenticazione del pool di utenti in un'app iOS nativa con un AWS SDK. In questo scenario, puoi creare un archivio di sessione personalizzato che non sia basato sui cookie.

Effetti della modifica della versione di accesso gestito

Considerate i seguenti effetti dell'aggiunta di domini e dell'impostazione della versione di accesso gestito.

- Quando aggiungi un dominio con prefisso, con accesso gestito o branding dell'interfaccia utente ospitata (classica), possono essere necessari fino a 60 secondi prima che le pagine di accesso siano disponibili.
- Quando aggiungi un dominio personalizzato, con login gestito o branding dell'interfaccia utente ospitata (classica), possono essere necessari fino a cinque minuti prima che le pagine di accesso siano disponibili.
- Quando modifichi la versione del branding per il tuo dominio, possono essere necessari fino a quattro minuti prima che le pagine di accesso siano disponibili nella nuova versione di branding.

- Quando passi dall'accesso gestito al branding dell'interfaccia utente ospitata (classica), Amazon Cognito non mantiene le sessioni utente. Devono accedere nuovamente con la nuova interfaccia.

Stile predefinito

Quando crei un client per app in Console di gestione AWS, Amazon Cognito assegna automaticamente uno stile di branding al client dell'app. Quando crei in modo programmatico un client di app con l'CreateUserPoolClientoperazione, non viene creato uno stile di branding. L'accesso gestito non è disponibile per un client di app creato con un AWS SDK finché non ne crei uno con una richiesta. CreateManagedLoginBranding

Scade il timeout della richiesta di autenticazione dell'accesso gestito

Amazon Cognito annulla le richieste di autenticazione che non vengono completate entro cinque minuti e reindirizza l'utente all'accesso gestito. Viene visualizzato il messaggio di errore `Something went wrong` nella pagina.

Configurazione di un dominio di bacino d'utenza

La configurazione di un dominio è una parte facoltativa della configurazione di un pool di utenti. Un dominio con pool di utenti ospita funzionalità per l'autenticazione degli utenti, la federazione con provider di terze parti e i flussi OpenID Connect (OIDC). Dispone di accesso gestito, un'interfaccia predefinita per operazioni chiave come la registrazione, l'accesso e il recupero della password. Ospita anche gli endpoint OpenID Connect (OIDC) standard come authorize, UserInfo e token, per l'autorizzazione machine-to-machine (M2M) e altri flussi di autenticazione e autorizzazione OIDC e OAuth 2.0.

L'autorizzazione M2M è disponibile anche senza dominio. Puoi richiedere i token di accesso alle credenziali del client con l'operazione GetClientTokenAPI, che utilizza l'API Amazon Cognito anziché l'endpoint del token.

Gli utenti si autenticano con pagine di accesso gestite nel dominio associato al tuo pool di utenti. Hai due opzioni per configurare questo dominio: puoi utilizzare il dominio con prefisso Amazon Cognito predefinito o configurare un dominio personalizzato di tua proprietà.

L'opzione di dominio personalizzato offre più opzioni di flessibilità, sicurezza e controllo. Ad esempio, un dominio familiare di proprietà dell'organizzazione può incoraggiare la fiducia degli utenti e rendere più intuitiva la procedura di accesso. Tuttavia, l'approccio personalizzato al dominio richiede alcuni costi aggiuntivi, come la gestione del certificato SSL e della configurazione DNS.

Gli endpoint di rilevamento OIDC, `/.well-known/openid-configuration` per gli URL degli endpoint e `/.well-known/jwks.json` per le chiavi di firma dei token, non sono ospitati nel tuo dominio. Per ulteriori informazioni, consulta Endpoint del provider di identità e del relying party.

Capire come configurare e gestire il dominio per il tuo pool di utenti è un passo importante per integrare l'autenticazione nella tua applicazione. Sign-in con l'API dei pool di utenti e un AWS SDK possono essere un'alternativa alla configurazione di un dominio. Il API-based modello fornisce i token direttamente in una risposta API, ma per le implementazioni che utilizzano le funzionalità estese dei pool di utenti come IdP OIDC, è necessario configurare un dominio. Per ulteriori informazioni sui modelli di autenticazione disponibili nei pool di utenti, vedere. Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite

Argomenti

- Cose da sapere sui domini dei pool di utenti
- Utilizzo del dominio con prefisso Amazon Cognito per l'accesso gestito
- Utilizzo del proprio dominio per l'accesso gestito

Cose da sapere sui domini dei pool di utenti

I domini dei pool di utenti sono un punto di servizio per le parti che si affidano a OIDC nelle applicazioni e per gli elementi dell'interfaccia utente. Considerate i seguenti dettagli quando pianificate l'implementazione di un dominio per il vostro pool di utenti.

Termini riservati

Non puoi utilizzare il testo `aws` o `cognito` il nome di un dominio con prefisso Amazon Cognito. `amazon`

Gli endpoint Discovery si trovano su un dominio diverso

Gli endpoint `.well-known/openid-configuration` di rilevamento del pool di utenti `.well-known/jwks.json` non fanno parte del dominio personalizzato o con prefisso del pool di utenti. Il percorso verso questi endpoint è il seguente.

- `https://cognito-idp.`*`Region`*`.amazonaws.com/`*`your user pool ID`*`/.well-known/openid-configuration`
- `https://cognito-idp.`*`Region`*`.amazonaws.com/`*`your user pool ID`*`/.well-known/jwks.json`

Ora effettiva delle modifiche al dominio

Amazon Cognito può impiegare fino a un minuto per avviare o aggiornare la versione di branding di un dominio con prefisso. La propagazione delle modifiche a un dominio personalizzato può richiedere fino a cinque minuti. La propagazione dei nuovi domini personalizzati può richiedere fino a un'ora.

Domini personalizzati e con prefisso contemporaneamente

Puoi configurare un pool di utenti con un dominio personalizzato e un dominio con prefisso di proprietà di. AWS Poiché gli endpoint di rilevamento del pool di utenti sono ospitati in un dominio diverso, servono solo il dominio personalizzato. Ad esempio, `openid-configuration` fornisce un valore singolo per `"authorization_endpoint"` di`"https://auth.example.com/oauth2/authorize"`.

Quando in un pool di utenti sono presenti sia domini personalizzati che domini con prefisso, puoi utilizzare il dominio personalizzato con tutte le funzionalità di un provider OIDC. Il dominio con prefisso in un pool di utenti con questa configurazione non dispone di endpoint con chiave di rilevamento o di firma dei token e deve essere utilizzato di conseguenza.

Domini personalizzati preferiti come ID del relying party per la passkey

Quando si configura l'autenticazione del pool di utenti con passkey, è necessario impostare un ID relying party (RP). Se disponi di un dominio personalizzato e di un dominio con prefisso, puoi impostare l'ID RP solo come dominio personalizzato. Per impostare un dominio con prefisso come ID RP nella console Amazon Cognito, elimina il dominio personalizzato o inserisci il nome di dominio completo (FQDN) del dominio con prefisso come dominio. Third-party

Non utilizzare domini personalizzati a diversi livelli della gerarchia dei domini

Puoi configurare pool di utenti separati per avere domini personalizzati nello stesso dominio di primo livello (TLD), ad esempio auth.example.com e auth2.example.com. Il cookie della sessione di accesso gestito è valido per un dominio personalizzato e per tutti i sottodomini, ad esempio *.auth.example.com. Per questo motivo, nessun utente delle tue applicazioni deve accedere all'accesso gestito per qualsiasi dominio e sottodominio principale. Se i domini personalizzati utilizzano lo stesso TLD, mantienili allo stesso livello di sottodominio.

Supponiamo che tu abbia un pool di utenti con il dominio personalizzato auth.example.com. Quindi crei un altro pool di utenti e assegni il dominio personalizzato uk.auth.example.com. . Un utente accede con auth.example.com. e riceve un cookie che il browser presenta a qualsiasi sito Web nel percorso wildcard *.auth.example.com. Quindi provano ad accedere a uk.auth.example.com. .

Passano un cookie non valido al dominio del pool di utenti e ricevono un errore anziché una richiesta di accesso. Al contrario, un utente con un cookie per *.auth.example.com non ha problemi ad avviare una sessione di accesso su auth2.example.com.

Versione di branding

Quando crei un dominio, imposti una versione di branding. Le tue opzioni sono la nuova esperienza di accesso gestito e la classica esperienza di interfaccia utente ospitata. Questa scelta si applica a tutti i client di app che ospitano servizi nel tuo dominio.

Utilizzo del dominio con prefisso Amazon Cognito per l'accesso gestito

L'esperienza predefinita per l'accesso gestito è ospitata su un dominio proprietario. AWS Questo approccio presenta una bassa barriera all'ingresso: basta scegliere un nome di prefisso ed è attivo, ma non offre le caratteristiche di fiducia di un dominio personalizzato. Non c'è differenza di costo tra l'opzione di dominio Amazon Cognito e l'opzione di dominio personalizzato. L'unica differenza è il dominio dell'indirizzo web a cui indirizzi gli utenti. In caso di reindirizzamenti IdP di terze parti e flussi di credenziali client, il dominio del prefisso ha scarso effetto visibile. Un dominio personalizzato è preferibile nei casi in cui gli utenti accedono con accesso gestito e interagiscono con un dominio di autenticazione che non corrisponde al dominio dell'applicazione.

Il dominio con prefisso ha un prefisso a tua scelta ed è un sottodominio del dominio di autenticazione regionale`amazoncognito.com`. Di seguito è riportato un esempio:

```
https://cognitoexample.auth.ap-south-1.amazoncognito.com
```

Tutti i domini con prefisso seguono questo formato:. *`prefix`* `auth.` *`Regione AWS code`*. `amazoncognito. com`. I pool di utenti di dominio personalizzati possono ospitare l'accesso gestito o le pagine dell'interfaccia utente ospitate su qualsiasi dominio di tua proprietà.

> **Note**
>
> Per aumentare la sicurezza delle tue applicazioni Amazon Cognito, i domini principali degli endpoint del pool di utenti sono registrati nella Public Suffix List (PSL). La PSL aiuta i browser Web degli utenti a comprendere in modo coerente gli endpoint del pool di utenti e i cookie da essi impostati.
> I domini principali del pool di utenti hanno i seguenti formati.

```
auth.Region.amazoncognito.com
auth-fips.Region.amazoncognito.com
```

Il `auth-fips` dominio è un endpoint conforme al Federal Information Processing Standard (FIPS) 140-3. Fornisce la crittografia in transito con moduli crittografici. FIPS-validated Indirizza gli utenti verso questo endpoint quando l'applicazione deve soddisfare i requisiti di conformità FIPS. Gli endpoint FIPS per l'accesso gestito ad Amazon Cognito sono disponibili in un sottoinsieme AWS di regioni negli Stati Uniti e non in tutte le regioni. Per ulteriori informazioni su FIPS e sulle regioni in cui Amazon Cognito fornisce endpoint FIPS, consulta FIPS 140-3.

Per aggiungere un client per l'app e un dominio di pool di utenti con, consulta. Console di gestione AWSCreazione di un client dell'app

Argomenti

- Prerequisiti
- Configurare un prefisso di dominio Amazon Cognito
- Verifica la tua pagina di accesso

Prerequisiti

Prima di iniziare è necessario disporre di quanto segue:

- Un bacino d'utenza con un client dell'App. Per ulteriori informazioni, consulta Crea una nuova applicazione nella console Amazon Cognito.

Configurare un prefisso di dominio Amazon Cognito

Puoi utilizzare l'API o the Console di gestione AWS AWS CLI o per configurare un dominio con pool di utenti.

Amazon Cognito console

Configura un dominio

1. Vai al menu Dominio sotto Branding.

2. Accanto a Dominio, scegli Azioni e seleziona Crea dominio Cognito. Se hai già configurato un dominio con prefisso del pool di utenti, scegli Elimina dominio Cognito prima di creare il tuo nuovo dominio personalizzato.
3. Inserisci un prefisso dominio disponibile da utilizzare con un Dominio Amazon Cognito. Per informazioni sulla configurazione di un dominio personalizzato, consulta. Utilizzo del proprio dominio per l'accesso gestito
4. Scegli una versione di branding. La tua versione di branding si applica a tutte le pagine interattive con l'utente di quel dominio. Il tuo pool di utenti può ospitare l'accesso gestito o il branding dell'interfaccia utente ospitata per tutti i client dell'app.

   > **Note**
   >
   > Puoi avere un dominio personalizzato e un dominio con prefisso, ma Amazon Cognito serve solo `/.well-known/openid-configuration` l'endpoint per il dominio personalizzato.

5. Scegli Create (Crea).

CLI/API

Utilizza i seguenti comandi per creare un prefisso di dominio e assegnarlo al tuo bacino d'utenza.

Per configurare un dominio del bacino d'utenza

- AWS CLI: `aws cognito-idp create-user-pool-domain`

  Esempio: `aws cognito-idp create-user-pool-domain --user-pool-id <user_pool_id> --domain <domain_name> --managed-login-version 2`
- Funzionamento dell'API dei pool di utenti: CreateUserPoolDomain

Come ottenere le informazioni su un dominio

- AWS CLI: `aws cognito-idp describe-user-pool-domain`

  Esempio: `aws cognito-idp describe-user-pool-domain --domain <domain_name>`
- Funzionamento dell'API dei pool di utenti: DescribeUserPoolDomain

Come eliminare un dominio

- AWS CLI: `aws cognito-idp delete-user-pool-domain`

  Esempio: `aws cognito-idp delete-user-pool-domain --domain <domain_name>`
- Funzionamento dell'API dei pool di utenti: DeleteUserPoolDomain

Verifica la tua pagina di accesso

- Verifica che la pagina di accesso sia disponibile dal tuo dominio ospitato di Amazon Cognito.

```
https://<your_domain>/login?
response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callback_url>
```

Il dominio viene visualizzato nella pagina Nome dominio della console Amazon Cognito. L'ID del client app e l'URL di callback sono visualizzati nella pagina App client settings (Impostazioni client dell'app).

Utilizzo del proprio dominio per l'accesso gestito

Dopo aver configurato un client per l'app, puoi configurare il tuo pool di utenti con un dominio personalizzato per i servizi di dominio dell'accesso gestito. Con un dominio personalizzato, gli utenti possono accedere all'applicazione utilizzando il proprio indirizzo Web anziché il dominio con `amazoncognito.com` prefisso predefinito. I domini personalizzati migliorano la fiducia degli utenti nell'applicazione con un nome di dominio familiare, soprattutto quando il dominio principale corrisponde al dominio che ospita l'applicazione. I domini personalizzati possono migliorare la conformità ai requisiti di sicurezza organizzativi.

Un dominio personalizzato presenta alcuni prerequisiti, tra cui un pool di utenti, un client di app e un dominio Web di tua proprietà. I domini personalizzati richiedono anche un certificato SSL per il dominio personalizzato, gestito con AWS Certificate Manager (ACM) negli Stati Uniti orientali (Virginia settentrionale). Amazon Cognito crea una CloudFront distribuzione Amazon, protetta in transito con il tuo certificato ACM. Poiché sei il proprietario del dominio, devi creare un record DNS che indirizzi il traffico verso la CloudFront distribuzione del tuo dominio personalizzato.

Dopo che questi elementi sono pronti, puoi aggiungere il dominio personalizzato al tuo pool di utenti tramite la console o l'API di Amazon Cognito. Ciò comporta la specificazione del nome di dominio

e del certificato SSL e quindi l'aggiornamento della configurazione DNS con l'alias target fornito. Dopo aver apportato queste modifiche, puoi verificare che la pagina di accesso sia accessibile nel tuo dominio personalizzato.

Il modo più semplice per creare un dominio personalizzato è utilizzare una zona ospitata pubblica in Amazon Route 53. La console Amazon Cognito può creare i record DNS corretti in pochi passaggi. Prima di iniziare, valuta la possibilità di creare una zona ospitata su Route 53 per un dominio o sottodominio di tua proprietà.

Argomenti

- Aggiunta di un dominio personalizzato a un bacino d'utenza
- Prerequisiti
- Fase 1: Inserimento del nome di dominio personalizzato
- Fase 2: Aggiunta di una destinazione alias e di sottodomini
- Fase 3: Verifica della pagina di accesso
- Modifica del certificato SSL per il dominio personalizzato
- Impostazione della versione TLS minima per il tuo dominio personalizzato

Aggiunta di un dominio personalizzato a un bacino d'utenza

Per aggiungere un dominio personalizzato al bacino d'utenza, devi specificare il nome di dominio nella console Amazon Cognito e fornire un certificato da gestire con AWS Certificate Manager (ACM). Dopo avere aggiunto il dominio, Amazon Cognito fornisce una destinazione alias che devi aggiungere alla configurazione DNS.

Prerequisiti

Prima di iniziare è necessario disporre di quanto segue:

- Un bacino d'utenza con un client dell'App. Per ulteriori informazioni, consulta Crea una nuova applicazione nella console Amazon Cognito.
- Un dominio Web di tua proprietà, Il relativo dominio padre deve avere un record A DNS valido. Puoi assegnare qualsiasi valore a questo record. Il padre può essere la root del dominio o un dominio figlio che è un gradino più in alto nella gerarchia dei domini. Ad esempio, se il dominio personalizzato è auth.xyz.example.com, Amazon Cognito deve poter risolvere xyz.example.com in un indirizzo IP. Per evitare ripercussioni accidentali sull'infrastruttura del cliente, Amazon

Cognito non supporta l'uso di domini di primo livello (TLD) per i domini personalizzati. Per ulteriori informazioni, consulta la pagina relativa ai nomi di dominio.

- La possibilità di creare un dominio secondario per il dominio personalizzato. Ti consigliamo l'autenticazione per il nome del tuo sottodominio. Ad esempio: *auth.example.com*.

> **Note**
>
> Se non disponi di un certificato jolly, potresti dover ottenere un nuovo certificato per il dominio secondario del tuo dominio personalizzato.

- Un SSL/TLS certificato pubblico gestito da ACM negli Stati Uniti orientali (Virginia settentrionale). Il certificato deve essere in us-east-1 perché il certificato è associato a una distribuzione CloudFront in, un servizio globale.
- Client browser che supportano l'indicazione del nome del server (SNI). La CloudFront distribuzione che Amazon Cognito assegna ai domini personalizzati richiede SNI. Non puoi modificare questa impostazione. Per ulteriori informazioni su SNI nelle CloudFront distribuzioni, consulta Usare SNI per servire le richieste HTTPS nella Amazon CloudFront Developer Guide.
- Un'applicazione che consente al server di autorizzazione del pool di utenti di aggiungere cookie alle sessioni utente. Amazon Cognito imposta diversi cookie necessari per le pagine di accesso gestite. Tra queste vi sono `cognito`, `cognito-fl` e `XSRF-TOKEN`. Sebbene ogni singolo cookie sia conforme ai limiti di dimensione del browser, le modifiche alla configurazione del pool di utenti potrebbero causare un aumento delle dimensioni dei cookie di accesso gestiti. Un servizio intermedio come un Application Load Balancer (ALB) davanti al tuo dominio personalizzato potrebbe imporre una dimensione massima dell'intestazione o una dimensione totale dei cookie. Se l'applicazione imposta anche i propri cookie, le sessioni degli utenti potrebbero superare questi limiti. Per evitare conflitti relativi ai limiti di dimensione, consigliamo all'applicazione di non impostare i cookie sul sottodominio che ospita i servizi di dominio del pool di utenti.
- Autorizzazione ad aggiornare CloudFront le distribuzioni Amazon. A tale scopo, puoi associare la dichiarazione di policy IAM seguente a un utente nel tuo Account AWS:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Sid": "AllowCloudFrontUpdateDistribution",
            "Effect": "Allow",
```

```
            "Action": [
                "cloudfront:updateDistribution"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Per ulteriori informazioni sull'autorizzazione delle azioni in CloudFront, consulta Using Identity-Based Policies (IAM Policies) per. CloudFront

Amazon Cognito inizialmente utilizza le tue autorizzazioni IAM per configurare la CloudFront distribuzione, ma la distribuzione è gestita da. AWS Non puoi modificare la configurazione della CloudFront distribuzione associata da Amazon Cognito al tuo pool di utenti. Puoi selezionare la versione TLS minima per la politica di sicurezza della distribuzione quando crei o aggiorni il tuo dominio personalizzato con l'API Amazon Cognito. Per ulteriori informazioni, consulta Impostazione della versione TLS minima per il tuo dominio personalizzato.

Fase 1: Inserimento del nome di dominio personalizzato

È possibile aggiungere il dominio al bacino d'utenza utilizzando la console o l'API Amazon Cognito.

Amazon Cognito console

Aggiungere il dominio al bacino d'utenza dalla console Amazon Cognito:

1. Vai al menu Dominio sotto Branding.
2. Accanto a Dominio, seleziona Operazioni, quindi seleziona Crea dominio personalizzato o Crea dominio Amazon Cognito. Se hai già configurato un dominio personalizzato per un pool di utenti, scegli Elimina dominio personalizzato prima di creare il tuo nuovo dominio personalizzato.
3. Accanto a Dominio, scegli Azioni e seleziona Crea dominio personalizzato. Se hai già configurato un dominio personalizzato, scegli Elimina dominio personalizzato per eliminare il dominio esistente prima di creare il nuovo dominio personalizzato.
4. Alla voce Custom domain (Dominio personalizzato), inserisci l'URL del dominio che vuoi utilizzare con Amazon Cognito. Il nome di dominio può contenere solo lettere minuscole,

numeri e trattini. Non utilizzare un trattino come primo o ultimo carattere. Utilizzare i punti per separare i nomi di dominio secondario.

5. Alla voce ACM certificate (Certificato ACM), scegli il certificato SSL che desideri utilizzare per il tuo dominio. Solo i certificati ACM negli Stati Uniti orientali (Virginia settentrionale) possono essere utilizzati con un dominio personalizzato Amazon Cognito, indipendentemente dal pool di Regione AWS utenti.

   Se non disponi di un certificato, è possibile utilizzare ACM per effettuare il provisioning di uno negli Stati Uniti orientali (Virginia settentrionale). Per ulteriori informazioni, consulta Nozioni di base nella Guida per l'utente di AWS Certificate Manager .

6. Scegli una versione di branding. La tua versione di branding si applica a tutte le pagine interattive con l'utente di quel dominio. Il tuo pool di utenti può ospitare l'accesso gestito o il branding dell'interfaccia utente ospitata per tutti i client dell'app.

   > **Note**
   >
   > Puoi avere un dominio personalizzato e un dominio con prefisso, ma Amazon Cognito serve solo */.well-known/openid-configuration* l'endpoint per il dominio personalizzato.

7. Scegli Crea.
8. Amazon Cognito ti riporta al menu Dominio. Viene visualizzato un messaggio Create an alias record in your domain's DNS (Crea un registro di alias nel DNS del tuo dominio). Prendi nota del Domain (Dominio) e della Alias target (Destinazione alias) visualizzati nella console. Saranno utilizzati nella fase successiva per indirizzare il traffico verso il dominio personalizzato.

API

Il seguente corpo della CreateUserPoolDomainrichiesta crea un dominio personalizzato.

```
{
   "Domain": "auth.example.com",
   "UserPoolId": "us-east-1_EXAMPLE",
   "ManagedLoginVersion": 2,
   "CustomDomainConfig": {
    "CertificateArn": "arn:aws:acm:us-east-1:111122223333:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
    "SecurityPolicy": "TLS_V1_3_2025"
   }
}
```

Il `SecurityPolicy` campo opzionale imposta la versione TLS minima per il dominio personalizzato. Per ulteriori informazioni, consulta Impostazione della versione TLS minima per il tuo dominio personalizzato.

Fase 2: Aggiunta di una destinazione alias e di sottodomini

In questa fase, è possibile configurare, mediante il fornitore di servizi DNS (Domain Name Server), un alias che punti alla destinazione di alias della fase precedente. Se per la risoluzione dell'indirizzo DNS utilizzi Amazon Route 53, passa alla sezione Come aggiungere una destinazione alias e un sottodominio utilizzando Route 53.

Per aggiungere una destinazione alias e un sottodominio all'attuale configurazione DNS

- Se non utilizzi Route 53 per la risoluzione dell'indirizzo DNS, dovrai utilizzare gli strumenti di configurazione del tuo provider di servizi DNS per aggiungere la destinazione alias della fase precedente al registro DNS del tuo dominio. Il fornitore DNS dovrà inoltre configurare il sottodominio per il dominio personalizzato.

Come aggiungere una destinazione alias e un sottodominio utilizzando Route 53

1. Accedi alla console Route 53. Se richiesto, inserisci le tue AWS credenziali.
2. Se non disponi di una zona ospitata pubblica in Route 53, creane una con una radice che sia l'elemento principale del tuo dominio personalizzato. Per ulteriori informazioni, consulta Creazione di una zona ospitata pubblica nella Amazon Route 53 Developer Guide.
   a. Scegli Create Hosted Zone (Crea zona ospitata).
   b. Inserisci il dominio principale, ad esempio*auth.example.com*, del tuo dominio personalizzato*myapp.auth.example.com*, ad esempio dall'elenco dei nomi di dominio.
   c. Inserisci una Descrizione per la zona ospitata.
   d. Scegli un Tipo di zona ospitata di una Zona ospitata pubblica per consentire ai client pubblici di risolvere il tuo dominio personalizzato. La scelta della Zona ospitata privata non è supportata.
   e. Applica Tag come vuoi.

f. Scegli Crea zona ospitata.

> **Note**
>
> Puoi anche creare una nuova zona ospitata per il tuo dominio personalizzato con una delega impostata nella zona ospitata principale che indirizza le query alla zona ospitata del sottodominio. Altrimenti, crea un record A. Questo metodo offre maggiore flessibilità e sicurezza con le zone ospitate. Per ulteriori informazioni, consulta Creazione di un sottodominio per un dominio ospitato tramite Amazon Route 53.

3. Nella pagina Hosted Zones (Zone ospitate), scegli il nome della tua zona ospitata.
4. Aggiungi un record DNS per il dominio principale del tuo dominio personalizzato, se non ne hai già uno. Crea un record DNS per il dominio principale con le seguenti proprietà:

   - Nome del record: lascia vuoto.
   - Tipo di record:`A`.
   - Alias: non abilitare.
   - Valore: inserisci un obiettivo a tua scelta. Questo record deve risolversi in qualcosa, ma il valore del record non ha importanza per Amazon Cognito.
   - TTL: imposta il TTL preferito o lascialo come predefinito.
   - Politica di routing: scegli Simple routing.

5. Scegli Crea record. Di seguito è riportato un record di esempio per il dominio: *`example.com`*

   `example.com.` `60 IN A` *`198.51.100.1`*

   > **Note**
   >
   > Amazon Cognito verifica che sia presente un registro DNS per il dominio principale del tuo dominio personalizzato per proteggerti dal dirottamento accidentale dei domini di produzione. Se non disponi di un record DNS per il dominio principale, Amazon Cognito restituisce un errore quando tenti di impostare il dominio personalizzato. Un record Start of Authority (SOA) non è un record DNS sufficiente ai fini della verifica del dominio principale.

6. Aggiungi un altro record DNS per il tuo dominio personalizzato con le seguenti proprietà:

- Nome del record: il prefisso di dominio personalizzato, ad esempio per `auth` cui creare un record per. `auth.example.com`
- Tipo di record:A.
- Alias: Abilita.
- Indirizza il traffico verso: scegli Alias verso la distribuzione Cloudfront. Inserisci, ad esempio, l'alias target che hai registrato in precedenza. `123example.cloudfront.net`
- Politica di routing: scegli Simple routing.

7. Scegli Crea record.

> **Note**
>
> I nuovi registri possono richiedere circa 60 secondi per la propagazione a tutti i server DNS di Route 53. Puoi utilizzare il metodo dell'GetChangeAPI Route 53 per verificare che le modifiche si siano propagate.

Fase 3: Verifica della pagina di accesso

- Verifica che la pagina di accesso sia disponibile dal tuo dominio personalizzato.

  Accedi con il tuo dominio personalizzato e con il sottodominio immettendo questo indirizzo nel browser. Questo è un esempio di URL di un dominio personalizzato *`example.com`* con il *`auth`* sottodominio:

```
https://myapp.auth.example.com/login?
response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callback_url>
```

Modifica del certificato SSL per il dominio personalizzato

Se necessario, puoi utilizzare Amazon Cognito per modificare il certificato che hai applicato al tuo dominio personalizzato.

Di solito, questa operazione non è necessaria dopo la procedura di rinnovo dei certificati con ACM. Quando rinnovi il certificato esistente in ACM, l'ARN del tuo certificato rimane invariato e il tuo dominio personalizzato utilizza automaticamente il nuovo certificato.

Tuttavia, se sostituisci il certificato esistente con uno nuovo, ACM assegna un nuovo ARN al nuovo certificato. Per applicare il nuovo certificato al dominio personalizzato, devi fornire questo ARN ad Amazon Cognito.

Dopo avere fornito il nuovo certificato, Amazon Cognito impiega fino a 1 ora per distribuirlo nel dominio personalizzato.

> **Prima di iniziare**
>
> Prima di poter modificare il certificato in Amazon Cognito, devi aggiungerlo ad ACM. Per ulteriori informazioni, consulta Nozioni di base nella Guida per l'utente di AWS Certificate Manager .
> Quando aggiungi il certificato ad ACM, devi scegliere Stati Uniti orientali (Virginia settentrionale) come regione AWS .

Puoi modificare il tuo certificato utilizzando la console o l'API Amazon Cognito.

Console di gestione AWS

Rinnovare un certificato dalla console Amazon Cognito:

1. Accedi Console di gestione AWS e apri la console Amazon Cognito all'indirizzo. https://console.aws.amazon.com/cognito/home
2. Scegli Bacini d'utenza.
3. Scegli il bacino d'utenza per cui desideri aggiornare il certificato.
4. Scegli il menu Dominio.
5. Scegli Operazioni, Edit ACM certificate (Modifica del certificato ACM.
6. Scegli il nuovo certificato che desideri associare al dominio personalizzato.
7. Scegli Save changes (Salva modifiche).

API

Come rinnovare un certificato (API Amazon Cognito)

- Usa l'azione UpdateUserPoolDomain.

Impostazione della versione TLS minima per il tuo dominio personalizzato

Quando crei o aggiorni un dominio personalizzato con la console o l'API di Amazon Cognito, puoi selezionare la versione TLS minima per la distribuzione Amazon. CloudFront La distribuzione utilizza questa versione quando comunica con i client. Nell'API, imposta la versione TLS minima nel `SecurityPolicy` campo dell'`CustomDomainConfig`oggetto in una richiesta CreateUserPoolDomainor UpdateUserPoolDomain. La politica di sicurezza definisce la versione TLS minima e le suite di crittografia CloudFront supportate durante la comunicazione con i client. Per indicazioni specifiche, consulta Protocolli e cifrature supportati tra i visualizzatori e CloudFront nella Amazon CloudFront Developer Guide. Puoi selezionare uno dei seguenti valori.

`TLS_V1_3_2025`(il più rigoroso)

Una policy pronta per l'uso post-quantistico che richiede TLS 1.3. Fornisce il livello di sicurezza più elevato ed è ideale per carichi di lavoro in cui tutti i client e i browser sono aggiornati alle versioni più recenti. Per ulteriori informazioni, consulta Protocolli e cifrari supportati per _2025. TLSv1.3

`TLS_V1_2_2021` (consigliato)

Una politica post-quantum ready che preferisce TLS 1.3 ma consente il fallback a TLS 1.2 per soddisfare i client più vecchi. È il valore minimo consigliato per le tipiche applicazioni consumer di livello commerciale. Per ulteriori informazioni, consulta Protocolli e cifrari supportati per _2021. TLSv1.2

`TLS_V1`(fortemente sconsigliato)

Consente il fallback a TLS 1.0. Offre la più ampia compatibilità, incluso il supporto per client legacy che hanno più di dieci anni. Questa compatibilità va a scapito dell'autorizzazione di versioni TLS e algoritmi crittografici che non sono più considerati sicuri per l'uso commerciale. Per ulteriori informazioni, consulta Protocolli e cifrari supportati per TLSv1.

> **Note**
>
> I domini personalizzati che utilizzano l'accesso gestito (versione di branding 2) richiedono una versione TLS minima pari o superiore. `TLS_V1_2_2021` Una richiesta impostata `TLS_V1` per un dominio di accesso gestito restituisce un errore. `InvalidInputException`

Puoi selezionare la versione TLS minima dalla console Amazon Cognito o con l'API Amazon Cognito, AWS Command Line Interface il AWS CLI() e gli SDK. AWS

Il seguente corpo della UpdateUserPoolDomainrichiesta aggiorna la versione TLS minima di un dominio personalizzato esistente.

```
{
   "Domain": "auth.example.com",
   "UserPoolId": "us-east-1_EXAMPLE",
   "ManagedLoginVersion": 2,
   "CustomDomainConfig": {
    "CertificateArn": "arn:aws:acm:us-east-1:111122223333:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "SecurityPolicy": "TLS_V1_3_2025"
   }
}
```

Applica il branding alle pagine di accesso gestite

Potresti voler fornire un'esperienza utente coerente tra il tuo servizio di autenticazione e l'applicazione. Puoi raggiungere questo obiettivo con moduli personalizzati e operazioni API di back-end in un AWS SDK o con l'accesso gestito. L'accesso gestito e la classica interfaccia utente ospitata sono front-end Web per il componente dell'applicazione che serve l'autenticazione con pool di utenti. Per sincronizzare i servizi di autenticazione gestiti con l'esperienza utente dell'applicazione, sono disponibili due opzioni di personalizzazione: l'editor di branding e il branding dell'interfaccia utente ospitata. Puoi scegliere la tua esperienza preferita nella console Amazon Cognito e con le operazioni API del pool di utenti.

L'editor di branding

> L'editor di branding è l'opzione di personalizzazione più recente per la più recente esperienza di interfaccia utente dei pool di utenti, l'accesso gestito. Il branding editor è un editor visivo senza codice per la gestione degli asset e dello stile di accesso e un set di operazioni API per la configurazione programmatica di un gran numero di opzioni di configurazione. I pool di utenti configurati con un dominio e l'accesso gestito visualizzano automaticamente la versione di branding-designer delle tue pagine di accesso.

Branding dell'interfaccia utente ospitata (classica)

> L'esperienza di branding dell'interfaccia utente ospitata (classica) offre due opzioni: modificare un file CSS (Cascading Stylesheets) con un set fisso di opzioni di stile e aggiungere un'immagine

del logo personalizzata. Puoi impostare queste opzioni nella console Amazon Cognito o con l'operazione SetUICustomization API. Al momento del lancio del servizio, Amazon Cognito aveva solo questa opzione. I pool di utenti configurati con un dominio e la versione di branding dell'interfaccia utente ospitata visualizzano automaticamente la versione classica delle tue pagine di accesso. Il tuo piano di funzionalità potrebbe inoltre supportare solo l'interfaccia utente ospitata.

> **Note**
>
> L'editor di branding e la classica esperienza di branding modificano le proprietà visive del servizio di autenticazione ospitato. Al momento, non è possibile modificare il testo visualizzato nelle pagine di accesso gestite, tranne per applicare la localizzazione in una delle diverse lingue. Per ulteriori informazioni sulla localizzazione, consulta. Managed login localization

Scegli un'esperienza di branding e assegna stili

Nella console Amazon Cognito, i nuovi pool di utenti utilizzano per impostazione predefinita l'esperienza di branding di accesso gestito. I pool di utenti configurati prima che l'accesso gestito fosse disponibile hanno il marchio Hosted UI (classico). Puoi passare dall'accesso gestito al branding dell'interfaccia utente ospitata. Quando modifichi la versione del tuo marchio, Amazon Cognito applica immediatamente la modifica alle pagine interattive con l'utente del dominio del tuo pool di utenti. Con l'accesso gestito e l'interfaccia utente ospitata, il tuo pool di utenti può avere uno stile per ogni client dell'app.

Ogni client dell'app può avere uno stile di branding distinto, ma un dominio del pool di utenti serve l'accesso gestito o l'interfaccia utente ospitata. Uno stile è l'insieme di impostazioni di personalizzazione applicate a un client di app. È possibile configurare un dominio personalizzato e un dominio con prefisso per pool di utenti. Puoi assegnare diverse versioni di branding ai tuoi domini personalizzati e con prefisso. Tuttavia, un dominio con prefisso non è completamente funzionante quando si dispone anche di un dominio personalizzato: gli endpoint di rilevamento OIDC presentano solo percorsi di dominio `.well-known` personalizzati. Puoi utilizzare il dominio del prefisso solo per operazioni che non richiedono endpoint discovery () in un pool di utenti con questa configurazione. `openid-configuration` Grazie a queste proprietà dei pool di utenti, è possibile scegliere in modo efficace una versione di branding per pool di utenti.

È possibile assegnare stili ai client dell'app in un pool di utenti in cui un dominio è impostato sulla versione di branding ad accesso gestito. Gli stili sono un insieme di impostazioni visive composto da file di immagine, opzioni di visualizzazione, valori CSS. Quando assegni uno stile a un client di app, Amazon Cognito invia immediatamente gli aggiornamenti alle tue pagine di accesso interattive per l'utente. Amazon Cognito rende le tue pagine interattive per l'utente con la versione di branding scelta e la personalizzazione che hai applicato.

Aggiorna ed elimina gli stili

Quando crei uno stile, lo colleghi a un client dell'app. Per modificare l'assegnazione di uno stile per un client di app, devi prima eliminare lo stile originale. Al momento, non è possibile copiare le impostazioni tra gli stili. È necessario eseguire questa operazione a livello di codice. Per replicare le impostazioni tra stili e client di app, ottieni le impostazioni relative a uno stile con l'operazione DescribeManagedLoginBrandingAPI e applicale con o. CreateManagedLoginBrandingUpdateManagedLoginBranding Non puoi modificare gli stili assegnati a un client di app, puoi solo eliminare l'originale e impostarne uno nuovo. Per ulteriori informazioni sulla gestione degli stili con le operazioni API e SDK, consulta. Operazioni API e SDK per il branding degli accessi gestiti

> **Note**
>
> Le richieste programmatiche che creano o aggiornano lo stile di branding non devono avere una dimensione di richiesta superiore a 2 MB. Se la tua richiesta supera questo limite, suddividi la richiesta in più `UpdateManagedLoginBranding` richieste per gruppi di parametri che non superano la dimensione massima della richiesta. Queste richieste non comportano l'impostazione di parametri non specificati come predefiniti, quindi puoi inviare richieste parziali senza alcun effetto sulle impostazioni esistenti.

Elimini uno stile nella console Amazon Cognito dal menu Accesso gestito. In Stili, scegli lo stile che desideri eliminare e scegli Elimina stile.

Ad alto livello, il processo di assegnazione del marchio a un dominio consiste nei seguenti passaggi.

1. Crea un dominio e imposta la versione di branding.
2. Crea uno stile di branding e assegnalo a un client dell'app.

Per assegnare uno stile a un client di app

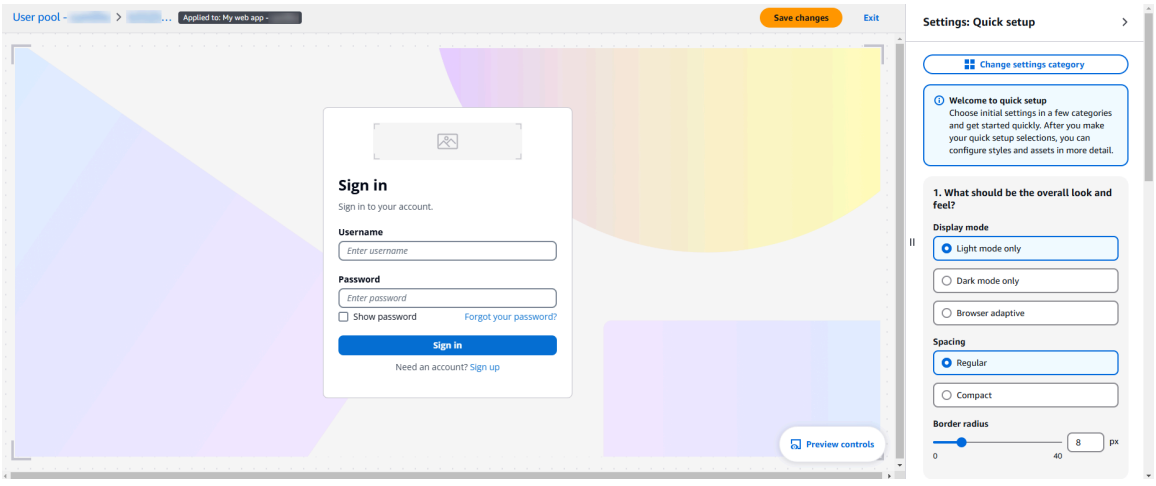
1. Nel menu Dominio del tuo pool di utenti, crea un dominio e imposta la versione di branding su Accesso gestito.
2. Vai al menu Accesso gestito. In Stili, scegli Crea uno stile.
3. Scegli il client dell'app a cui vuoi assegnare il tuo stile o crea un nuovo client per l'app.
4. Per iniziare a configurare le impostazioni di branding, scegli Launch branding editor.

Argomenti

- Il branding editor e la personalizzazione dell’accesso gestito

Il branding editor e la personalizzazione dell’accesso gestito

L'editor di branding è uno strumento di progettazione visiva e modifica per le pagine Web di accesso gestite. È integrato nella console Amazon Cognito. Nell'editor di branding, inizi con un'anteprima delle tue pagine di accesso e puoi passare a un'opzione di configurazione rapida o a una visualizzazione dettagliata con opzioni avanzate. Puoi modificare e visualizzare in anteprima i parametri di stile o aggiungere un'immagine di sfondo e un logo personalizzati. È possibile configurare la modalità luce e la modalità scura.



Per iniziare, crea uno stile da applicare al tuo pool di utenti o a un client di app.

Per iniziare a usare il branding editor

1. Crea un dominio dalla scheda Dominio o aggiorna il dominio esistente. In Versione di branding, imposta il dominio per utilizzare l'accesso gestito.

2. Elimina lo stile del client dell'app esistente, se presente.

   a. Nel menu App client, seleziona il client dell'app.

   b. In Stile di accesso gestito, seleziona lo stile assegnato al client dell'app.

   c. Scegli Elimina stile. Conferma la selezione.

3. Vai al menu di accesso gestito nel tuo pool di utenti. Se non l'hai già fatto, segui le istruzioni per selezionare un piano di funzionalità che includa l'accesso gestito. Puoi anche selezionare Visualizza l'anteprima di questa funzionalità se desideri dare un'occhiata all'editor di branding senza apportare modifiche.

4. In Stili, scegli Crea uno stile.

5. Scegli il client dell'app a cui vuoi assegnare lo stile e seleziona Crea. Puoi anche creare un nuovo client per l'app.

6. La console Amazon Cognito lancia l'editor di branding.

7. Scegli una scheda in cui desideri iniziare a modificare oppure seleziona Avvia editor e accedi alla configurazione rapida. Sono disponibili le seguenti schede:

   Anteprima

   Guarda come appaiono le tue selezioni correnti nelle pagine di accesso gestite.

   Concetti di base

   Imposta un tema generale, configura i collegamenti a provider di identità esterni e definisci lo stile dei campi del modulo.

   Componenti

   Configura gli stili per intestazioni, piè di pagina e singoli elementi dell'interfaccia utente.

8. Per effettuare delle scelte sulle impostazioni iniziali, accedi alla configurazione rapida. Seleziona la categoria Modifica impostazioni e scegli Configurazione rapida. Quando selezioni Procedi, viene avviato l'editor di branding con una serie di opzioni di base da configurare.

Testo e localizzazione

Non puoi modificare o localizzare il testo nell'editor di branding. Aggiungi invece un parametro di `lang` query all'URL che distribuisci agli utenti. Questo parametro fa sì che le pagine di accesso gestite vengano localizzate in una delle diverse lingue disponibili. Per ulteriori informazioni, consulta Managed login localization.

Configurazione rapida

Il pulsante Launch branding editor carica un editor visuale per la configurazione di accesso gestito in cui è possibile scegliere tra una varietà di opzioni di personalizzazione principali. Man mano che effettui le selezioni, Amazon Cognito visualizza le modifiche di accesso gestite in una finestra di anteprima. Per tornare al menu delle impostazioni dettagliate, seleziona il pulsante Modifica la categoria delle impostazioni.

Quale dovrebbe essere l'aspetto generale?

Configura le impostazioni di base del tema per l'accesso gestito.

Modalità di visualizzazione

Scegli una modalità chiara, una modalità oscura o un'esperienza adattiva per il tuo accesso gestito. Le impostazioni adattive rimandano alle preferenze del browser dell'utente quando Amazon Cognito esegue il rendering dell'accesso gestito. Quando scegli una modalità adattiva al browser, puoi scegliere colori e immagini del logo diversi per la modalità chiara e scura.

Spacing

Imposta la spaziatura predefinita tra gli elementi della pagina.

Raggio del bordo

Imposta la profondità di arrotondamento del bordo esterno degli elementi.

Come deve apparire lo sfondo della pagina?

Tipo di sfondo

La casella di controllo Mostra immagine indica se si desidera un'immagine di sfondo o se si desidera impostare un colore di sfondo solido.

1. Per usare un'immagine, seleziona Mostra immagine e scegli un'immagine di sfondo per le modalità chiare e scure. Puoi anche impostare un colore di sfondo della pagina in modalità scura e chiara per le aree dello sfondo che non sono coperte dall'immagine.
2. Per utilizzare solo un colore per lo sfondo, deselezionate Mostra immagine e scegliete un colore di sfondo della pagina in modalità chiara e scura.

Che aspetto devono avere i moduli?

Configura le impostazioni per gli elementi del modulo di accesso gestito. Esempi di elementi del modulo sono le istruzioni di accesso e di codice.

Allineamento orizzontale

Imposta l'allineamento orizzontale dei campi del modulo.

Logo del modulo

Imposta il posizionamento dell'immagine del logo.

Immagine del logo

Scegliete un file di immagine del logo da includere nell'elemento del modulo per le modalità chiare e scure. Per caricare un'immagine, seleziona il menu a discesa Immagine del logo, scegli Aggiungi nuova risorsa e aggiungi un file del logo.

Colore principale del marchio

Imposta un colore del tema per le modalità chiaro e scuro. Questo colore viene applicato come colore di sfondo a tutti gli elementi classificati come primari.

Che aspetto devono avere le intestazioni?

Scegli se includere un'intestazione nelle tue pagine di accesso gestite. L'intestazione può contenere un'immagine del logo.

Logo dell'intestazione

Imposta la posizione dell'immagine del logo nell'intestazione.

Immagine del logo

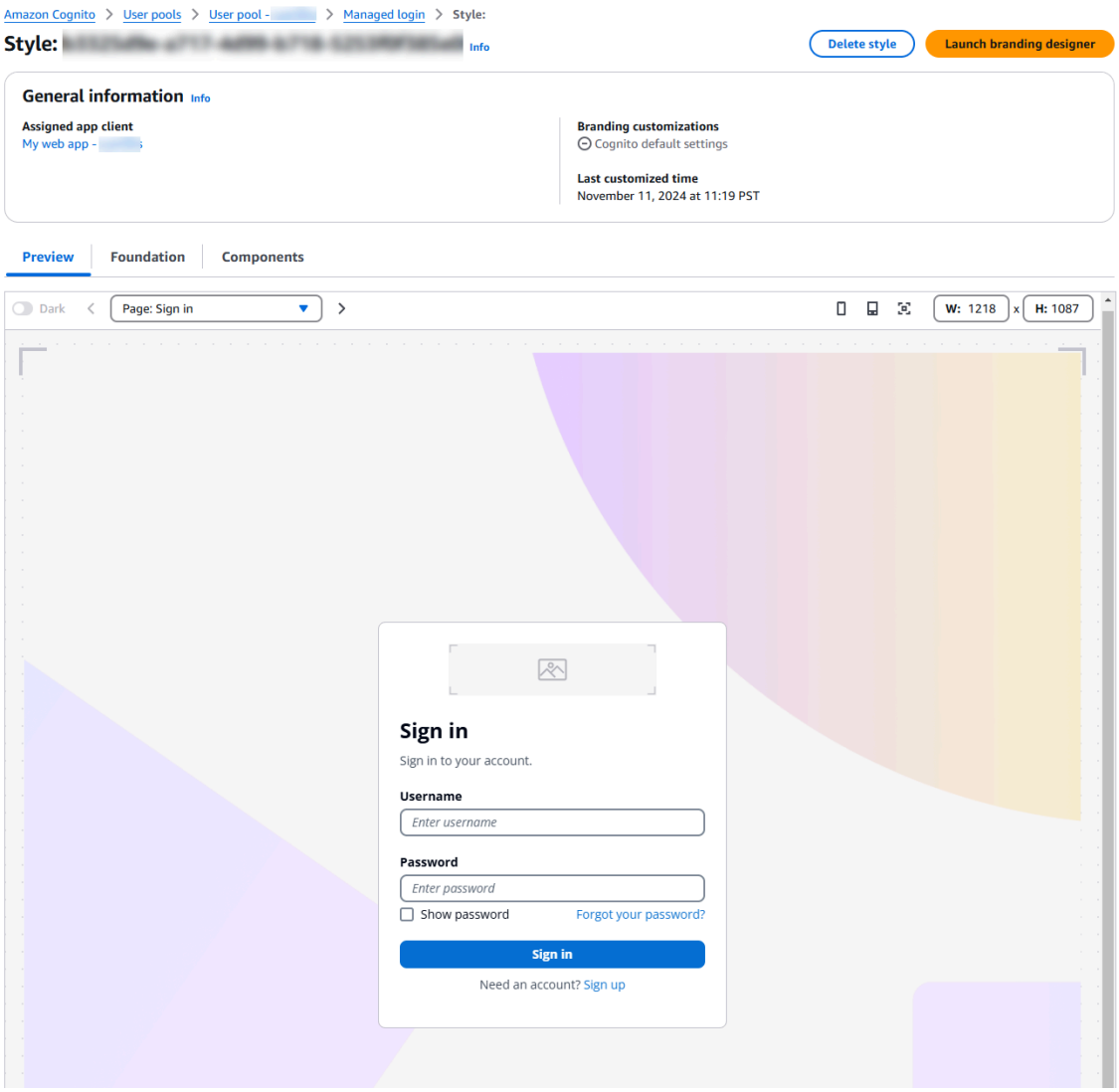
Scegli una posizione del logo e un file di immagine del logo da includere nell'intestazione. Per caricare un'immagine, seleziona il menu a discesa Immagine del logo, scegli Aggiungi nuova risorsa e aggiungi un file del logo.

Colore di sfondo dell'intestazione

Imposta i colori delle modalità chiaro e scuro per lo sfondo dell'intestazione.

Impostazioni dettagliate

Nella vista delle impostazioni dettagliate, è possibile modificare i singoli componenti in Foundation e Components. La scheda Anteprima mostra un'anteprima dell'accesso gestito nel contesto corrente con le personalizzazioni.



Per accedere all'editor visivo di un componente, scegli l'icona di modifica nel riquadro del componente. Dall'editor dello studio tematico, puoi passare da un componente all'altro con il pulsante Modifica la categoria delle impostazioni.

Concetti di base

Stile dell'app

Configura le basi della tua configurazione di accesso gestito. Questa categoria include impostazioni per il tema generale, la spaziatura del testo e l'intestazione e il piè di pagina della pagina.

Modalità di visualizzazione

Scegli una modalità chiara, una modalità scura o un'esperienza adattiva per le tue pagine di accesso gestite. Quando scegli una modalità adattiva al browser, puoi scegliere colori e immagini del logo diversi per la modalità chiara e scura.

Spacing

Imposta la spaziatura predefinita tra gli elementi della pagina.

Comportamento di autenticazione

Configura gli stili per i pulsanti che collegano gli utenti ai provider di identità esterni (IdPs). Questa sezione include l'opzione Domain search input per fare in modo che l'accesso gestito richieda agli utenti un indirizzo e-mail e lo abbini all'identificativo del provider di identità SAML.

Comportamento del modulo

Configura gli stili per i moduli di input: posizionamento degli input, colori e allineamento degli elementi.

Componenti

Pulsanti

Stili per i pulsanti visualizzati da Amazon Cognito nelle pagine di accesso gestite.

Divisore

Stili per le linee di divisione e i confini tra gli elementi di accesso gestiti come il modulo di input e il selettore di accesso del provider esterno.

Elenco a discesa

Stili per i menu a discesa.

Favicon

Stili per l'immagine che Amazon Cognito fornisce per l'icona della scheda e del segnalibro.

anelli di messa a fuoco

Stili per le evidenziazioni che indicano un input attualmente selezionato.

Contenitore del modulo

Stili per gli elementi che legano un modulo.

Piè di pagina globale

Stili per il piè di pagina visualizzato da Amazon Cognito nella parte inferiore delle pagine di accesso gestite.

Intestazione globale

Stili per l'intestazione che Amazon Cognito mostra nella parte superiore delle pagine di accesso gestite.

Indicazioni

Stili per i messaggi di errore e di successo.

Controlli delle opzioni

Stili per caselle di controllo, selezioni multiple e altri prompt di input.

Sfondo della pagina

Stili per lo sfondo generale dell'accesso gestito.

Input

Stili per le richieste di immissione dei campi del modulo.

Link

Stili per i collegamenti ipertestuali nelle pagine di accesso gestite.

Testo per pagina

Stili per il testo all'interno della pagina.

Testo per campo

Stili per il testo attorno agli input del modulo.

Operazioni API e SDK per il branding degli accessi gestiti

Puoi anche applicare il branding a uno stile di accesso gestito con le operazioni API e. CreateManagedLoginBrandingUpdateManagedLoginBranding Queste operazioni sono ideali per creare versioni identiche o leggermente modificate di uno stile di branding per un altro client di app o pool di utenti. Interroga il branding di accesso gestito di uno stile esistente con l'operazione API

DescribeManagedLoginBranding, quindi modifica l'output in base alle esigenze e applicalo a un'altra risorsa.

L'`UpdateManagedLoginBranding`operazione non modifica il client dell'app a cui viene applicato lo stile. Aggiorna solo lo stile esistente assegnato a un client dell'app. Per sostituire completamente lo stile di un client di app, elimina lo stile esistente con DeleteManagedLoginBrandinge assegnane uno nuovo con`CreateManagedLoginBranding`. Nella console Amazon Cognito, lo stesso vale: devi eliminare lo stile esistente e crearne uno nuovo.

La configurazione del branding di accesso gestito in una richiesta API o SDK richiede che le impostazioni siano incorporate in un file JSON convertito in un tipo di dati. `Document` Di seguito sono riportate le indicazioni per le immagini che è possibile aggiungere e per generare richieste programmatiche per configurare uno stile di branding.

Risorse di immagini

CreateManagedLoginBrandinge UpdateManagedLoginBrandingincludi un `Assets` parametro. Questo parametro è una matrice di file di immagine in formato binario con codifica base64.

> **Note**
>
> Le richieste programmatiche che creano o aggiornano lo stile di branding devono avere una dimensione non superiore a 2 MB. Le risorse contenute nella tua richiesta potrebbero farla superare questo limite. In tal caso, suddividi la richiesta in più `UpdateManagedLoginBranding` richieste di gruppi di parametri che non superino la dimensione massima della richiesta. Queste richieste non comportano l'impostazione di parametri non specificati come predefiniti, quindi puoi inviare richieste parziali senza alcun effetto sulle impostazioni esistenti.

Alcune risorse presentano limitazioni sui tipi di file che è possibile inviare.

Asset	Estensioni di file accettate
FAVICON_ICO	ico
FAVICON_SVG	svg
GRAFICA_EMAIL	png, svg, jpeg

Asset	Estensioni di file accettate
GRAFICA SMS	png, svg, jpeg
AUTH_APP_GRAPHIC	png, svg, jpeg
GRAFICA DELLA PASSWORD	png, svg, jpeg
PASSKEY_GRAPHIC	png, svg, jpeg
LOGO_HEADER_PAGINA	png, svg, jpeg
SFONDO_INTESTAZIONE_PAGINA	png, svg, jpeg
LOGO_PAGINA A PIÈ DI PAGINA	png, svg, jpeg
SFONDO DELLA PAGINA A PIÈ DI PAGINA	png, svg, jpeg
SFONDO DELLA PAGINA	png, svg, jpeg
FORM_SFONDO	png, svg, jpeg
FORM_LOGO	png, svg, jpeg
ICONA_IDP_BUTTON_	icona, svg

I file di tipo SVG supportano i seguenti attributi ed elementi.

Attributes

```
accent-height, accumulate, additivive, alignment-baseline, ascent, attributename,
 attributetype, azimuth, basefrequency, baseline-shift, begin, bias, by, class,
 clip, clip-path, clip-rule, color, color-interpolation, color-interpolation-
filters, color-profile, color-rendering, cx, cy, d, dx, dy, diffuseconstant,
 direction, display, divisor, dur, edgemode, elevation, end, fill, fill-opacity,
 fill-rule, filter, filterunits, flood-color, flood-opacity, font-family, font-
size, font-size-adjust, font-stretch, font-style, font-variant, font-weight, fx,
 fy, g1, g2, glyph-name, glyphref, gradientunits, gradienttransform, height, href,
 id, image-rendering, in, in2, k, k1, k2, k3, k4, kerning, keypoints, keysplines,
 keytimes, lang, lengthadjust, letter-spacing, kernelmatrix, kernelunitlength,
 lighting-color, local, marker-end, marker-mid, marker-start, markerheight,
 markerunits, markerwidth, maskcontentunits, maskunits, max, mask, media,
```

```
 method, mode, min, name, numoctaves, offset, operator, opacity, order, orient,
 orientation, origin, overflow, paint-order, path, pathlength, patterncontentunits,
 patterntransform, patternunits, points, preservealpha, preserveaspectratio, r,
 rx, ry, radius, refx, refy, repeatcount, repeatdur, restart, result, rotate,
 scale, seed, shape-rendering, specularconstant, specularexponent, spreadmethod,
 stddeviation, stitchtiles, stop-color, stop-opacity, stroke-dasharray, stroke-
dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity,
 stroke, stroke-width, style, surfacescale, tabindex, targetx, targety, transform,
 text-anchor, text-decoration, text-rendering, textlength, type, u1, u2, unicode,
 values, viewbox, visibility, vert-adv-y, vert-origin-x, vert-origin-y, width, word-
spacing, wrap, writing-mode, xchannelselector, ychannelselector, x, x1, x2, xmlns,
 y, y1, y2, z, zoomandpan
```

Elements

```
svg, a, altglyph, altglyphdef, altglyphitem, animatecolor, animatemotion,
 animatetransform, audio, canvas, circle, clippath, defs, desc, ellipse, filter,
 font, g, glyph, glyphref, hkern, image, line, lineargradient, marker, mask,
 metadata, mpath, path, pattern, polygon, polyline, radialgradient, rect, stop,
 style, switch, symbol, text, textpath, title, tref, tspan, video, view, vkern,
 feBlend, feColorMatrix, feComponentTransfer, feComposite, feConvolveMatrix,
 feDiffuseLighting, feDisplacementMap, feDistantLight, feFlood, feFuncA, feFuncB,
 feFuncG, feFuncR, feGaussianBlur, feMerge, feMergeNode, feMorphology, feOffset,
 fePointLight, feSpecularLighting, feSpotLight, feTile, feTurbulence
```

Strumenti per operazioni gestite di branding degli accessi

Amazon Cognito gestisce un file nel JSON-Schema formato per l'oggetto delle impostazioni di branding di accesso gestito. Di seguito viene descritto come aggiornare in modo programmatico lo stile di branding.

Per aggiornare il branding nell'API dei pool di utenti

1. Nella console Amazon Cognito, crea uno stile di branding di accesso gestito predefinito dal menu Accesso gestito del tuo pool di utenti. Assegnalo a un client dell'app.
2. Ad esempio`1example23456789`, registra l'ID del client dell'app per cui hai creato lo stile.
3. Recupera le impostazioni per lo stile di branding con una richiesta DescribeManagedLoginBrandingByClientAPI `ReturnMergedResources` impostata su. `true` Di seguito è riportato un esempio di corpo della richiesta.

```
{
   "ClientId": "1example23456789",
   "ReturnMergedResources": true,
   "UserPoolId": "us-east-1_EXAMPLE"
}
```

4. Modifica l'output di `DescribeManagedLoginBrandingByClient` con le tue personalizzazioni.

   a. Il corpo della risposta è racchiuso in un `ManagedLoginBranding` elemento che non fa parte della sintassi per le operazioni di creazione e aggiornamento. Rimuovi questo livello superiore dell'oggetto JSON.

   b. Per sostituire le immagini, sostituisci il `Bytes` valore con i dati Base64-encoded binari di ogni file di immagine.

   c. Per aggiornare le impostazioni, modificate l'output dell'`Settings`oggetto e includetelo nella richiesta successiva. Amazon Cognito ignora tutti i valori `Settings` dell'oggetto che non sono presenti nello schema ricevuto nella risposta all'API.

5. Usa il corpo della risposta aggiornato in una richiesta CreateManagedLoginBrandingor UpdateManagedLoginBranding. Se questa richiesta supera i 2 MB, suddividila in più richieste. Queste operazioni funzionano in un `PATCH` modello in cui le impostazioni originali rimangono invariate, a meno che non venga specificato diversamente.

Migrazione dalla classica interfaccia utente ospitata all'accesso gestito

Se il tuo pool di utenti utilizza attualmente la classica interfaccia utente ospitata, puoi migrarla all'accesso gestito. L'accesso gestito e l'interfaccia utente ospitata classica sono le due versioni di branding. Devi impostare la versione di branding sul dominio del pool di utenti e la tua scelta si applica a tutti i client di app che utilizzano quel dominio, quindi la migrazione cambia tutti i client di app del dominio contemporaneamente. L'accesso gestito aggiunge un editor di branding visivo, l'accesso con passkey e funzionalità di autenticazione aggiuntive. L'interfaccia utente ospitata classica utilizza il branding basato su file ed è disponibile in tutti i piani di funzionalità. Per un confronto funzionalità per funzionalità, consulta. the section called “Login gestito”

> **Important**
>
> L'accesso gestito e il relativo editor di branding richiedono il piano di funzionalità Essentials o Plus. La classica interfaccia utente ospitata è disponibile in tutti i piani, incluso Lite. Conferma il tuo piano di funzionalità prima di effettuare la migrazione. Consulta the section called “Login gestito”.

Ad alto livello, lo spostamento di un pool di utenti esistente dalla classica interfaccia utente ospitata all'accesso gestito prevede i seguenti passaggi. Esamina ogni passaggio rispetto alla configurazione corrente prima di passare:

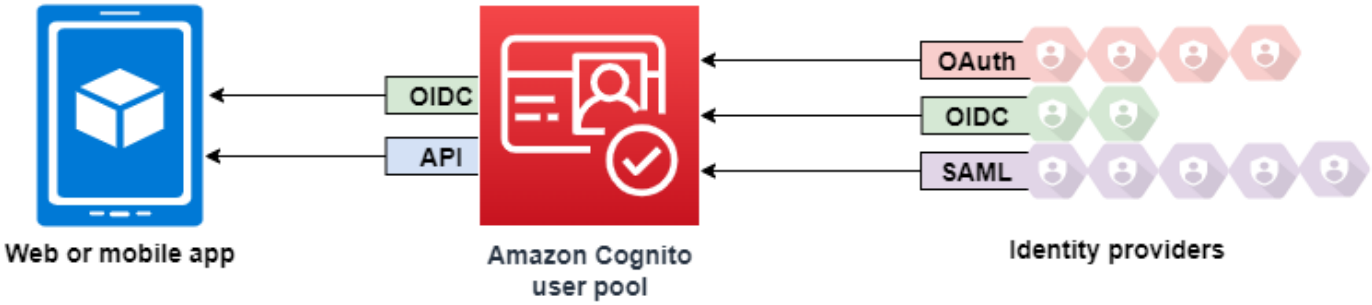
1. Verifica che il tuo pool di utenti utilizzi un piano di funzionalità che supporti l'accesso gestito. Consulta the section called “Login gestito”.
2. Imposta la versione di branding sul dominio del tuo pool di utenti su Managed Login. La versione di branding è un'impostazione del dominio, quindi si applica a tutti i client dell'app che utilizzano quel dominio.
3. Configura il tuo marchio nell'editor di branding visivo. L'accesso gestito utilizza un modello di branding diverso rispetto ai CSS e alla personalizzazione del logo basati su file dell'interfaccia utente ospitata classica, quindi rivedi il tuo marchio esistente confrontandolo con l'editor di branding come parte del passaggio. Consulta the section called “Branding e personalizzazione”.
4. Testa i flussi di accesso, iscrizione, autenticazione a più fattori (MFA) e reimpostazione della password nelle pagine di accesso gestite prima di indirizzare il traffico di produzione verso di esse.

> **Note**
>
> Questa pagina riassume la migrazione a livello concettuale. Le procedure dettagliate di configurazione e branding a cui fa riferimento sono la fonte autorevole per ogni passaggio.

Autenticazione degli utenti con pool di utenti di Amazon Cognito

Un pool di utenti Amazon Cognito è una directory utente per l'autenticazione e l'autorizzazione di app web e per dispositivi mobili. Dal punto di vista dell'app, un pool di utenti Amazon Cognito è un gestore dell'identità digitale OpenID Connect (OIDC). Un pool di utenti aggiunge ulteriori livelli di funzionalità per la sicurezza, la federazione delle identità, l'integrazione app e la personalizzazione dell'esperienza utente.

Questo capitolo descrive il modo in cui gli utenti si autenticano: registrazione, accesso e i flussi di autenticazione supportati da un pool di utenti, incluso l'accesso tramite provider di identità esterni. Le attività correlate sono trattate nei rispettivi capitoli: gestione degli utenti nella directory, personalizzazione del comportamento con i AWS Lambda trigger e protezione del pool con protezione dalle minacce.



Argomenti

- Funzionalità
- Piani di funzionalità del pool di utenti
- Autenticazione con pool di utenti Amazon Cognito
- Accesso alle risorse dopo aver effettuato correttamente l'accesso

Funzionalità

Di seguito sono riportate le funzionalità dei pool di utenti Amazon Cognito.

Sign-up

I pool di utenti Amazon Cognito dispongono di metodi programmatici, basati sugli utenti e sugli amministratori per aggiungere profili utente al pool di utenti. I pool di utenti di Amazon Cognito supportano i seguenti modelli di iscrizione. Puoi usare qualsiasi combinazione di questi modelli nell'app.

> **Important**
>
> Se attivi la registrazione dell'utente nel pool di utenti, chiunque su Internet può effettuare la registrazione a un account e accedere alle tue app. Non abilitare la registrazione self-service nel pool di utenti a meno che non desideri aprire l'app alla registrazione pubblica. Per modificare questa impostazione, aggiorna l'Self-service iscrizione nel Sign-upmenu sotto Autenticazione nella console del pool di utenti o aggiorna il valore di AllowAdminCreateUserOnlyin una richiesta CreateUserPoolo UpdateUserPoolAPI. Per informazioni sulle funzionalità di sicurezza che puoi configurare nei pool di utenti, consulta Protezione dei pool di utenti con protezione dalle minacce.

1. Gli utenti possono inserire le proprie informazioni nell'app e creare un profilo utente nativo per il pool di utenti. Puoi chiamare operazioni di registrazione delle API per registrare gli utenti nel pool di utenti. Puoi aprire queste operazioni di registrazione a chiunque oppure puoi autorizzarle con un segreto o credenziali del client. AWS
2. Puoi reindirizzare gli utenti a un IdP di terze parti che può essere autorizzato a passare le informazioni ad Amazon Cognito. Amazon Cognito elabora i token ID OIDC, i dati `userInfo` OAuth 2.0 e le asserzioni SAML 2.0 nei profili utente del pool di utenti. Puoi controllare gli attributi che deve ricevere Amazon Cognito in base alle regole di mappatura degli attributi.
3. Puoi saltare l'iscrizione pubblica o federata e creare utenti tu stesso, in base alla tua fonte di dati e al tuo schema, in uno dei seguenti modi:
   - Aggiungi utenti direttamente nella console Amazon Cognito o con l'API.
   - Importa utenti in blocco da un file CSV.
   - Esegui una AWS Lambda funzione just-in-time che cerca ogni nuovo utente in una directory esistente e compila il suo profilo con quei dati.

Dopo la registrazione, è possibile aggiungerli ai gruppi elencati da Amazon Cognito nei token di accesso e ID. Puoi anche collegare i gruppi di pool di utenti ai ruoli IAM quando passi il token ID a un pool di identità.

Argomenti correlati

- Gestione degli utenti nel tuo bacino d'utenza
- Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite
- Esempi di codice per l'utilizzo di Amazon Cognito Identity Provider AWS SDK

Sign-in

Amazon Cognito può essere una directory utente autonoma e gestore dell'identità digitale per l'app. I tuoi utenti possono accedere con pagine di accesso gestite ospitate da Amazon Cognito o con un servizio di autenticazione utente personalizzato tramite l'API dei pool di utenti di Amazon Cognito. Il livello di applicazione alla base del front-end personalizzato può autorizzare le richieste sul back-end con uno qualsiasi dei diversi metodi per confermare le richieste legittime.

Gli utenti possono configurare e accedere con nomi utente e password, password monouso e password monouso per e-mail e SMS. Puoi offrire l'accesso consolidato con directory utente esterne, l'autenticazione a più fattori (MFA) dopo l'accesso, i dispositivi ricordati attendibili e i flussi di autenticazione personalizzati progettati da te.

Per accedere agli utenti con una directory esterna, facoltativamente combinata con la directory utente integrata in Amazon Cognito, puoi aggiungere le seguenti integrazioni.

1. Accedi e importa i dati degli utenti dei clienti con l'accesso social OAuth 2.0. Amazon Cognito supporta l'accesso con Google, Facebook, Amazon e Apple tramite OAuth 2.0.
2. Accedi e importa i dati degli utenti aziendali e scolastici con l'accesso SAML e OIDC. Puoi anche configurare Amazon Cognito per accettare richieste da qualsiasi gestore dell'identità digitale SAML o OpenID Connect (OIDC).
3. Collega i profili utente esterni ai profili utente nativi. Un utente collegato può accedere con un'identità utente di terze parti e ricevere l'accesso assegnato a un utente nella directory integrata.

Argomenti correlati

- Accesso al pool di utenti con provider di identità di terze parti

- Collegamento di utenti federati a un profilo utente esistente

Machine-to-machine autorizzazione

Alcune sessioni non sono un'interazione uomo-macchina. Potrebbe essere necessario un account di servizio in grado di autorizzare una richiesta a un'API tramite un processo automatico. Per generare token di accesso per l'autorizzazione da macchina a macchina con ambiti OAuth 2.0, puoi aggiungere un client di app che genera concessioni di credenziali client dall'endpoint del token o chiamare l'operazione API, che non richiede alcun dominio del pool di utenti. `GetClientToken`

Argomenti correlati

- Ambiti, M2M e server di risorse

Accesso gestito

Se non desideri creare un'interfaccia utente, puoi presentare ai tuoi utenti pagine di accesso gestite personalizzate. L'accesso gestito è un insieme di pagine Web per la registrazione, l'accesso, l'autenticazione a più fattori (MFA) e la reimpostazione della password. Puoi aggiungere l'accesso gestito al tuo dominio esistente o utilizzare un dominio con prefisso in un sottodominio. AWS

L'accesso gestito offre un'alternativa: la classica interfaccia utente ospitata, una versione di prima generazione con un design più semplice e un minor numero di funzionalità. Scegli l'uno o l'altro per il dominio del pool di utenti: non vengono utilizzati insieme. La tabella seguente confronta le due opzioni.

Funzionalità	Interfaccia utente ospitata classica	Login gestito
Versione di branding	Interfaccia utente ospitata (classica)	Login gestito
Personalizzazione	File-based branding (immagine del logo e file di valori CSS)	Editor visivo di branding nella console Amazon Cognito
Piani di funzionalità	Tutti i piani, incluso Lite	Essentials e Plus

Funzionalità	Interfaccia utente ospitata classica	Login gestito
Accesso con passkey	Non disponibile	Disponibilità

> **Note**
>
> La classica interfaccia utente ospitata è disponibile in tutti i piani di funzionalità, incluso Lite. L'accesso gestito e il relativo editor di branding richiedono il piano di funzionalità Essentials o Plus. Per ulteriori informazioni, consulta Piani di funzionalità del pool di utenti.

Per configurare l'accesso gestito, completa questi prerequisiti nell'ordine:

1. Crea un pool di utenti. Consulta Crea una nuova applicazione nella console Amazon Cognito.
2. Aggiungi un dominio del pool di utenti (un dominio con prefisso o un dominio personalizzato). Consulta Configurazione di un dominio di bacino d'utenza.
3. Crea un client per l'app e imposta la versione di branding e l'URL di ritorno. Consulta Accesso gestito dal pool di utenti.

Argomenti correlati

- Accesso gestito dal pool di utenti
- Configurazione di un dominio di bacino d'utenza

Protezione dalle minacce

Proteggi i tuoi utenti dall'indovinazione delle password, dall'acquisizione di account e dal traffico dannoso con l'autenticazione a più fattori (MFA), l'autenticazione adattiva e gli ACL Web. AWS WAF Protezione dei pool di utenti con protezione dalle minacceQueste protezioni sono coperte da.

Esperienza utente personalizzata

Nella maggior parte delle fasi di registrazione, accesso o aggiornamento del profilo, puoi personalizzare il modo in cui Amazon Cognito gestisce la richiesta con trigger Lambda e flussi

di autenticazione personalizzati e conferire all'accesso gestito un aspetto familiare. Consulta Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda.

Monitoraggio e analisi

I pool di utenti di Amazon Cognito registrano le richieste API AWS CloudTrail, pubblicano i parametri delle prestazioni su Amazon CloudWatch Logs e monitorano la consegna dei messaggi e il volume delle richieste. Consulta Accesso ad Amazon Cognito AWS CloudTrail.

Integrazione dei pool di identità di Amazon Cognito

L'altra metà di Amazon Cognito è costituita da pool di identità. I pool di identità forniscono credenziali che autorizzano e monitorano le richieste API inviate dai Servizi AWS tuoi utenti, ad esempio ad Amazon DynamoDB o Amazon S3. Puoi creare policy di accesso basate sull'identità che proteggono i dati in base alla classificazione degli utenti nel pool di utenti. I pool di identità possono anche accettare token e asserzioni SAML 2.0 da un'ampia gamma di provider di identità, a prescindere dall'autenticazione del pool di utenti.

Argomenti correlati

- Accesso Servizi AWS utilizzo di un pool di identità dopo l'accesso
- Pool di identità di Amazon Cognito

Piani di funzionalità del pool di utenti

La comprensione dei costi è un passaggio fondamentale nella preparazione all'implementazione dell'autenticazione dei pool di utenti di Amazon Cognito. Amazon Cognito offre piani di funzionalità per i pool di utenti. Ogni piano ha una serie di funzionalità e un costo mensile per utente attivo. Ogni piano di funzionalità sblocca l'accesso a più funzionalità rispetto a quello precedente.

I pool di utenti dispongono di una varietà di funzionalità che è possibile attivare e disattivare. Ad esempio, puoi attivare l'autenticazione a più fattori (MFA) e disattivare l'accesso con provider di identità di terze parti (). IdPs Alcune modifiche richiedono di cambiare il piano di funzionalità. Le seguenti caratteristiche del tuo pool di utenti determinano il costo che ti verrà AWS fatturato mensilmente per l'utilizzo.

- Le funzionalità che scegli
- Le richieste al secondo che l'applicazione invia all'API dei pool di utenti

- Il numero di utenti con attività di autenticazione, aggiornamento o query in un mese, denominati anche utenti attivi mensili o MAU
- Il numero di utenti attivi mensili da SAML 2.0 o OpenID Connect (OIDC) di terze parti IdPs
- Il numero di client di app e pool di utenti che concedono credenziali client per l'autorizzazione da macchina a macchina

Per le informazioni più aggiornate sui prezzi dei pool di utenti, consulta i prezzi di Amazon Cognito.

Feature-plan le selezioni si applicano a un pool di utenti. Pool di utenti diversi nello stesso Account AWS possono avere selezioni di piano diverse. Non puoi applicare piani di funzionalità separati ai client di app all'interno di un pool di utenti. Il piano di selezione predefinito per i nuovi pool di utenti è Essentials.

Puoi passare da un piano di funzionalità all'altro in qualsiasi momento per soddisfare i requisiti delle tue applicazioni. Alcune modifiche tra i piani richiedono la disattivazione delle funzionalità attive. Per ulteriori informazioni, consulta Disattivazione delle funzionalità per modificare i piani di funzionalità.

Piani di funzionalità del pool di utenti

Lite

Lite è un piano di funzionalità a basso costo per pool di utenti con un numero inferiore di utenti attivi mensili. Questo piano è sufficiente per le directory utente con funzionalità di autenticazione di base. Include funzionalità di accesso e la classica interfaccia utente ospitata, un predecessore più sottile e meno personalizzabile dell'accesso gestito. Molte funzionalità più recenti, come la personalizzazione dei token di accesso e l'autenticazione con chiave di accesso, non sono incluse nel piano Lite.

Essenziali

Essentials dispone di tutte le più recenti funzionalità di autenticazione del pool di utenti. Questo piano aggiunge nuove opzioni alle applicazioni, indipendentemente dal fatto che le pagine di accesso siano gestite con accesso gestito o personalizzate. Essentials dispone di funzionalità di autenticazione avanzate come l'accesso basato sulla scelta e l'autenticazione MFA via e-mail.

Inoltre

Plus include tutte le funzionalità del piano Essentials e aggiunge funzionalità di sicurezza avanzate che proteggono gli utenti. Monitora le richieste di accesso, iscrizione e gestione delle password degli utenti per individuare eventuali indicatori di compromissione. Ad esempio, i pool

di utenti possono rilevare se gli utenti accedono da una posizione inaspettata o utilizzano una password che è stata oggetto di una violazione pubblica.

I pool di utenti con il piano Plus generano registri dei dettagli delle attività degli utenti e delle valutazioni dei rischi. È possibile applicare la propria analisi di utilizzo e sicurezza a questi registri quando li si esporta su servizi esterni.

> **Note**
>
> In precedenza, alcune funzionalità del pool di utenti erano incluse in una struttura tariffaria delle funzionalità di sicurezza avanzate. Le funzionalità incluse in questa struttura sono ora disponibili nel piano Essentials o Plus.

Argomenti

- Seleziona un piano di funzionalità
- Funzionalità suddivise per piano
- Caratteristiche del piano Essentials
- Inoltre, le funzionalità del piano
- Disattivazione delle funzionalità per modificare i piani di funzionalità

Seleziona un piano di funzionalità

Console di gestione AWS

Per scegliere un piano di funzionalità

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Seleziona il menu Impostazioni e consulta la scheda Piani delle funzionalità.
5. Consulta le funzionalità disponibili nei piani Lite, Essentials e Plus.
6. Per modificare il piano, seleziona Passa a Essentials o Passa a Plus. Per passare al piano Lite, scegli Altri piani, quindi Confronta con Lite.
7. Nella schermata successiva, rivedi la tua scelta e seleziona Conferma.

CLI/API/SDK

Le UpdateUserPooloperazioni CreateUserPoole impostano il piano delle funzionalità nel `UserPoolTier` parametro. Quando non specifichi un valore per`UserPoolTier`, il tuo pool di utenti viene impostato come predefinito su. `Essentials` Se è impostato su `AdvancedSecurityMode` `AUDIT` o`ENFORCED`, il livello del pool di utenti deve essere `PLUS` impostato come predefinito `PLUS` quando non è specificato.

Per informazioni sulla sintassi, consulta Esempi. CreateUserPool Vedi anche in CreateUserPool per i collegamenti a questa funzione negli AWS SDK per una varietà di linguaggi di programmazione.

```
"UserPoolTier": "PLUS"
```

In AWS CLI, questa opzione è un `--user-pool-tier` argomento.

```
--user-pool-tier PLUS
```

Per ulteriori informazioni, vedere create-user-pool e update-user-pool nel riferimento ai AWS CLI comandi.

Funzionalità suddivise per piano

Funzionalità e piani nei pool di utenti

Funzionalità	Description	Piano delle funzionalità
Protezione da password non sicure	Controlla le password in chiaro per individuare eventuali indicatori di compromissione in fase di esecuzione	Inoltre
Protezione dai tentativi di accesso malevoli	Controlla le proprietà della sessione per individua re eventuali indicatori di compromissione in fase di esecuzione	Inoltre

Funzionalità	Description	Piano delle funzionalità
Registra e analizza l'attività degli utenti	Genera registri delle proprietà delle sessioni di autentica zione degli utenti e dei punteggi di rischio	Inoltre
Esporta i registri delle attività degli utenti	Invia i registri delle sessioni utente e dei rischi a un file esterno Servizio AWS	Inoltre
Personalizza le pagine di accesso gestite con un editor visuale	Usa un editor visivo nella console Amazon Cognito per applicare branding e stile alle tue pagine di accesso gestite	Essentials + Plus
MFA con codici monouso per e-mail	Richiedi o richiedi agli utenti locali di fornire un fattore di accesso aggiuntivo per i messaggi di posta elettroni ca dopo l'autenticazione del nome utente	Essentials + Plus
Personalizza gli ambiti e le attestazioni dei token di accesso in fase di esecuzione	Usa un trigger Lambda per estendere le capacità di autorizzazione dei token di accesso al pool di utenti	Essentials + Plus
Accesso senza password con codici monouso	Consenti agli utenti di ricevere una password monouso tramite e-mail o SMS come primo fattore di autenticazione	Essentials + Plus
Accesso tramite password con autenticatori FIDO2 hardware o software	Consenti agli utenti di utilizzar e una chiave crittografica memorizzata su un autentica tore FIDO2 come primo fattore di autenticazione	Essentials + Plus

Funzionalità	Description	Piano delle funzionalità
Sign-up e accedi	Esegui operazioni di autentica zione e consenti ai nuovi utenti di registrare un account nella tua applicazione.	Lite+ Essentials + Plus
Gruppi di utenti	Crea raggruppamenti logici di utenti e assegna ruoli IAM predefiniti per le operazioni del pool di identità.	Lite+ Essentials + Plus
Sign-in con provider social, SAML e OIDC	Offri agli utenti la possibilità di accedere direttamente o con il loro provider preferito.	Lite+ Essentials + Plus
OAuth 2. 0/OIDC server di autorizzazione	Agisce come emittente OIDC.	Lite+ Essentials + Plus
Pagine di accesso	Una raccolta ospitata di pagine Web per l'autenti cazione. L'accesso gestito è disponibile nei livelli Essential s e Plus. L'interfaccia utente ospitata classica è disponibile in tutti i livelli di funzionalità.	Lite + Essentials + Plus
Password, autenticazione personalizzata, refresh-token e SRP	Richiedi agli utenti un nome utente e una password nell'applicazione.	Lite+ Essentials + Plus
Machine-to-machine (M2M) con credenziali del cliente	Emetti token di accesso per l'autorizzazione di entità non umane.	Lite+ Essentials + Plus

Funzionalità	Description	Piano delle funzionalità
Autorizzazione API con server di risorse	Emetti token di accesso con ambiti personalizzati che autorizzano l'accesso a sistemi esterni.	Lite+ Essentials + Plus
Importazione utente	Imposta i processi di importazione dai file CSV ed esegui la migrazione tempestiva degli utenti al momento dell'accesso.	Lite + Essentials + Plus
MFA con app di autenticazione e codici monouso SMS	Richiedi o richiedi agli utenti locali di fornire un messaggio SMS aggiuntivo o un fattore di accesso all'app di autenticazione dopo l'autenticazione del nome utente	Lite+ Essentials + Plus
Personalizza gli ambiti e le attestazioni dei token ID in fase di esecuzione	Usa un trigger Lambda per estendere le funzionalità di autenticazione dei token di identità (ID) del pool di utenti	Lite+ Essentials + Plus
Azioni di runtime personalizzate con trigger Lambda	Personalizza il processo di accesso in fase di esecuzione con funzioni Lambda che eseguono azioni esterne e influenzano l'autenticazione	Lite+ Essentials + Plus
Personalizza le pagine di accesso gestite con CSS	Scarica un modello CSS e modifica alcuni stili nelle tue pagine di accesso gestite	Lite+ Essentials + Plus

Caratteristiche del piano Essentials

Il piano di funzionalità Essentials include la maggior parte delle funzionalità migliori e più recenti dei pool di utenti di Amazon Cognito. Passando dal piano Lite a quello Essentials, ottieni nuove funzionalità per le tue pagine di accesso gestite, l'autenticazione a più fattori con password monouso per i messaggi di posta elettronica, una politica di password migliorata e token di accesso personalizzati. Per rimanere aggiornato sulle nuove funzionalità del pool di utenti, scegli il piano Essentials per i tuoi pool di utenti.

Le sezioni che seguono presentano una breve panoramica delle funzionalità che è possibile aggiungere all'applicazione con il piano Essentials. Per informazioni dettagliate, consulta le pagine seguenti.

Risorse aggiuntive

- Personalizzazione dei token di accesso: Trigger Lambda di pre-generazione del token
- MFA via e-mail: MFA per SMS e messaggi e-mail
- Cronologia delle password: Password, ripristino dell'account e politiche relative alle password
- Interfaccia utente migliorata: Applica il branding alle pagine di accesso gestite

Argomenti

- Personalizzazione del token di accesso
- MFA per e-mail
- Prevenzione del riutilizzo delle password
- Accesso gestito, server di accesso e autorizzazione ospitato
- Choice-based authentication

Personalizzazione del token di accesso

I token di accesso al pool di utenti concedono le autorizzazioni alle applicazioni: per accedere a un'API, per recuperare gli attributi utente dall'endpoint UserInfo o per stabilire l'appartenenza al gruppo per un sistema esterno. Negli scenari avanzati, potresti voler aggiungere ai token di accesso predefiniti i dati della directory del pool di utenti con parametri temporanei aggiuntivi che l'applicazione determina in fase di esecuzione. Ad esempio, potresti voler verificare le autorizzazioni API di un utente con Amazon Verified Permissions e modificare di conseguenza gli ambiti nel token di accesso.

Il piano Essentials si aggiunge alle funzioni esistenti di un trigger precedente alla generazione di token. Con i piani di livello inferiore, puoi personalizzare i token ID con reclami, ruoli e appartenenza a gruppi aggiuntivi. Essentials aggiunge nuove versioni dell'evento trigger input che personalizzano le richieste, i ruoli, l'appartenenza ai gruppi e gli ambiti dei token di accesso. La personalizzazione dei token di accesso è disponibile per le credenziali client da macchina a macchina (M2M) con la versione tre dell'evento.

Per personalizzare i token di accesso

1. Seleziona il piano di funzionalità Essentials o Plus.
2. Crea una funzione Lambda per il tuo trigger. Per utilizzare la nostra funzione di esempio, configurala per Node.js.
3. Compila la tua funzione Lambda con il nostro codice di esempio o componine uno tuo. La tua funzione deve elaborare un oggetto di richiesta da Amazon Cognito e restituire le modifiche che desideri includere.
4. Assegna la tua nuova funzione come trigger della versione due o tre prima della generazione di token. Gli eventi della seconda versione personalizzano i token di accesso per le identità degli utenti. La versione tre personalizza i token di accesso per le identità degli utenti e delle macchine.

Ulteriori informazioni

- Personalizzazione del token di accesso
- Come personalizzare i token di accesso nei pool di utenti di Amazon Cognito

MFA per e-mail

I pool di utenti di Amazon Cognito possono essere configurati per utilizzare l'e-mail come secondo fattore dell'autenticazione a più fattori (MFA). Con l'autenticazione a più fattori, Amazon Cognito può inviare agli utenti un'e-mail con un codice di verifica da inserire per completare il processo di autenticazione. Ciò aggiunge un importante ulteriore livello di sicurezza al flusso di accesso degli utenti. Per abilitare la MFA basata su e-mail, il pool di utenti deve essere configurato per utilizzare la configurazione di invio e-mail di Amazon SES anziché la configurazione e-mail predefinita.

Quando l'utente seleziona MFA tramite messaggio e-mail, Amazon Cognito invia un codice di verifica monouso all'indirizzo e-mail registrato dell'utente ogni volta che tenta di accedere. L'utente deve quindi fornire questo codice al tuo pool di utenti per completare il flusso di autenticazione e ottenere

l'accesso. Ciò garantisce che, anche se il nome utente e la password di un utente sono compromessi, deve fornire un fattore aggiuntivo, il codice inviato tramite e-mail, prima di poter accedere alle risorse dell'applicazione.

Per ulteriori informazioni, consulta MFA per SMS e messaggi e-mail. Di seguito è riportata una panoramica su come configurare il pool di utenti e gli utenti per l'MFA della posta elettronica.

Per configurare l'autenticazione a più fattori per la posta elettronica nella console Amazon Cognito

1. Seleziona il piano di funzionalità Essentials o Plus.
2. Nel Sign-inmenu del tuo pool di utenti, modifica Multi-factor l'autenticazione.
3. Scegli il livello di applicazione della MFA che desideri configurare. Con Require MFA, gli utenti dell'API ricevono automaticamente una richiesta di configurazione, conferma e accesso con MFA. Nei pool di utenti che richiedono l'autenticazione a più fattori, l'accesso gestito richiede loro di scegliere e configurare un fattore MFA. Con l'autenticazione a più fattori opzionale, l'applicazione deve offrire agli utenti la possibilità di configurare l'autenticazione a più fattori e impostare le preferenze dell'utente per la MFA via e-mail.
4. In Metodi MFA, seleziona Messaggio di posta elettronica come una delle opzioni.

Ulteriori informazioni

- MFA per SMS e messaggi e-mail

Prevenzione del riutilizzo delle password

Per impostazione predefinita, una policy di password in pool di utenti di Amazon Cognito stabilisce i requisiti di lunghezza e tipo di carattere della password e la scadenza temporanea della password. Il piano Essentials aggiunge la possibilità di applicare la cronologia delle password. Quando un utente tenta di reimpostare la propria password, il gruppo di utenti può impedirgli di impostarla su una password precedente. Per ulteriori informazioni sulla configurazione della politica in materia di password, vedereAggiunta di requisiti password del bacino d'utenza. Di seguito è riportata una panoramica su come configurare il pool di utenti con una politica di cronologia delle password.

Per configurare la cronologia delle password nella console Amazon Cognito

1. Seleziona il piano di funzionalità Essentials o Plus.

2. Nel menu Metodi di autenticazione del tuo pool di utenti, individua la politica delle password e seleziona Modifica.
3. Configura altre opzioni disponibili e imposta un valore per Impedisci l'uso di password precedenti.

Ulteriori informazioni

- Password, ripristino dell'account e politiche relative alle password

Accesso gestito, server di accesso e autorizzazione ospitato

I pool di utenti di Amazon Cognito dispongono di pagine Web opzionali che supportano le seguenti funzioni: un IdP OpenID Connect (OIDC), un fornitore di servizi o un relying party di IdPs terze parti e pagine pubbliche interattive per l'iscrizione e l'accesso. Queste pagine sono denominate collettivamente login gestito. Quando scegli un dominio per il tuo pool di utenti, Amazon Cognito attiva automaticamente queste pagine. Se il piano Lite prevede l'interfaccia utente ospitata, il piano Essentials apre questa versione avanzata delle pagine di registrazione e accesso.

Le pagine di accesso gestite hanno un'interfaccia pulita e aggiornata con più funzionalità e opzioni per personalizzare il marchio e gli stili. Il piano Essentials è il livello di piano più basso che sblocca l'accesso all'accesso gestito.

Per configurare l'accesso gestito nella console Amazon Cognito

1. Dal menu Impostazioni, seleziona il piano di funzionalità Essentials o Plus.
2. Dal menu Dominio, assegna un dominio al tuo pool di utenti e seleziona una versione di branding di Managed login.
3. Dal menu Accesso gestito, nella scheda Stili, scegli Crea uno stile e assegna lo stile a un client di app oppure crea un nuovo client per l'app.

Ulteriori informazioni

- Accesso gestito dal pool di utenti

Choice-based authentication

Il livello Essentials introduce un nuovo flusso di autenticazione per le operazioni di autenticazione nelle operazioni avanzate dell'interfaccia utente e dell' SDK-based API. Questo flusso è un'autenticazione basata sulla scelta. Choice-based l'autenticazione è un metodo in cui l'autenticazione degli utenti non inizia con una dichiarazione sul lato dell'applicazione di un metodo di accesso, ma con una query sui possibili metodi di accesso seguita da una scelta. È possibile configurare il pool di utenti per supportare l'autenticazione basata sulla scelta e sbloccare l'autenticazione con nome utente, password, password e password. Nell'API, questo è il flusso. `USER_AUTH`

Per configurare l'autenticazione basata sulla scelta nella console Amazon Cognito

1. Seleziona il piano di funzionalità Essentials o Plus.
2. Nel Sign-inmenu del tuo pool di utenti, modifica le opzioni per l'accesso basato sulla scelta. Seleziona e configura i metodi di autenticazione che desideri abilitare nell'autenticazione basata sulla scelta.
3. Nel menu Metodi di autenticazione del tuo pool di utenti, modifica la configurazione delle operazioni di accesso.

Ulteriori informazioni

- Autenticazione con pool di utenti Amazon Cognito

Inoltre, le funzionalità del piano

Il piano di funzionalità Plus include funzionalità di sicurezza avanzate per i pool di utenti di Amazon Cognito. Queste funzionalità registrano e analizzano il contesto dell'utente in fase di esecuzione alla ricerca di potenziali problemi di sicurezza in dispositivi, posizioni, dati di richiesta e password. Quindi mitigano i potenziali rischi con risposte automatiche che bloccano o aggiungono protezioni di sicurezza agli account utente. Puoi anche esportare i log di sicurezza su Amazon S3, Amazon Data Firehose o CloudWatch Amazon Logs per ulteriori analisi.

Passando dal piano Essentials al piano Plus, ottieni tutte le funzionalità di Essentials e le funzionalità aggiuntive che seguono. Queste includono il set di opzioni di sicurezza per la protezione dalle minacce, noto anche come funzionalità di sicurezza avanzate. Per configurare i pool di utenti in modo

che si adattino automaticamente alle minacce nel front-end di autenticazione, scegli il piano Plus per i tuoi pool di utenti.

Le sezioni che seguono presentano una breve panoramica delle funzionalità che puoi aggiungere alla tua applicazione con il piano Plus. Per informazioni dettagliate, consulta le pagine seguenti.

Risorse aggiuntive

- Autenticazione adattiva: Utilizzo dell'autenticazione adattiva
- Credenziali compromesse: Utilizzo del rilevamento di credenziali compromesse
- Esportazione del registro: Esportazione dei log dai pool di utenti di Amazon Cognito

Argomenti

- Protezione dalle minacce: autenticazione adattiva
- Protezione dalle minacce: rilevamento delle credenziali compromesse
- Protezione dalle minacce: registrazione delle attività degli utenti

Protezione dalle minacce: autenticazione adattiva

Il piano Plus include una funzionalità di autenticazione adattiva. Quando attivi questa funzionalità, il tuo pool di utenti effettua una valutazione del rischio di ogni sessione di autenticazione utente. In base alle valutazioni di rischio risultanti, puoi bloccare l'autenticazione o attivare l'autenticazione MFA per gli utenti che accedono con un livello di rischio superiore a una soglia da te determinata. Con l'autenticazione adattiva, il pool di utenti e l'applicazione bloccano o configurano automaticamente l'autenticazione MFA per gli utenti i cui account sospetti siano stati attaccati. Puoi anche fornire feedback sulle valutazioni di rischio del tuo pool di utenti per modificare le valutazioni future.

Per configurare l'autenticazione adattiva nella console Amazon Cognito

1. Seleziona il piano di funzionalità Plus.
2. Dal menu Protezione dalle minacce del tuo pool di utenti, modifica l'autenticazione standard e personalizzata in Protezione dalle minacce.
3. Imposta la modalità di applicazione per l'autenticazione standard o personalizzata su Full-function.
4. In Adaptive Authentication, configura le risposte automatiche al rischio per diversi livelli di rischio.

Ulteriori informazioni

- Utilizzo dell'autenticazione adattiva
- Raccolta di dati per la protezione dalle minacce nelle applicazioni

Protezione dalle minacce: rilevamento delle credenziali compromesse

Il piano Plus include una funzione di rilevamento delle credenziali compromesse. Questa funzionalità protegge dall'uso di password non sicure e dalla minaccia di accesso involontario alle applicazioni causata da questa pratica. Quando consenti ai tuoi utenti di accedere con nome utente e password, potrebbero riutilizzare una password che hanno usato altrove. La password potrebbe essere stata divulgata o semplicemente indovinata. Grazie al rilevamento delle credenziali compromesse, il pool di utenti legge le password inviate dagli utenti e le confronta con i database di password. Se l'operazione porta a decidere che la password è probabilmente compromessa, puoi configurare il tuo pool di utenti per bloccare l'accesso e quindi avviare una reimpostazione della password per l'utente nell'applicazione.

Compromised-credentials il rilevamento può reagire alle password non sicure quando nuovi utenti si iscrivono, quando gli utenti esistenti accedono e quando gli utenti tentano di reimpostare le proprie password. Con questa funzionalità, il tuo pool di utenti può prevenire o avvisare in caso di accesso con password non sicure ovunque gli utenti le inseriscano.

Per configurare il rilevamento delle credenziali compromesse nella console Amazon Cognito

1. Seleziona il piano di funzionalità Plus.
2. Dal menu Protezione dalle minacce del tuo pool di utenti, modifica l'autenticazione standard e personalizzata in Protezione dalle minacce.
3. Imposta la modalità di applicazione per l'autenticazione standard o personalizzata su Full-function.
4. In Credenziali compromesse, configura i tipi di operazioni di autenticazione che desideri controllare e la risposta automatica che desideri ricevere dal tuo pool di utenti.

Ulteriori informazioni

- Utilizzo del rilevamento di credenziali compromesse

Protezione dalle minacce: registrazione delle attività degli utenti

Il piano Plus aggiunge una funzionalità di registrazione che fornisce analisi di sicurezza e dettagli sui tentativi di autenticazione degli utenti. Puoi visualizzare le valutazioni dei rischi, gli indirizzi IP degli utenti, gli user agent e altre informazioni sul dispositivo collegato all'applicazione. È possibile agire in base a queste informazioni con le funzionalità integrate di protezione dalle minacce oppure è possibile analizzare i log nei propri sistemi e intraprendere le azioni appropriate. Puoi esportare i log dalla protezione dalle minacce in Amazon S3 CloudWatch , Logs o Amazon DynamoDB.

Per configurare la registrazione delle attività degli utenti nella console Amazon Cognito

1. Seleziona il piano di funzionalità Plus.
2. Dal menu Protezione dalle minacce del tuo pool di utenti, modifica l'autenticazione standard e personalizzata in Protezione dalle minacce.
3. Imposta la modalità di applicazione per l'autenticazione standard o personalizzata su Audit-only. Questa è l'impostazione minima per i registri. Puoi anche attivarlo in Full-functionmodalità e configurare altre funzionalità di protezione dalle minacce.
4. Per esportare i log in un altro account Servizio AWS per l'analisi da parte di terze parti, accedi al menu Log streaming del tuo pool di utenti e configura una destinazione di esportazione.

Ulteriori informazioni

- Esportazione degli eventi di autenticazione degli utenti
- Esportazione dei log dai pool di utenti di Amazon Cognito

Disattivazione delle funzionalità per modificare i piani di funzionalità

I piani di funzionalità aggiungono opzioni di configurazione al pool di utenti. È possibile configurare e utilizzare queste funzionalità solo quando il relativo piano di funzionalità è attivo. Ad esempio, puoi configurare la personalizzazione dei token di accesso nei piani Plus ed Essentials, ma non nel piano Lite. Per disattivare queste funzionalità, è necessario disattivare ogni componente attivo. L'opzione Passa a nel menu Impostazioni della console Amazon Cognito ti avvisa delle funzionalità che devi disattivare prima di poter modificare il tuo piano di funzionalità. In questo capitolo, puoi apprendere le modifiche apportate dalla disattivazione alla configurazione del tuo pool di utenti e come disattivare queste funzionalità singolarmente.

Personalizzazione del token di accesso

Per passare a un piano che non include la personalizzazione del token di accesso, devi rimuovere il trigger Lambda precedente alla generazione del token dal tuo pool di utenti. Per aggiungere un nuovo trigger precedente alla generazione del token senza personalizzare il token di accesso, assegna una nuova funzione al trigger e configuralo per gli eventi. `V1_0` Questi eventi di attivazione della prima versione possono elaborare solo modifiche ai token ID.

Per disattivare manualmente la personalizzazione dei token di accesso, rimuovi il trigger precedente alla generazione del token e aggiungi un nuovo trigger di versione one.

Protezione dalle minacce

Per passare a un piano senza protezione dalle minacce, disattiva tutte le funzionalità dal menu Protezione dalle minacce del tuo pool di utenti.

Esportazione del registro

Per passare a un piano senza esportazione dei log, disattivalo dal menu Log streaming del tuo pool di utenti. Il tuo pool di utenti non genera più registri delle attività degli utenti locali o esportati. Puoi anche inviare una richiesta SetLogDeliveryConfigurationAPI che rimuove qualsiasi configurazione con un `EventSource` valore di. `UserActivity`

MFA per e-mail

Per passare a un piano senza MFA via e-mail, vai al Sign-inmenu del tuo pool di utenti. Modifica Multi-factor l'autenticazione e deseleziona il messaggio e-mail come uno dei metodi MFA disponibili.

Autenticazione con pool di utenti Amazon Cognito

Amazon Cognito include diversi metodi per autenticare gli utenti. Gli utenti possono accedere con password e WebAuthn passkey. Amazon Cognito può inviare loro una password monouso tramite e-mail o SMS. Puoi implementare funzioni Lambda che orchestrano la tua sequenza di sfide e risposte. Si tratta di flussi di autenticazione. Nei flussi di autenticazione, gli utenti forniscono un segreto e Amazon Cognito lo verifica, quindi emette token web JSON (JWT) per le applicazioni da elaborare con le librerie OIDC. Questo capitolo descrive come configurare i pool di utenti e i client delle app per vari flussi di autenticazione in vari ambienti applicativi. Descrive le opzioni per l'uso delle pagine di accesso ospitate di Managed Login e per la creazione di una logica e di un front-end personalizzati in un AWS SDK.

Tutti i pool di utenti, a prescindere che si disponga di un dominio, possono autenticare gli utenti nell'API dei pool di utenti. Se aggiungi un dominio al pool di utenti, puoi utilizzare gli endpoint del pool di utenti. L'API dei pool di utenti supporta una vasta gamma di modelli di autorizzazione e flussi di richiesta per le richieste API.

Per verificare l'identità degli utenti, Amazon Cognito supporta flussi di autenticazione che incorporano tipi di sfida oltre a password come e-mail e SMS, password monouso e passkey.

Prima di configurare i dettagli, scegli l'approccio di autenticazione adatto alla tua applicazione. Il seguente aiuto decisionale riassume i due punti di ingresso e, all'interno di un'applicazione personalizzata, i due flussi di accesso iniziali. Ogni opzione si collega alla relativa sezione dettagliata.

Vuoi che Amazon Cognito ospiti le pagine di accesso o crei il tuo front-end?

- Se desideri il percorso con il minor sforzo possibile, utilizza l'accesso gestito. Amazon Cognito ospita le pagine di accesso, disconnessione e reimpostazione della password e l'applicazione elabora il risultato con una libreria relying-party OpenID Connect (OIDC). L'accesso gestito presenta automaticamente i metodi di autenticazione consentiti dal pool di utenti e dalla configurazione del client dell'app.

- Se desideri creare la tua interfaccia utente e controllare la logica di accesso, usa un AWS SDK. La tua applicazione chiama direttamente l'API dei pool di utenti e tu implementi la logica challenge-response.

In un'applicazione personalizzata, raccogli il metodo di accesso dall'utente o lo dichiari in anticipo?

- Scegli l'autenticazione basata sulla scelta (il `USER_AUTH` flusso) quando desideri offrire agli utenti un elenco di metodi di accesso disponibili e consentire loro di selezionarne uno. Questo flusso è l'unico modo per offrire l'accesso senza password e password.

- Scegli l'autenticazione basata sul client quando l'applicazione sa già come l'utente accederà e dichiara il flusso in anticipo, ad esempio. `USER_SRP_AUTH` Client-basedl'autenticazione è l'unico modo per utilizzare l'autenticazione personalizzata e i flussi di token di aggiornamento.

Prerequisiti per l'autenticazione basata sulla scelta

- Choice-based l'autenticazione si applica sia all'accesso gestito che agli SDK. AWS Per utilizzarla, aggiungila `ALLOW_USER_AUTH` ai flussi di autenticazione consentiti del client dell'app. I metodi senza password e passkey richiedono inoltre un dominio di accesso gestito e un piano di funzionalità superiore al livello Lite (Essentials o Plus).

Argomenti

- Application-specific impostazioni con i client dell'app
- Implementa flussi di autenticazione
- Cose da sapere sull'autenticazione con pool di utenti
- Un esempio di sessione di autenticazione
- Configura i metodi di autenticazione per l'accesso gestito
- Gestisci i metodi di autenticazione in AWS SDK
- Flusso di autenticazione
- Modelli di autorizzazione per l'autenticazione tramite API e SDK
- Accesso al pool di utenti con provider di identità di terze parti

Application-specific impostazioni con i client dell'app

Un client dell'app del pool di utenti è una configurazione all'interno di un pool di utenti che interagisce con un'applicazione per dispositivi mobili o Web che esegue l'autenticazione con Amazon Cognito. I client di app possono chiamare operazioni API autenticate e non autenticate, nonché leggere o modificare alcuni o tutti gli attributi degli utenti. L'app deve identificarsi con il client dell'app durante le operazioni di registrazione, accesso e gestione delle password dimenticate. Queste richieste API devono includere identificazione automatica con un ID del client dell'app e l'autorizzazione con un segreto client opzionale. È responsabilità dell'utente proteggere eventuali ID o segreti del client dell'app in modo che solo le app client autorizzate possano chiamare queste operazioni non autenticate. Inoltre, se configuri l'app per firmare richieste API autenticate con AWS credenziali, devi proteggere le tue credenziali dall'ispezione da parte degli utenti.

È possibile creare più client app per un bacino d'utenza. Un client dell'app potrebbe essere collegato alla piattaforma di codice di un'app o a un tenant separato nel pool di utenti. Ad esempio, puoi creare un'app per un'applicazione lato server e un'app Android diversa. Ogni app ha il proprio ID client.

È possibile applicare le impostazioni per le seguenti funzionalità del pool di utenti a livello di client dell'app:

1. Analisi
2. Accesso gestito IdPs, tipi di concessione, URL di callback e personalizzazione
3. Server di risorse e ambiti personalizzati
4. Protezione dalle minacce

5. Attribuisci le autorizzazioni di lettura e scrittura
6. Scadenza e revoca del token
7. Flussi di autenticazione

Tipi di client di app

Quando crei un client dell'app in Amazon Cognito, puoi precompilare le opzioni in base ai tipi di client OAuth standard client pubblico e client riservato. Configurare un client riservato con un client secret. Per ulteriori informazioni sui tipi di client, vedere IETF RFC 6749 #2.1.

Client pubblico

Un client pubblico viene eseguito in un browser o su un dispositivo mobile. Poiché non dispone di risorse affidabili sul lato server, non ha un client secret.

Client riservato

Un client riservato dispone di risorse lato server che possono essere attendibili e di un client secret per operazioni API non autenticate L'app potrebbe essere eseguita come demone o script di shell sul server di back-end.

Client secret

Un client secret, o client password, è una stringa fissa che l'app deve utilizzare in tutte le richieste API che invia al client dell'app. Il client dell'app deve avere un client secret per eseguire la concessione di `client_credentials`. Per ulteriori informazioni, consulta IETF RFC 6749 #2.3.1.

Ogni client dell'app può avere fino a due segreti alla volta, abilitando la rotazione segreta senza tempi di inattività. Quando crei un client per app, puoi consentire ad Amazon Cognito di generare un valore segreto o fornire un valore segreto personalizzato. Non puoi modificare il client secret una volta che hai creato un'app. Puoi aggiungere un secondo segreto con l'operazione AddUserPoolClientSecretAPI per ruotare i segreti. Quando aggiungi un segreto, puoi consentire ad Amazon Cognito di generare un valore segreto o fornire un valore segreto personalizzato. Per eliminare un segreto, utilizza l'operazione DeleteUserPoolClientSecretAPI. Non è possibile eliminare l'unico segreto associato a un client dell'app. Puoi anche eliminare un'app per bloccare l'accesso da app che usano l'ID client di quell'app.

> **Note**
>
> La console Amazon Cognito crea client di app con segreti client quando selezioni l'applicazione Web tradizionale e Machine-to-machine le opzioni di applicazione per il tipo di applicazione. Scegli una di queste opzioni per generare un client secret oppure crea il client programmaticamente con CreateUserPoolClient e impostalo su. `GenerateSecret` `true`

> **Important**
>
> Non è possibile modificare se un client dell'app ha un client segreto dopo aver creato il client dell'app. L'UpdateUserPoolClientoperazione non può aggiungere o rimuovere un segreto client. Per modificare questa configurazione, crea un nuovo client per l'app. Non è necessario ricreare il pool di utenti e gli altri client dell'app non sono interessati.

È possibile utilizzare un client riservato e un client secret con un'app pubblica. Usa un CloudFront proxy Amazon per aggiungere un proxy `SECRET_HASH` in transito. Per ulteriori informazioni, consulta Proteggere i client pubblici per Amazon Cognito utilizzando un CloudFront proxy Amazon sul AWS blog.

Token web JSON

I client dell'app Amazon Cognito possono emettere token web JSON (JWT) dei seguenti tipi.

Token di identità (ID)

: Una dichiarazione verificabile attestante che l'utente è autenticato dal pool di utenti. OpenID Connect (OIDC) ha aggiunto la specifica del token ID agli standard dei token di accesso e aggiornamento definiti da OAuth 2.0. Il token ID contiene informazioni sull'identità, come gli attributi utente, che l'app può utilizzare per creare un profilo utente e fornire risorse. Per ulteriori informazioni, consulta Comprendere il token di identità (ID).

Token di accesso

: Una dichiarazione verificabile dei diritti di accesso dell'utente. Il token di accesso contiene gli ambiti, una funzionalità di OIDC e OAuth 2.0. L'app può presentare gli ambiti delle risorse di back-end e dimostrare che il pool di utenti ha autorizzato un utente o una macchina ad accedere ai dati

da un'API o ai propri dati utente. Un token di accesso con ambiti personalizzati, spesso ottenuto tramite una concessione di credenziali client M2M, autorizza l'accesso a un server di risorse. Per ulteriori informazioni, consulta Comprendere il token di accesso.

Token di aggiornamento

Una dichiarazione crittografata di autenticazione iniziale che l'app può presentare al pool di utenti quando i token dell'utente scadono. Una richiesta refresh-token restituisce token di accesso e ID nuovi e non scaduti. Per ulteriori informazioni, consulta Aggiorna i token.

Puoi impostare la scadenza di questi token per ogni app client dal menu App client del tuo pool di utenti nella console Amazon Cognito.

Termini del client dell'app

I termini seguenti sono proprietà disponibili dei client di app nella console Amazon Cognito.

URL di callback consentiti

Un URL di callback indica dove l'utente viene reindirizzato dopo un accesso riuscito. Scegli almeno un URL di callback. L'URL di callback deve:

- Deve essere un URI assoluto.
- Deve essere preregistrato con un client.
- Non deve includere un componente di frammento.

Consulta OAuth 2.0 - redirection endpoint (OAuth 2.0 - Endpoint di reindirizzamento).

Amazon Cognito richiede `HTTPS` su `HTTP` tranne che per `http://localhost` a solo scopo di test.

Sono supportati anche URL di callback come `myapp://example`.

URL di disconnessione consentiti

Un URL di disconnessione indica dove l'utente deve essere reindirizzato dopo la disconnessione.

Autorizzazioni di lettura e scrittura dell'attributo

Il tuo pool di utenti potrebbe avere molti clienti, ognuno con il proprio client di app e. IdPs Puoi configurare il client dell'app in modo che abbia accesso in lettura e scrittura solo agli attributi

utente pertinenti all'app. In casi come l'autorizzazione da macchina a macchina (M2M), non puoi concedere l'accesso a nessuno degli attributi utente.

Considerazioni sulla configurazione delle autorizzazioni di lettura e scrittura degli attributi

- Quando crei un client per app e non personalizzi le autorizzazioni di lettura e scrittura degli attributi, Amazon Cognito concede le autorizzazioni di lettura e scrittura a tutti gli attributi del pool di utenti.
- Puoi concedere l'accesso in scrittura ad attributi personalizzati non modificabili. Il client dell'app può scrivere valori in attributi immutabili quando crei o registri un utente. Dopodiché, non puoi scrivere valori in alcun attributo personalizzato non modificabile per l'utente.
- I client dell'app devono avere accesso in scrittura agli attributi richiesti nel pool di utenti. La console Amazon Cognito imposta automaticamente gli attributi richiesti come scrivibili.
- Non puoi consentire a un client dell'app di avere accesso in scrittura a `email_verified` o a `phone_number_verified`. Un amministratore del pool di utenti può modificare questi valori. Un utente può modificare il valore di questi attributi solo tramite la verifica degli attributi.

Flusso di autenticazione

I metodi di accesso consentiti dal client dell'app. La tua app può supportare l'autenticazione con nome utente e password, OTP per messaggi e-mail e SMS, autenticatori con chiave di accesso, autenticazione personalizzata con trigger Lambda e aggiornamento dei token. Come migliore pratica di sicurezza, utilizza l'autenticazione SRP per l'autenticazione di nome utente e password in applicazioni personalizzate.

Ambiti personalizzati

Un ambito personalizzato è quello che viene definito per il proprio server di risorse nella scheda Resource Servers (Server di Risorse). Il formato è*`resource-server-identifier`*/*`scope`*. Consulta Ambiti, M2M e server di risorse.

URI di reindirizzamento predefinito

Sostituisce il `redirect_uri` parametro nelle richieste di autenticazione per gli utenti con terze parti. IdPs Configura questa impostazione del client dell'app con il `DefaultRedirectURI` parametro di una richiesta CreateUserPoolClientoUpdateUserPoolClientAPI. Questo URL deve inoltre essere un membro del client `CallbackURLs` for your app. Amazon Cognito reindirizza le sessioni autenticate a questo URL quando:

1. Al client dell'app è assegnato un provider di identità e sono definiti più URL di callback. Il pool di utenti reindirizza le richieste di autenticazione al server di autorizzazione all'URI di reindirizzamento predefinito quando non includono un parametro. `redirect_uri`

2. Il client dell'app ha un provider di identità assegnato e un URL di callback definito. In questo scenario non è necessario definire un URL di callback predefinito. Le richieste che non includono un `redirect_uri` parametro reindirizzano all'unico URL di callback disponibile.

Fornitori di identità

Puoi scegliere alcuni o tutti i provider di identità esterni del tuo pool di utenti (IdPs) per autenticare i tuoi utenti. Il client di app può inoltre autenticare solo gli utenti locali del pool di utenti. Quando aggiungi un IdP al client dell'app, puoi generare link di autorizzazione all'IdP e visualizzarli nella pagina di accesso gestito. Puoi assegnarne più di uno IdPs, ma devi assegnarne almeno uno. Per ulteriori informazioni sull'utilizzo di dispositivi esterni IdPs, vedere. Accesso al pool di utenti con provider di identità di terze parti

Ambiti OpenID Connect

Scegli uno o più dei seguenti ambiti `OAuth` per specificare i privilegi di accesso che possono essere richiesti per i token d'accesso.

- L'ambito `openid` dichiara che desideri recuperare un token ID e l'ID univoco di un utente. Richiede inoltre tutti o alcuni attributi utente, a seconda degli ambiti aggiuntivi nella richiesta. Amazon Cognito non restituisce un token ID a meno che non venga richiesto l'ambito `openid`. L'ambito `openid` autorizza le attestazioni dei token ID strutturali come la scadenza e l'ID chiave e determina gli attributi utente che ricevi in una risposta da Endpoint UserInfo.
  - Quando `openid` è l'unico ambito richiesto, Amazon Cognito popola il token ID con tutti gli attributi utente che il client dell'app corrente è in grado di leggere. La risposta `userInfo` a un token di accesso con questo ambito restituisce tutti gli attributi utente.
  - Quando richiedi `openid` con altri ambiti come `phone`, `email` o `profile`, il token ID e `userInfo` restituiscono l'ID univoco dell'utente e gli attributi definiti dagli ambiti aggiuntivi.
- L'ambito `phone` permette l'accesso alle richieste `phone_number` e `phone_number_verified`. Questo ambito può essere richiesto solo con l'ambito `openid`.
- L'ambito `email` permette l'accesso alle richieste `email` e `email_verified`. Questo ambito può essere richiesto solo con l'ambito `openid`.
- L'`aws.cognito.signin.user.admin`ambito consente l'accesso alle operazioni API dei pool di utenti di Amazon Cognito che richiedono token di accesso, come e. UpdateUserAttributesVerifyUserAttribute
- L'ambito `profile` permette l'accesso a tutti gli attributi dell'utente negli ID token che sono leggibili dal client. Questo ambito può essere richiesto solo con l'ambito `openid`.

Per ulteriori informazioni sugli ambiti, consulta la lista degli ambiti OIDC standard.

Tipi di concessione OAuth

Una concessione OAuth è un metodo di autenticazione che recupera i token del pool di utenti. Amazon Cognito supporta i seguenti tipi di concessioni. Per integrare queste concessioni OAuth nella tua app, devi aggiungere un dominio al tuo pool di utenti.

Concessione codice autorizzazione

La concessione del codice di autorizzazione genera un codice che l'app può scambiare con i token del pool di utenti con Endpoint Token. Durante lo scambio di un codice di autorizzazione, l'app riceve token ID, di accesso e di aggiornamento. Questo flusso OAuth, come la concessione implicita, avviene nei browser degli utenti. La concessione di un codice di autorizzazione è la concessione più sicura offerta da Amazon Cognito, poiché i token non sono visibili nelle sessioni degli utenti. Invece, l'app genera la richiesta che restituisce i token e può memorizzarli nella cache in uno spazio di archiviazione protetto. Per ulteriori informazioni, vedere Codice di autorizzazione in IETF RFC 6749 #1 .3.1

> **Note**
>
> Come best practice di sicurezza nelle app per client pubblici, attiva solo il flusso OAuth di concessione del codice di autorizzazione e implementa Proof Key for Code Exchange (PKCE) per limitare lo scambio di token. Con PKCE, un client può scambiare un codice di autorizzazione solo dopo aver fornito all'endpoint del token lo stesso segreto presentato nella richiesta di autenticazione originale. Per ulteriori informazioni, consulta IETF RFC 7636.

Implicit grant (Concessione implicita)

La concessione implicita fornisce un token di accesso e ID, ma non un token di aggiornamento, alla sessione del browser dell'utente direttamente da Endpoint Authorize. Una concessione implicita rimuove il requisito di una richiesta separata all'endpoint del token, ma non è compatibile con PKCE e non restituisce token di aggiornamento. Questa concessione supporta scenari di test e architetture di app che non consentono di completare la concessione di codici di autorizzazione. Per ulteriori informazioni, consulta Concessione implicita in IETF RFC 6749 #1.3.2. Puoi attivare sia la concessione del codice di autorizzazione sia la concessione implicita in un client dell’app e usare entrambe in base alle esigenze.

Concessione credenziali del client

La concessione delle credenziali del client è per le comunicazioni da macchina a macchina (M2M). Authorization-code e le concessioni implicite emettono token a utenti umani autenticati. Le credenziali del client concedono l'autorizzazione basata sull'ambito da un sistema non interattivo a un'API. L'app può richiedere le credenziali del client direttamente dall'endpoint del token e ricevere un token di accesso. Per ulteriori informazioni, consulta Credenziali del client in IETF RFC 6749 #1 .3.4. È possibile attivare la concessione di credenziali client solo nei client di app che dispongono di un client secret e che non supportano il codice di autorizzazione o le concessioni implicite. In alternativa, che non richiede alcun dominio di pool di utenti, un client di app con un client segreto e un flusso di `ALLOW_CLIENT_TOKEN_AUTH` autenticazione può ottenere token di accesso M2M con l'operazione API. GetClientToken

> **Note**
>
> Poiché il flusso di credenziali del client non viene richiamato come un utente, questa concessione può solo aggiungere ambiti personalizzati per accedere ai token. Un ambito personalizzato è quello che viene definito per il proprio server di risorse. Gli ambiti predefiniti come `openid` e `profile` non si applicano agli utenti non umani.
> Poiché i token ID sono una convalida degli attributi utente, non sono rilevanti per la comunicazione M2M e la concessione delle credenziali del client non li rilascia. Consulta Ambiti, M2M e server di risorse.

Le concessioni di credenziali al cliente aggiungono costi alla fattura. AWS Per ulteriori informazioni, consultare Prezzi di Amazon Cognito.

Creazione di un client dell'app

Console di gestione AWS

Come creare un client dell'app (console)

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo. Entrambe le opzioni richiedono di configurare un client dell'app con impostazioni specifiche dell'applicazione.
4. Scegliete un tipo di applicazione che rifletta l'architettura dell'applicazione.

> **Important**
>
> La selezione del tipo di applicazione imposta anche i valori predefiniti per le altre impostazioni del client dell'app, incluso se Amazon Cognito genera un segreto client. L'applicazione Web tradizionale e i tipi di Machine-to-machine applicazione generano un segreto client. Per ulteriori informazioni, consulta Tipi di client di app.

5. Assegna un nome alla tua applicazione con un identificatore intuitivo.
6. Inserisci un URL di ritorno.
7. Scegli Create app client (Crea client dell'app). Puoi modificare le opzioni avanzate dopo aver creato il client dell'app.
8. Amazon Cognito ti riporta ai dettagli del client dell'app. Per accedere al codice di esempio per la tua applicazione, seleziona una piattaforma dalla scheda Guida alla configurazione rapida.

AWS CLI

```
aws cognito-idp create-user-pool-client --user-pool-id MyUserPoolID --client-
name myApp
```

> **Note**
>
> Utilizza il formato JSON per URL di callback e disconnessione per impedire a CLI di trattarli come file di parametri remoti:
>
> ```
> --callback-urls "["https://example.com"]"
> --logout-urls "["https://example.com"]"
> ```

Per ulteriori informazioni, consulta il riferimento ai AWS CLI comandi: create-user-pool-client

Amazon Cognito user pools API

Genera una richiesta API. CreateUserPoolClient È necessario specificare un valore per tutti i parametri che non si desidera impostare su un valore predefinito.

Aggiornamento di un client di app per pool di utenti (AWS CLI and AWS API)

Al AWS CLI, inserisci il seguente comando:

```
aws cognito-idp update-user-pool-client --user-pool-id  "MyUserPoolID" --client-id
 "MyAppClientID" --allowed-o-auth-flows-user-pool-client --allowed-o-auth-flows "code"
 "implicit" --allowed-o-auth-scopes "openid" --callback-urls "["https://example.com"]"
 --supported-identity-providers "["MySAMLIdP", "LoginWithAmazon"]"
```

Se il comando ha esito positivo, AWS CLI restituisce una conferma:

```
{
    "UserPoolClient": {
        "ClientId": "MyClientID",
        "SupportedIdentityProviders": [
            "LoginWithAmazon",
            "MySAMLIdP"
        ],
        "CallbackURLs": [
            "https://example.com"
        ],
        "AllowedOAuthScopes": [
            "openid"
        ],
        "ClientName": "Example",
        "AllowedOAuthFlows": [
            "implicit",
            "code"
        ],
        "RefreshTokenValidity": 30,
        "AuthSessionValidity": 3,
        "CreationDate": 1524628110.29,
        "AllowedOAuthFlowsUserPoolClient": true,
        "UserPoolId": "MyUserPoolID",
        "LastModifiedDate": 1530055177.553
    }
}
```

Per ulteriori informazioni, vedere il riferimento ai AWS CLI comandi: update-user-pool-client.

AWS API: UpdateUserPoolClient

Ottenere informazioni su un client di app per pool di utenti (AWS CLI and AWS API)

```
aws cognito-idp describe-user-pool-client --user-pool-id MyUserPoolID --client-
id MyClientID
```

Per ulteriori informazioni, vedere il riferimento al AWS CLI comando: describe-user-pool-client.

AWS API: DescribeUserPoolClient

Elenco di tutte le informazioni sul client dell'app in un pool di utenti (AWS CLI and AWS API)

```
aws cognito-idp list-user-pool-clients --user-pool-id "MyUserPoolID" --max-results 3
```

Per ulteriori informazioni, vedere il riferimento al AWS CLI comando: list-user-pool-clients.

AWS API: ListUserPoolClients

Eliminazione di un client di app per pool di utenti (AWS CLI and AWS API)

```
aws cognito-idp delete-user-pool-client --user-pool-id "MyUserPoolID" --client-id
 "MyAppClientID"
```

Per ulteriori informazioni, vedere il riferimento al AWS CLI comando: delete-user-pool-client

AWS API: DeleteUserPoolClient

Implementa flussi di autenticazione

Che tu stia implementando l'accesso gestito o un front-end applicativo personalizzato con un AWS SDK per l'autenticazione, devi configurare il client dell'app per i tipi di autenticazione che desideri implementare. Le seguenti informazioni descrivono la configurazione dei flussi di autenticazione nei client dell'app e nell'applicazione.

App client supported flows

> Puoi configurare i flussi supportati per i client delle tue app nella console Amazon Cognito o con l'API in un AWS SDK. Dopo aver configurato il client dell'app per supportare questi flussi, puoi distribuirli nell'applicazione.

La procedura seguente configura i flussi di autenticazione disponibili per un client di app con la console Amazon Cognito.

Per configurare un client di app per i flussi di autenticazione (console)

1. Accedi AWS e accedi alla console dei pool di utenti di Amazon Cognito. Scegli un pool di utenti o creane uno nuovo.
2. Nella configurazione del pool di utenti, seleziona il menu App clients. Scegli un client per l'app o creane uno nuovo.
3. In Informazioni sul client dell'app, seleziona Modifica.
4. In Flussi client dell'app, scegli i flussi di autenticazione che desideri supportare.

Per configurare un client di app per i flussi di autenticazione (API/SDK)

Per configurare i flussi di autenticazione disponibili per un client di app con l'API Amazon Cognito, imposta il valore di `ExplicitAuthFlows` in a CreateUserPoolClientor UpdateUserPoolClientrequest. Di seguito è riportato un esempio che fornisce una password remota sicura (SRP) e l'autenticazione basata sulla scelta a un client.

```
"ExplicitAuthFlows": [
   "ALLOW_USER_AUTH",
   "ALLOW_USER_SRP_AUTH"
]
```

Quando configuri i flussi supportati dal client dell'app, puoi specificare le seguenti opzioni e valori API.

Supporto per il flusso dei client dell'app

Flusso di autenticazione	Compatibilità	Console	"Hello, World!"
Choice-basedautenticazione	Server-side, lato client	Seleziona un tipo di autenticazione al momento dell'accesso	`ALLOW_USER_AUTH`

Flusso di autentica zione	Compatibilità	Console	"Hello, World!"
Sign-in con password persistenti	Client-side	Accedi con nome utente e password	`ALLOW_USER_PASSWORD_AUTH`
Sign-in con password persistenti e payload sicuro	Server-side, lato client	Accedi con una password remota sicura (SRP)	`ALLOW_USER_SRP_AUTH`
Aggiorna i token	Server-side, lato client	Ottieni nuovi token utente dalle sessioni autenticate esistenti	`ALLOW_REFRESH_TOKEN_AUTH`
Server-side autentica zione	Server-side	Accedi con credenzia li amministrative lato server	`ALLOW_ADMIN_USER_PASSWORD_AUTH`
Autenticazione personalizzata	Server-side e applicazioni personali zzate lato client. Non compatibile con l'accesso gestito.	Accedi con flussi di autenticazione personalizzati dai trigger Lambda	`ALLOW_CUSTOM_AUTH`

Implement flows in your application

L'accesso gestito rende automaticamente disponibili le opzioni di autenticazione configurate nelle pagine di accesso. Nelle applicazioni personalizzate, avvia l'autenticazione con una dichiarazione del flusso iniziale.

- Per scegliere da un elenco di opzioni di flusso per un utente, dichiara l'autenticazione basata sulla scelta con il flusso. `USER_AUTH` Questo flusso dispone di metodi di autenticazione che non sono disponibili nei flussi di autenticazione basati su client, ad esempio l'autenticazione con chiave di accesso e l'autenticazione senza password.
- Per scegliere in anticipo il flusso di autenticazione, dichiara l'autenticazione basata sul client con qualsiasi altro flusso disponibile nel client dell'app.

Quando accedi utenti, il corpo della AdminInitiateAuthrichiesta InitiateAutho della richiesta deve includere un `AuthFlow` parametro.

Choice-based autenticazione:

```
"AuthFlow": "USER_AUTH"
```

Client-based autenticazione con SRP:

```
"AuthFlow": "USER_SRP_AUTH"
```

Cose da sapere sull'autenticazione con pool di utenti

Prendi in considerazione le seguenti informazioni nella progettazione del tuo modello di autenticazione con i pool di utenti di Amazon Cognito.

Flussi di autenticazione nell'accesso gestito e nell'interfaccia utente ospitata

L'accesso gestito offre più opzioni di autenticazione rispetto alla classica interfaccia utente ospitata. Ad esempio, gli utenti possono eseguire l'autenticazione senza password e con chiave di accesso solo nell'accesso gestito.

I flussi di autenticazione personalizzati sono disponibili solo nell'autenticazione SDK AWS

Non puoi eseguire flussi di autenticazione personalizzati o autenticazioni personalizzate con trigger Lambda, con l'accesso gestito o la classica interfaccia utente ospitata. L'autenticazione personalizzata è disponibile nell'autenticazione con AWS SDK.

Accesso gestito per l'accesso con provider di identità esterno (IdP)

Non puoi far accedere gli utenti tramite l'autenticazione di terze parti con IdPs AWS gli SDK. È necessario implementare l'accesso gestito o la classica interfaccia utente ospitata, reindirizzare IdPs e quindi elaborare l'oggetto di autenticazione risultante con le librerie OIDC nell'applicazione. Per ulteriori informazioni sull'accesso gestito, vedere. Accesso gestito dal pool di utenti

Effetto dell'autenticazione senza password su altre funzionalità utente

L'attivazione dell'accesso senza password con password o passkey monouso nel pool di utenti e nel client dell'app ha un effetto sulla creazione e sulla migrazione degli utenti. Quando l'accesso senza password è attivo:

1. Gli amministratori possono creare utenti senza password. Il modello di messaggio di invito predefinito viene modificato e non include più il segnaposto per la `{###}` password. Per ulteriori informazioni, consulta Creazione di account utente come amministratore.
2. Per quanto riguarda SDK-based SignUple operazioni, gli utenti non sono tenuti a fornire una password al momento della registrazione. L'accesso gestito e l'interfaccia utente ospitata richiedono una password nella pagina di registrazione, anche se è consentita l'autenticazione senza password. Per ulteriori informazioni, consulta Registrazione e conferma degli account utente.
3. Gli utenti importati da un file CSV possono accedere immediatamente con opzioni senza password, senza reimpostare la password, se i loro attributi includono un indirizzo e-mail o un numero di telefono per un'opzione di accesso senza password disponibile. Per ulteriori informazioni, consulta Importazione di utenti nel bacino d'utenza da un file CSV.
4. L'autenticazione senza password non richiama il trigger Lambda per la migrazione degli utenti.
5. Gli utenti che accedono con una password monouso (OTP) non possono aggiungere un fattore di autenticazione a più fattori (MFA) alla sessione. Le passkey con verifica utente possono soddisfare i requisiti MFA se configurate con. `MULTI_FACTOR_WITH_USER_VERIFICATION`

Gli URL delle parti che si affidano a Passkey non possono essere presenti nell'elenco pubblico dei suffissi

Puoi usare nomi di dominio di tua proprietà, ad esempio`www.example.com`, come ID relying party (RP) nella configurazione della tua passkey. Questa configurazione è pensata per supportare applicazioni personalizzate che vengono eseguite su domini di tua proprietà. L'elenco dei suffissi pubblici, o PSL, contiene domini di alto livello protetti. Amazon Cognito restituisce un errore quando tenti di impostare l'URL RP su un dominio su PSL.

Durata del flusso della sessione di autenticazione

A seconda delle funzionalità del tuo pool di utenti, puoi rispondere a diverse sfide `RespondToAuthChallenge` prima `InitiateAuth` e prima che l'app recuperi i token da Amazon Cognito. Amazon Cognito include una stringa di sessione nella risposta a ciascuna richiesta. Per combinare le richieste API in un flusso di autenticazione, includere la stringa di sessione della risposta alla richiesta precedente in ogni richiesta successiva. Per impostazione predefinita, gli utenti hanno a disposizione tre minuti per completare ogni verifica prima della scadenza della stringa della sessione. Per modificare questo periodo, modificare il client dell'app Authentication flow session duration (Durata della sessione del flusso di autenticazione). La procedura seguente descrive come modificare questa impostazione nella configurazione del client dell'app.

> **Note**
>
> Le impostazioni Durata della sessione del flusso di autenticazione si applicano all'autenticazione con l'API dei pool di utenti Amazon Cognito. L'accesso gestito imposta la durata della sessione a 3 minuti per l'autenticazione a più fattori e a 8 minuti per la modifica forzata della password durante il primo accesso dell'utente finale con password temporanea.

Amazon Cognito console

Per configurare la durata della sessione del flusso di autenticazione del client dell'app (Console di gestione AWS)

1. Nella scheda App integration (Integrazione app) del pool di utenti, seleziona il nome del client dell'app nel container App clients and analytics (Client di app e analisi dei dati).
2. Scegli Modifica nel container Informazioni sul client dell'app.
3. Modifica il valore della durata della sessione del flusso di autenticazione sulla durata di validità desiderata, in minuti, per i codici MFA SMS ed e-mail. Ciò modifica anche il tempo che ogni utente ha a disposizione per completare qualsiasi richiesta di autenticazione nel client dell'app.
4. Scegli Save changes (Salva modifiche).

User pools API

Configurare la durata della sessione del flusso di autenticazione del client dell'app (API Amazon Cognito)

1. Prepara una richiesta `UpdateUserPoolClient` con le impostazioni di un pool di utenti esistente in base a una richiesta `DescribeUserPoolClient`. La richiesta `UpdateUserPoolClient` deve includere tutte le proprietà client dell'app esistenti.
2. Cambia il valore del parametro `AuthSessionValidity` impostando la durata di validità desiderata, espressa in minuti, per i codici dell'autenticazione MFA con SMS. Ciò modifica anche il tempo che ogni utente ha a disposizione per completare qualsiasi richiesta di autenticazione nel client dell'app.

Per ulteriori informazioni sui client di app, consulta Application-specific impostazioni con i client dell'app.

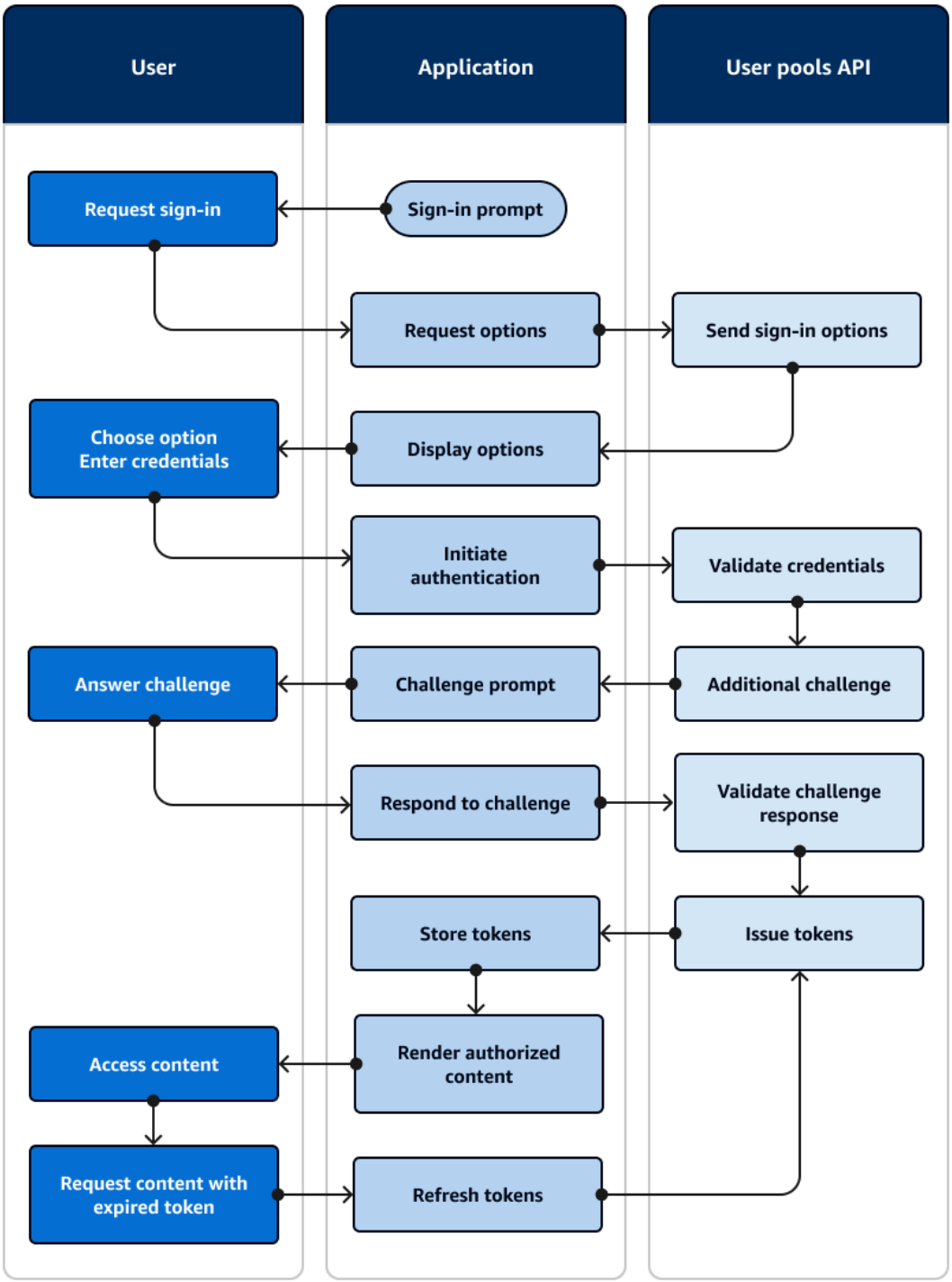
Comportamento di blocco in caso di tentativi di accesso non riusciti

Dopo cinque tentativi di accesso falliti con la password di un utente, indipendentemente dal fatto che vengano richiesti con operazioni IAM-authorized API o non autenticate, Amazon Cognito blocca l'utente per un secondo. La durata del blocco quindi raddoppia dopo ogni ulteriore tentativo non riuscito, fino a un massimo di circa 15 minuti.

I tentativi effettuati durante un periodo di blocco generano un'eccezione `Password attempts exceeded` e non influiscono sulla durata dei periodi di blocco successivi. Per un numero cumulativo di tentativi di accesso non riusciti n, ad esclusione delle eccezioni `Password attempts exceeded`, Amazon Cognito blocca l'utente per 2^(n-5) secondi. Per ripristinare lo stato iniziale n=0 del blocco, l'utente deve effettuare un accesso riuscito dopo la scadenza del periodo di blocco oppure non deve iniziare alcun tentativo di accesso per 15 minuti consecutivi in qualsiasi momento dopo il blocco. Questo comportamento è soggetto a modifiche. Questo comportamento non si applica alle sfide personalizzate a meno che non eseguano anche l'autenticazione basata su password.

Un esempio di sessione di autenticazione

Il diagramma seguente e la guida dettagliata illustrano uno scenario tipico in cui un utente accede a un'applicazione. L'applicazione di esempio presenta all'utente diverse opzioni di accesso. Ne selezionano una inserendo le proprie credenziali, forniscono un fattore di autenticazione aggiuntivo e accedono.



Immagina un'applicazione con una pagina di accesso in cui gli utenti possono accedere con nome utente e password, richiedere un codice monouso in un messaggio e-mail o scegliere un'opzione di impronta digitale.

1. Sign-in prompt: l'applicazione mostra una schermata iniziale con un pulsante di accesso.
2. Richiedi l'accesso: l'utente seleziona Accedi. Da un cookie o da una cache, l'applicazione recupera il nome utente o chiede loro di inserirlo.
3. Opzioni di richiesta: l'applicazione richiede le opzioni di accesso dell'utente con una richiesta `InitiateAuth` API con il `USER_AUTH` flusso, richiedendo i metodi di accesso disponibili per l'utente.
4. Invia opzioni di accesso: Amazon Cognito risponde `PASSWORD` con`EMAIL_OTP`, e. `WEB_AUTHN` La risposta include un identificatore di sessione da riprodurre nella risposta successiva.
5. Opzioni di visualizzazione: l'applicazione mostra gli elementi dell'interfaccia utente che consentono all'utente di inserire nome utente e password, ottenere un codice monouso o scansionare l'impronta digitale.
6. Scegli option/Enter le credenziali: l'utente inserisce nome utente e password.
7. Avvia l'autenticazione: l'applicazione fornisce le informazioni di accesso dell'utente con una richiesta `RespondToAuthChallenge` API che conferma l'accesso con nome utente e password e fornisce il nome utente e la password.
8. Convalida delle credenziali: Amazon Cognito conferma le credenziali dell'utente.
9. Sfida aggiuntiva: l'utente dispone di un'autenticazione a più fattori configurata con un'app di autenticazione. Amazon Cognito restituisce una `SOFTWARE_TOKEN_MFA` sfida.
10. Richiesta di sfida: l'applicazione visualizza un modulo che richiede una password monouso (TOTP) basata sul tempo dall'app di autenticazione dell'utente.
11. Sfida di risposta: l'utente invia il TOTP.
12. Rispondi alla sfida: in un'altra `RespondToAuthChallenge` richiesta, l'applicazione fornisce il TOTP dell'utente.
13. Convalida la risposta alla sfida: Amazon Cognito conferma il codice dell'utente e determina che il pool di utenti è configurato per non inviare ulteriori sfide all'utente corrente.
14. Emissione di token: Amazon Cognito restituisce ID, accede e aggiorna i token web JSON (JWT). L'autenticazione iniziale dell'utente è completa.
15. Archivia token: l'applicazione memorizza nella cache i token dell'utente in modo che possa fare riferimento ai dati dell'utente, autorizzare l'accesso alle risorse e aggiornare i token quando scadono.
16. Esegui il rendering dei contenuti autorizzati: l'applicazione determina l'accesso dell'utente alle risorse in base alla sua identità e ai suoi ruoli e fornisce il contenuto dell'applicazione.
17. Accesso ai contenuti: l'utente ha effettuato l'accesso e inizia a utilizzare l'applicazione.

18. Richiedi contenuti con token scaduto: Successivamente, l'utente richiede una risorsa che richiede l'autorizzazione. Il token memorizzato nella cache dell'utente è scaduto.
19. Token di aggiornamento: l'applicazione effettua una **`InitiateAuth`** richiesta con il token di aggiornamento salvato dall'utente.
20. Emetti token: Amazon Cognito restituisce un nuovo ID e accede a JWT. La sessione dell'utente viene aggiornata in modo sicuro senza ulteriori richieste di credenziali.

È possibile utilizzare i AWS Lambda trigger per personalizzare il modo in cui gli utenti effettuano l'autenticazione. Questi trigger generano e verificano le proprie richieste come parte del flusso di autenticazione.

Puoi anche utilizzare il flusso di autenticazione amministrativa per server back-end sicuri. Puoi utilizzare il flusso di autenticazione della migrazione degli utenti per rendere possibile la migrazione degli utenti senza richiedere agli utenti di reimpostare le proprie password.

Configura i metodi di autenticazione per l'accesso gestito

È possibile richiamare pagine di accesso gestite, un front-end web per l'autenticazione del pool di utenti, quando si desidera che gli utenti accedano, si disconnettano o reimpostino la password. In questo modello, l'applicazione importa le librerie OIDC per elaborare i tentativi di autenticazione basati su browser con pagine di accesso gestite da pool di utenti. Le forme di autenticazione disponibili per gli utenti dipendono dalla configurazione del pool di utenti e del client dell'app. Implementa il `ALLOW_USER_AUTH` flusso nel client dell'app e Amazon Cognito richiede agli utenti di selezionare un metodo di accesso tra le opzioni disponibili. Implementa `ALLOW_USER_PASSWORD_AUTH` e assegna un provider SAML e le tue pagine di accesso richiedono agli utenti la possibilità di inserire nome utente e password o di connettersi con il proprio IdP.

La console dei pool di utenti di Amazon Cognito può aiutarti a configurare l'autenticazione di accesso gestita per la tua applicazione. Quando crei un nuovo pool di utenti, specifica la piattaforma per cui stai sviluppando e la console ti fornisce esempi di implementazione delle librerie OIDC e OAuth con codice iniziale per implementare i flussi di accesso e disconnessione. Puoi creare un accesso gestito con molte implementazioni di relying-party OIDC. Ti consigliamo di utilizzare librerie relying party OIDC certificate, ove possibile. Per ulteriori informazioni, consulta Crea una nuova applicazione nella console Amazon Cognito.

In genere, le librerie relying party OIDC controllano periodicamente l'`.well-known/openid-configuration`endpoint del pool di utenti per determinare gli URL dell'emittente, come l'endpoint

del token e l'endpoint di autorizzazione. Come best practice, implementa questo comportamento di rilevamento automatico laddove ne hai la possibilità. La configurazione manuale degli endpoint dell'emittente introduce un potenziale di errore. Ad esempio, è possibile modificare il dominio del pool di utenti. Il percorso di destinazione `openid-configuration` non è collegato al dominio del pool di utenti, pertanto le applicazioni che individuano automaticamente gli endpoint del servizio rilevano automaticamente la modifica del dominio.

Impostazioni del pool di utenti per l'accesso gestito

Potresti voler consentire l'accesso con più provider per la tua applicazione oppure potresti voler utilizzare Amazon Cognito come directory utente indipendente. Potresti anche voler raccogliere gli attributi utente, configurare e richiedere l'MFA o richiedere gli indirizzi e-mail come nomi utente. Non puoi modificare direttamente i campi nell'accesso gestito e nell'interfaccia utente ospitata. Al contrario, la configurazione del pool di utenti imposta automaticamente la gestione dei flussi di autenticazione di accesso gestito.

I seguenti elementi di configurazione del pool di utenti determinano i metodi di autenticazione che Amazon Cognito presenta agli utenti nell'accesso gestito e nell'interfaccia utente ospitata.

User pool options (Sign-in menu)

Le seguenti opzioni si trovano nel Sign-inmenu di un pool di utenti nella console Amazon Cognito.

Opzioni di accesso al pool di utenti di Cognito

Dispone di opzioni per i nomi utente. L'accesso gestito e le pagine dell'interfaccia utente ospitate accettano solo nomi utente nei formati selezionati. Quando, ad esempio, configuri un pool di utenti con Email come unica opzione di accesso, le pagine di accesso gestite accettano solo nomi utente in formato e-mail.

Attributi obbligatori

Quando imposti un attributo come obbligatorio nel tuo pool di utenti, l'accesso gestito richiede agli utenti un valore per quell'attributo al momento della registrazione.

Opzioni per l’accesso basato sulla scelta

Dispone di impostazioni per i metodi di autenticazione in. Choice-based authentication Qui puoi attivare o disattivare metodi di autenticazione come passkey e passwordless. Questi metodi sono disponibili solo per i pool di utenti con domini di accesso gestiti e piani di funzionalità superiori al livello Lite.

Multi-factor authentication

L'accesso gestito e l'interfaccia utente ospitata gestiscono le operazioni di registrazione e autenticazione per l'MFA. Quando l'MFA è richiesta nel tuo pool di utenti, le pagine di accesso richiedono automaticamente agli utenti di impostare il proprio fattore aggiuntivo. Richiedono inoltre agli utenti che dispongono di una configurazione MFA di completare l'autenticazione con un codice MFA. Quando la MFA è disattivata o facoltativa nel tuo pool di utenti, nelle pagine di accesso non viene richiesto di configurare la MFA.

Ripristino dell'account utente

L'impostazione di ripristino dell'account in modalità self-service del pool di utenti determina se le pagine di accesso visualizzano un link in cui gli utenti possono reimpostare la password.

User pool options (Domain menu)

Le seguenti opzioni si trovano nel menu Dominio di un pool di utenti nella console Amazon Cognito.

Dominio

La scelta del dominio del pool di utenti imposta il percorso del collegamento che gli utenti aprono quando richiami i loro browser per l'autenticazione.

Versione di branding

La scelta di una versione di branding determina se nel dominio del pool di utenti viene visualizzato l'accesso gestito o l'interfaccia utente ospitata.

User pool options (Social and external providers menu)

L'opzione seguente si trova nel menu Social e provider esterni di un pool di utenti nella console Amazon Cognito.

Provider

I provider di identità (IdPs) che aggiungi al tuo pool di utenti possono essere lasciati attivi o inattivi per ogni client di app nel pool di utenti.

App client options

Le seguenti opzioni si trovano nel menu App client di un pool di utenti nella console Amazon Cognito. Per esaminare queste opzioni, seleziona un client dell'app dall'elenco.

Guida rapida alla configurazione

La guida rapida all'installazione contiene esempi di codice per una varietà di ambienti di sviluppo. Contengono le librerie necessarie per integrare l'autenticazione di accesso gestito con l'applicazione.

Informazioni sul client dell'app

Modifica questa configurazione in modo che sia assegnata IdPs all'applicazione rappresentata dal client dell'app corrente. Nelle pagine di accesso gestito, Amazon Cognito mostra le scelte per gli utenti. Queste scelte sono determinate dai metodi e dall'IdP assegnati. Ad esempio, se assegni un IdP SAML 2.0 denominato `MySAML` e un accesso al pool di utenti locali, le pagine di accesso gestite visualizzano le istruzioni del metodo di autenticazione e un pulsante per. `MySAML`

Impostazioni di autenticazione

Modifica questa configurazione per impostare i metodi di autenticazione per l'applicazione. Nelle pagine di accesso gestito, Amazon Cognito mostra le scelte per gli utenti. Queste scelte sono determinate dalla disponibilità del pool di utenti come IdP e dai metodi assegnati. Ad esempio, se assegni l'`ALLOW_USER_AUTH`autenticazione basata sulla scelta, le pagine di accesso gestite mostrano le opzioni disponibili come l'immissione di un indirizzo e-mail e l'accesso con una passkey. Le pagine di accesso gestite visualizzano anche i pulsanti relativi alle impostazioni assegnate. IdPs

Pagine di accesso

Imposta l'effetto visivo delle pagine interattive con l'utente di accesso gestito o dell'interfaccia utente ospitata con le opzioni disponibili in questa scheda. Per ulteriori informazioni, consulta Applica il branding alle pagine di accesso gestite.

Gestisci i metodi di autenticazione in AWS SDK

Gli utenti dei pool di utenti di Amazon Cognito possono accedere con una varietà di opzioni o fattori di accesso iniziali. Per alcuni fattori, gli utenti possono seguire l'autenticazione a più fattori (MFA). Questi primi fattori includono nome utente e password, password monouso, passkey e autenticazione personalizzata. Per ulteriori informazioni, consulta Flusso di autenticazione. Se l'applicazione dispone di componenti dell'interfaccia utente integrati e importa un modulo AWS SDK, è necessario creare una logica applicativa per l'autenticazione. È necessario scegliere uno dei due metodi principali e, tra tale metodo, i meccanismi di autenticazione che si desidera implementare.

È possibile implementare l'autenticazione basata su client laddove l'applicazione o il client dichiari in anticipo il tipo di autenticazione. L'altra opzione è l'autenticazione basata sulla scelta, in cui l'app

raccoglie un nome utente e richiede i tipi di autenticazione disponibili per gli utenti. Puoi implementare questi modelli insieme nella stessa applicazione o suddividerli tra i client dell'app, in base alle tue esigenze. Ogni metodo ha caratteristiche uniche, ad esempio l'autenticazione personalizzata in modalità client e l'autenticazione senza password in quella basata sulla scelta.

Nelle applicazioni personalizzate che eseguono l'autenticazione con l'implementazione AWS SDK dell'API dei pool di utenti, è necessario strutturare le richieste API in modo da allinearle alla configurazione del pool di utenti, alla configurazione del client dell'app e alle preferenze lato client. Una `InitiateAuth` sessione che inizia con un of inizia l'autenticazione basata sulla scelta. `AuthFlow` `USER_AUTH` Amazon Cognito risponde alla tua API proponendo un metodo di autenticazione preferito o un elenco di scelte. Una sessione che inizia con `AuthFlow` o `CUSTOM_AUTH` passa direttamente all'autenticazione personalizzata con trigger Lambda.

Alcuni metodi di autenticazione sono limitati a uno dei due tipi di flusso e alcuni metodi sono disponibili in entrambi.

Argomenti

- Choice-based authentication
- Client-based authentication
- Step-up autenticazione con il flusso USER_AUTH

Choice-based authentication

L'applicazione può richiedere i seguenti metodi di autenticazione nell'autenticazione basata sulla scelta. Dichiara queste opzioni nel `PREFERRED_CHALLENGE` parametro di InitiateAutho o AdminInitiateAuthnel parametro di o. `ChallengeName` RespondToAuthChallengeAdminRespondToAuthChallenge

> **Important**
>
> Choice-based l'accesso ha due prerequisiti. Il client dell'app deve attivare il flusso di **`ALLOW_USER_AUTH`** autenticazione e i metodi senza password e passkey (**`EMAIL_OTP`**,**`SMS_OTP`**,**`WEB_AUTHN`**) richiedono un pool di utenti con il piano di funzionalità Essentials o Plus, poiché non sono disponibili nel livello Lite. `PASSWORD`(incluso SRP) è disponibile in tutti i piani di funzionalità. Se agli utenti non viene visualizzato un metodo basato sulla scelta, conferma sia il flusso tra app e client che il piano delle funzionalità prima di procedere con la risoluzione dei problemi.

1. `EMAIL_OTP` e `SMS_OTP`

   Accesso senza password con codici monouso

2. `WEB_AUTHN`

   Accesso senza password con passkey WebAuthn

3. `PASSWORD`

   Sign-in con password persistenti

   Sign-in con password persistenti e payload sicuro

   MFA dopo l'accesso

Per esaminare queste opzioni nel loro contesto API, consulta in. `ChallengeName` RespondToAuthChallenge

Choice-based l'accesso presenta una sfida in risposta alla tua richiesta iniziale. Questa sfida verifica che l'opzione richiesta sia disponibile o fornisce un elenco di scelte disponibili. L'applicazione può mostrare queste scelte agli utenti, che quindi inseriscono le credenziali per il metodo di accesso preferito e procedono con l'autenticazione nelle risposte alla sfida.

Nel flusso di autenticazione sono disponibili le seguenti opzioni basate sulla scelta. Tutte le richieste di questo tipo richiedono che l'app raccolga prima un nome utente o lo recuperi da una cache.

1. Richiedi opzioni `USERNAME` solo con `AuthParameters` of. Amazon Cognito restituisce una `SELECT_CHALLENGE` sfida. Da lì, l'applicazione può richiedere all'utente di selezionare una sfida e restituire questa risposta al tuo pool di utenti.
2. Richiedi una sfida preferita con `AuthParameters` of `PREFERRED_CHALLENGE` e i parametri della tua sfida preferita, se presenti. Ad esempio, se richiedi una `PREFERRED_CHALLENGE` di`PASSWORD_SRP`, devi includere anche`SRP_A`. Se l'utente, il pool di utenti e il client dell'app sono tutti configurati per la sfida preferita, Amazon Cognito risponde con la fase successiva di tale sfida, ad esempio `PASSWORD_VERIFIER` nel `PASSWORD_SRP` flusso o CodeDeliveryDetailsnei flussi and. `EMAIL_OTP` `SMS_OTP` Se la sfida preferita non è disponibile, Amazon Cognito risponde con `SELECT_CHALLENGE` un elenco di sfide disponibili.
3. Effettua prima l'accesso degli utenti, quindi richiedi le opzioni di autenticazione in base alla loro scelta. Una GetUserAuthFactorsrichiesta con il token di accesso di un utente che ha effettuato l'accesso restituisce i fattori di autenticazione basati sulla scelta disponibili e le relative

impostazioni MFA. Con questa opzione, un utente può accedere prima con nome utente e password, quindi attivare una diversa forma di autenticazione. Puoi anche utilizzare questa operazione per verificare le opzioni aggiuntive per un utente che ha effettuato l'accesso con una sfida preferita.

Per configurare il client dell'app per l'autenticazione basata sulla scelta, aggiungilo `ALLOW_USER_AUTH` ai flussi di autenticazione consentiti. È inoltre necessario scegliere i fattori basati sulla scelta che si desidera consentire nella configurazione del pool di utenti. Il processo seguente illustra come scegliere i fattori di autenticazione basati sulla scelta.

Amazon Cognito console

Per configurare le opzioni di autenticazione basate sulla scelta in un pool di utenti

1. Accedi AWS e accedi alla console dei pool di utenti di Amazon Cognito. Scegli un pool di utenti o creane uno nuovo.
2. Nella configurazione del pool di utenti, seleziona il Sign-inmenu. Individua Opzioni per l'accesso basato sulla scelta e scegli Modifica.
3. L'opzione Password è sempre disponibile. Ciò include i `PASSWORD_SRP` flussi `PASSWORD` e. Seleziona le scelte aggiuntive che desideri aggiungere alle opzioni degli utenti. È possibile aggiungere una Passkey per`WEB_AUTHN`, una password monouso per un messaggio di posta elettronica e una password monouso per `EMAIL_OTP` un messaggio SMS. `SMS_OTP`
4. Scegli Save changes (Salva modifiche).

API/SDK

Il seguente corpo parziale CreateUserPoolo di UpdateUserPoolrichiesta configura tutte le opzioni disponibili per l'autenticazione basata sulla scelta.

```
"Policies": {
    "SignInPolicy": {
        "AllowedFirstAuthFactors": [
            "PASSWORD",
            "WEB_AUTHN",
            "EMAIL_OTP",
            "SMS_OTP"
        ]
    }
```

```
},
```

Client-based authentication

Client-based l'autenticazione supporta i seguenti flussi di autenticazione. Dichiara queste opzioni nel `AuthFlow` parametro InitiateAutho AdminInitiateAuth.

1. `USER_PASSWORD_AUTH` e `ADMIN_USER_PASSWORD_AUTH`

   Sign-in con password persistenti

   MFA dopo l'accesso

   Questo flusso di autenticazione è equivalente `PASSWORD` all'autenticazione basata sulla scelta.
2. `USER_SRP_AUTH`

   Sign-in con password persistenti e payload sicuro

   MFA dopo l'accesso

   Questo flusso di autenticazione è equivalente all'`PASSWORD_SRP`autenticazione basata sulla scelta.
3. `REFRESH_TOKEN_AUTH`

   Aggiorna i token

   Questo flusso di autenticazione è disponibile solo nell'autenticazione basata sul client.
4. `CUSTOM_AUTH`

   Autenticazione personalizzata

   Questo flusso di autenticazione è disponibile solo nell'autenticazione basata su client.

Con l'autenticazione basata su client, Amazon Cognito presuppone che tu abbia determinato in che modo l'utente desidera autenticarsi prima di iniziare i flussi di autenticazione. La logica di determinazione del fattore di accesso che un utente desidera fornire deve essere determinata con impostazioni predefinite o istruzioni personalizzate, quindi dichiarata nella prima richiesta al tuo pool di utenti. La `InitiateAuth` richiesta dichiara un accesso `AuthFlow` che corrisponde direttamente a una delle opzioni elencate, ad esempio. `USER_SRP_AUTH` Con questa dichiarazione, la richiesta

include anche i parametri per iniziare l'autenticazione, ad esempio USERNAMESECRET_HASH, e. SRP_A Amazon Cognito potrebbe dare seguito a questa richiesta con ulteriori sfide, come l'SRP o PASSWORD_VERIFIER l'accesso tramite password con SOFTWARE_TOKEN_MFA TOTP MFA.

Per configurare il client dell'app per l'autenticazione basata su client, aggiungi qualsiasi flusso di autenticazione diverso ALLOW_USER_AUTH dai flussi di autenticazione consentiti. Alcuni esempi sono ALLOW_USER_PASSWORD_AUTHALLOW_CUSTOM_AUTH,ALLOW_REFRESH_TOKEN_AUTH. Per consentire i flussi di autenticazione basati sul client, non è richiesta alcuna configurazione aggiuntiva del pool di utenti.

Step-up autenticazione con il flusso USER_AUTH

Richiedete l'autenticazione avanzata nel USER_AUTH flusso esistente con le operazioni and. InitiateAuthRespondToAuthChallenge Non ci sono nuove operazioni API. Le operazioni lato server corrispondenti supportano gli stessi parametri AdminInitiateAuthe comportamenti di step-up e AdminInitiateAuth possono anche AdminRespondToAuthChallenge, facoltativamente, utilizzare un token di accesso. Questo flusso funziona sia quando un utente dispone di un token ID esistente sia quando un utente si autentica verso un livello target per la prima volta. Per ulteriori informazioni sul flusso, consulta. Un esempio di sessione di autenticazione

Per ulteriori informazioni sui livelli ACR e sui valori AMR, vedere. the section called “Livelli di autenticazione con attestazioni ACR e AMR”

> **Note**
>
> Step-up l'autenticazione richiede il piano di funzionalità Essentials o Plus.

Passate i parametri step-up nella AuthParameters mappa della vostra InitiateAuth (oAdminInitiateAuth) richiesta: la nuova TARGET_ACR_VALUES eMAX_AGE, insieme a quella esistenteUSERNAME, ACCESS_TOKEN e. PREFERRED_CHALLENGE Per la definizione e i vincoli di ogni parametro, consulta InitiateAuthe AdminInitiateAuthnell'API Reference dei pool di utenti di Amazon Cognito.

La tabella seguente descrive come Amazon Cognito risponde alle combinazioni di questi parametri.

Comportamento di autenticazione avanzato USER_AUTH

Parameters	Comportamento
`TARGET_AC R_VALUES eACCESS_TOKEN` , con `MAX_AGE` soddisfazione	Amazon Cognito consente all'utente di passare dai fattori correnti del token al livello di priorità soddisfacibile con la massima priorità.
`TARGET_AC R_VALUES eACCESS_TOKEN` , con «non soddisfatto» `MAX_AGE`	Amazon Cognito autentica l'utente da zero verso il livello target.
`TARGET_ACR_VALUES` senza `ACCESS_TOKEN`	Amazon Cognito autentica l'utente da zero verso il livello target.
Nessuno dei due parametri	Il `USER_AUTH` flusso si comporta come se non fosse necessaria l'autenticazione avanzata.

Tieni presente il seguente comportamento quando progetti un flusso di autenticazione graduale.

- Step-up l'autenticazione è tutto o niente. Se il raggiungimento del livello obiettivo richiede più sfide, ad esempio una password e un TOTP per il livello 4, e l'utente fallisce qualsiasi sfida, il tentativo di avanzamento fallisce completamente. Amazon Cognito non emette token intermedi.
- Se il token presentato soddisfa o supera già il livello richiesto, Amazon Cognito restituisce un errore invece di emettere un nuovo token. Ciò impedisce alle applicazioni di utilizzare l'autenticazione step-up come meccanismo di aggiornamento dei token.
- L'utente deve già disporre dei fattori per soddisfare il livello richiesto. Se l'utente non ha configurato alcun fattore per soddisfare il livello obiettivo richiesto, Amazon Cognito restituisce un errore. Step-upl'autenticazione non richiede all'utente di impostare nuovi fattori.
- L'MFA è un pavimento. Se le impostazioni MFA dell'utente o del pool di utenti richiedono l'autenticazione MFA, l'accesso iniziale applica e richiede la verifica MFA anche quando il livello di destinazione è inferiore e altrimenti non richiederebbe l'MFA. L'MFA richiesta dalla protezione dalle minacce (funzionalità di sicurezza avanzate) viene applicata anche all'accesso iniziale, anche quando il livello ACR di destinazione richiesto non richiederebbe altrimenti l'MFA. `TARGET_ACR_VALUES`non può sopprimere un prompt MFA richiesto. Al massimo, l'obiettivo può influenzare quale dei diversi fattori MFA abilitati richiede Amazon Cognito. Se una richiesta step-up si rivolge a un livello inferiore, ad esempio il livello 1, per un utente che deve utilizzare l'MFA,

Amazon Cognito applica silenziosamente l'MFA richiesta. Migliora l'utente e non restituisce alcun errore.

La richiesta di autenticazione avanzata `TARGET_ACR_VALUES` richiede il piano di funzionalità Essentials o Plus. Per ulteriori informazioni, consulta Requisiti del piano di funzionalità.

Flusso di autenticazione

Il processo di autenticazione con i pool di utenti di Amazon Cognito può essere meglio descritto come un flusso in cui gli utenti effettuano una scelta iniziale, inviano credenziali e rispondono a sfide aggiuntive. Quando implementi l'autenticazione di accesso gestito nella tua applicazione, Amazon Cognito gestisce il flusso di queste richieste e sfide. Quando implementi flussi con un AWS SDK nel backend dell'applicazione, devi costruire la logica delle richieste, richiedere input agli utenti e rispondere alle sfide.

In qualità di amministratore dell'applicazione, le caratteristiche utente, i requisiti di sicurezza e il modello di autorizzazione aiutano a determinare come consentire agli utenti di accedere. Ponetevi le seguenti domande.

- Voglio consentire agli utenti di accedere con le credenziali di altri provider di identità (IdPs)?
- Un nome utente e una password sono una prova di identità sufficiente?
- Le mie richieste di autenticazione per l'autenticazione nome utente e password potrebbero essere intercettate? Voglio che la mia applicazione trasmetta password o negozii l'autenticazione utilizzando hash e sali?
- Voglio consentire agli utenti di ignorare l'immissione della password e di ricevere una password monouso per l'accesso?
- Voglio consentire agli utenti di accedere con un'impronta digitale, un volto o una chiave di sicurezza hardware?
- Quando voglio richiedere l'autenticazione a più fattori (MFA), se possibile?
- Voglio rendere persistenti le sessioni utente senza richiedere nuovamente le credenziali?
- Voglio estendere il mio modello di autorizzazione oltre le funzionalità integrate di Amazon Cognito?

Quando avrai le risposte a queste domande, potrai imparare come attivare le funzionalità pertinenti e implementarle nelle richieste di autenticazione effettuate dalla tua applicazione.

Dopo aver configurato i flussi di accesso per un utente, puoi verificarne lo stato attuale per l'MFA e i fattori di autenticazione basati sulla scelta con le richieste all'operazione API. GetUserAuthFactors Questa operazione richiede l'autorizzazione con il token di accesso di un utente che ha effettuato l'accesso. Restituisce i fattori di autenticazione dell'utente e le impostazioni MFA.

Argomenti

- Sign-in con terze parti IdPs
- Sign-in con password persistenti
- Sign-in con password persistenti e payload sicuro
- Accesso senza password con codici monouso
- Accesso senza password con passkey WebAuthn
- MFA dopo l'accesso
- Aggiorna i token
- Autenticazione personalizzata
- Flusso di autenticazione per la migrazione degli utenti

Sign-in con terze parti IdPs

I pool di utenti di Amazon Cognito fungono da broker intermedio di sessioni di autenticazione tra servizi IdPs come Sign in with Apple, Login with Amazon e OpenID Connect (OIDC). Questo processo è anche chiamato accesso federato o autenticazione federata. L'autenticazione federata non utilizza nessuno dei flussi di autenticazione che puoi integrare nel client dell'app. Al contrario, assegni il pool di utenti configurato IdPs al client dell'app. L'accesso federato avviene quando gli utenti selezionano il proprio IdP nell'accesso gestito o l'applicazione richiama una sessione con un reindirizzamento alla pagina di accesso dell'IdP.

Con l'accesso federato, deleghi i fattori di autenticazione primari e MFA all'IdP dell'utente. Amazon Cognito non aggiunge gli altri flussi avanzati di questa sezione a un utente federato a meno che non li colleghi a un utente locale. Gli utenti federati non collegati hanno nomi utente, ma sono un archivio di dati di attributi mappati che in genere non vengono utilizzati per l'accesso indipendentemente dal flusso basato sul browser.

Risorse per l'implementazione

- Accesso al pool di utenti con provider di identità di terze parti

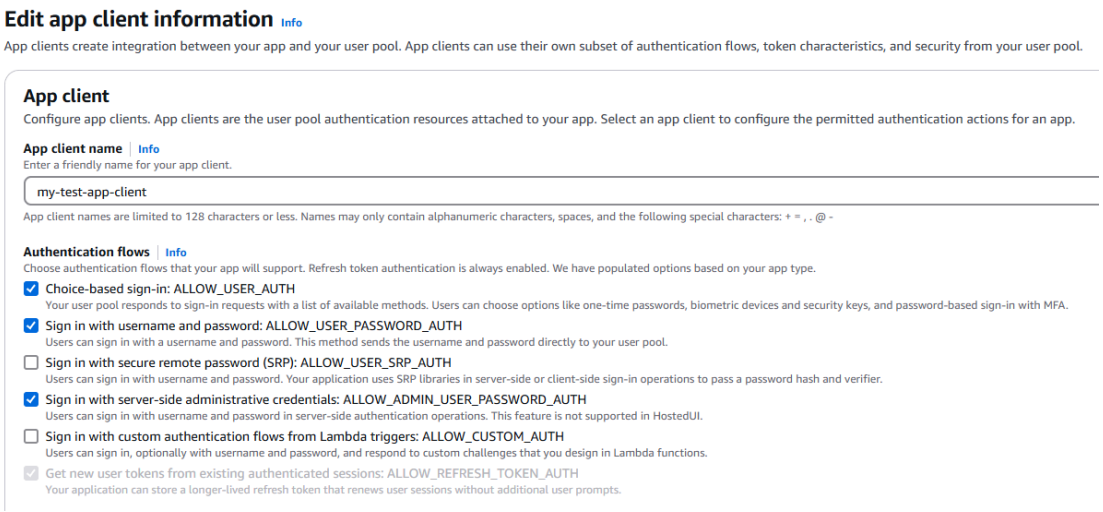
Sign-in con password persistenti

Nei pool di utenti di Amazon Cognito, ogni utente ha un nome utente. Potrebbe trattarsi di un numero di telefono, un indirizzo e-mail o un identificatore scelto o fornito dall'amministratore. Gli utenti di questo tipo possono accedere con il proprio nome utente e password e, facoltativamente, fornire l'autenticazione a più fattori. I pool di utenti possono eseguire l'accesso con nome utente e password con operazioni pubbliche o IAM-authorized API e metodi SDK. L'applicazione può inviare direttamente la password al pool di utenti per l'autenticazione. Il tuo pool di utenti risponde con ulteriori sfide o con i token web JSON (JWT) che sono il risultato di un'autenticazione riuscita.

Activate password sign-in

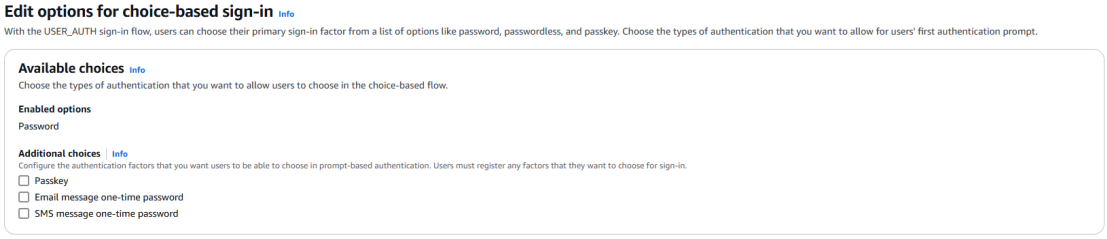
Per attivare l'autenticazione basata sul client con nome utente e password, configura il client dell'app in modo da consentirla. Nella console Amazon Cognito, accedi al menu App client in Applicazioni nella configurazione del pool di utenti. Per consentire l'accesso tramite password semplice per un'app mobile o nativa sul lato client, modifica un client dell'app e scegli Accedi con nome utente e password: ALLOW_USER_PASSWORD_AUTH in Flussi di autenticazione. Per consentire l'accesso con password semplice per un'app lato server, modifica un client dell'app e scegli Accedi con credenziali amministrative lato server: ALLOW_ADMIN_USER_PASSWORD_AUTH.

Per attivare l'autenticazione basata sulla scelta con nome utente e password, configura il client dell'app in modo da consentirla. Modifica il client dell'app e scegli l'Choice-based accesso: ALLOW_USER_AUTH.



Per verificare che l'autenticazione tramite password sia disponibile nei flussi di autenticazione basati sulla scelta, accedi al menu e consulta la sezione in Opzioni per l'accesso basato sulla

scelta. Sign-in Puoi accedere con l'autenticazione con password semplice se la password è visibile in Scelte disponibili. L'opzione Password include le varianti di autenticazione nome utente-password semplice e SRP.



Configura `ExplicitAuthFlows` con le tue opzioni di autenticazione preferite con nome utente e password in una richiesta or. CreateUserPoolClientUpdateUserPoolClient

```
"ExplicitAuthFlows": [
   "ALLOW_USER_PASSWORD_AUTH",
   "ALLOW_ADMIN_USER_PASSWORD_AUTH",
   "ALLOW_USER_AUTH"
]
```

In una CreateUserPoolo più UpdateUserPoolrichieste, esegui la configurazione `Policies` con i flussi di autenticazione basati sulla scelta che desideri supportare. Il `PASSWORD` valore in `AllowedFirstAuthFactors` include sia le opzioni del flusso di autenticazione con password semplice che quelle del flusso di autenticazione SRP.

```
"Policies": {
   "SignInPolicy": {
      "AllowedFirstAuthFactors": [
         "PASSWORD",
         "EMAIL_OTP",
         "WEB_AUTHN"
      ]
   }
}
```

Choice-based sign-in with a password

Per far accedere un utente a un'applicazione con l'autenticazione nome utente e password, configura il corpo della richiesta o della richiesta come segue. AdminInitiateAuthInitiateAuth Questa richiesta di accesso ha esito positivo o prosegue fino alla sfida successiva se l'utente corrente è idoneo all'autenticazione con nome utente e password. In caso contrario, risponde con

un elenco di problemi di autenticazione con fattori primari disponibili. Questo set di parametri è il minimo richiesto per l'accesso. Sono disponibili parametri aggiuntivi.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "PREFERRED_CHALLENGE" : "PASSWORD",
      "PASSWORD" : "[User's password]"
   },
   "ClientId": "1example23456789"
}
```

Puoi anche omettere il `PREFERRED_CHALLENGE` valore e ricevere una risposta che contiene un elenco di fattori di accesso idonei per l'utente.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser"
   },
   "ClientId": "1example23456789"
}
```

Se non hai inviato una sfida preferita o l'utente inviato non è idoneo per la sua sfida preferita, Amazon Cognito restituisce un elenco di opzioni in. `AvailableChallenges` Se `AvailableChallenges` include una `ChallengeName` di`PASSWORD`, puoi continuare l'autenticazione con una risposta RespondToAuthChallengeo una risposta alla AdminRespondToAuthChallengesfida nel formato seguente. È necessario passare un `Session` parametro che associ la risposta alla sfida alla risposta dell'API alla richiesta di accesso iniziale. Questo set di parametri è il minimo richiesto per l'accesso. Sono disponibili parametri aggiuntivi.

```
{
   "ChallengeName": "PASSWORD",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "PASSWORD" : "[User's Password]"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response"
```

```
}
```

Amazon Cognito risponde alle richieste e alle risposte alle sfide preferite idonee e riuscite con token o con un'ulteriore `PASSWORD` sfida obbligatoria come l'autenticazione a più fattori (MFA).

Client-based sign-in with a password

Per far accedere un utente a un'app lato client con autenticazione nome utente e password, configura il corpo della richiesta come segue. InitiateAuth Questo set di parametri è il minimo richiesto per l'accesso. Sono disponibili parametri aggiuntivi.

```
{
   "AuthFlow": "USER_PASSWORD_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "PASSWORD" : "[User's password]"
   },
   "ClientId": "1example23456789"
}
```

Per far accedere un utente a un'app lato server con autenticazione nome utente e password, configura il corpo della richiesta come segue. AdminInitiateAuth L'applicazione deve firmare questa richiesta con credenziali. AWS Questo set di parametri è il minimo richiesto per l'accesso. Sono disponibili parametri aggiuntivi.

```
{
   "AuthFlow": "ADMIN_USER_PASSWORD_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "PASSWORD" : "[User's password]"
   },
   "ClientId": "1example23456789"
}
```

Amazon Cognito risponde alle richieste riuscite con token o con un'ulteriore sfida obbligatoria come l'autenticazione a più fattori (MFA).

Sign-in con password persistenti e payload sicuro

Un'altra forma di metodo di accesso con nome utente e password nei pool di utenti è il protocollo Secure Remote Password (SRP). Questa opzione invia una prova di conoscenza di una password,

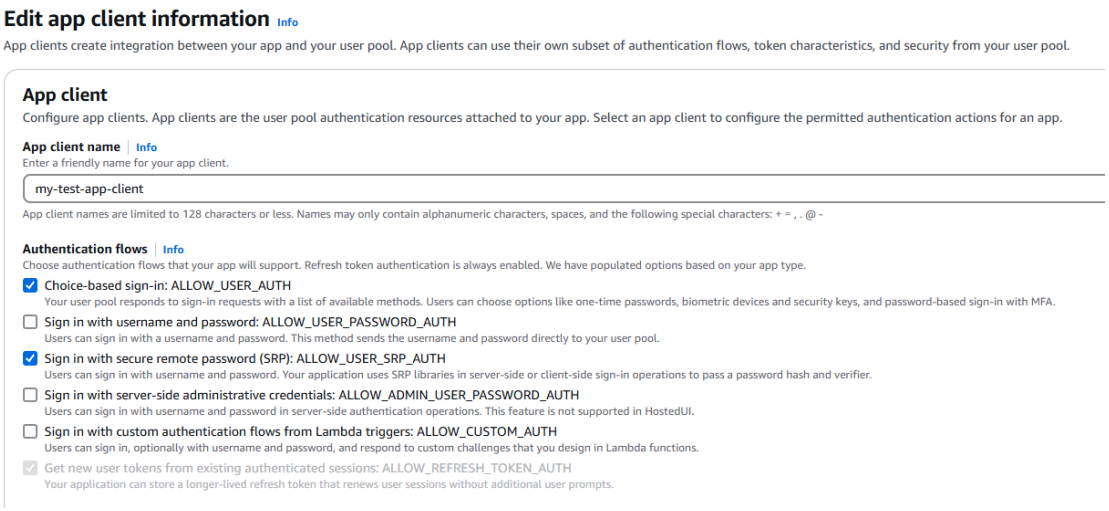
un hash e un codice salt, che il gruppo di utenti può verificare. Senza informazioni segrete leggibili nella richiesta ad Amazon Cognito, l'applicazione è l'unica entità che elabora le password inserite dagli utenti. L'autenticazione SRP prevede calcoli matematici che è preferibile eseguire utilizzando un componente esistente che puoi importare nel tuo SDK. Gli AWS SDK di basso livello espongono le operazioni `InitiateAuth` e le `RespondToAuthChallenge` API ma non calcolano i valori SRP al posto tuo. Higher-level Le librerie che implementano SRP, come le librerie Amplify, eseguono questi calcoli. Se il tuo SDK non fornisce un'implementazione SRP, devi calcolare tu stesso i valori SRP prima di avviare queste operazioni. L'SRP è in genere implementato in applicazioni lato client come le app mobili. Per ulteriori informazioni sul protocollo, consulta la home page di Stanford SRP. Wikipedia contiene anche risorse ed esempi. Sono disponibili diverse biblioteche pubbliche per eseguire i calcoli SRP per i flussi di autenticazione.

La sequenza iniziata-sfida-risposta dell'autenticazione Amazon Cognito convalida gli utenti e le relative password con SRP. Devi configurare il pool di utenti e il client dell'app per supportare l'autenticazione SRP, quindi implementare la logica delle richieste di accesso e delle risposte alle sfide nell'applicazione. Le vostre librerie SRP possono generare numeri casuali e valori calcolati che dimostrano al vostro pool di utenti che siete in possesso della password di un utente. L'applicazione inserisce questi valori calcolati nei `ChallengeParameters` campi JSON-formatted `AuthParameters` e nei pool di utenti di Amazon Cognito, nelle operazioni API e nei metodi SDK per l'autenticazione.

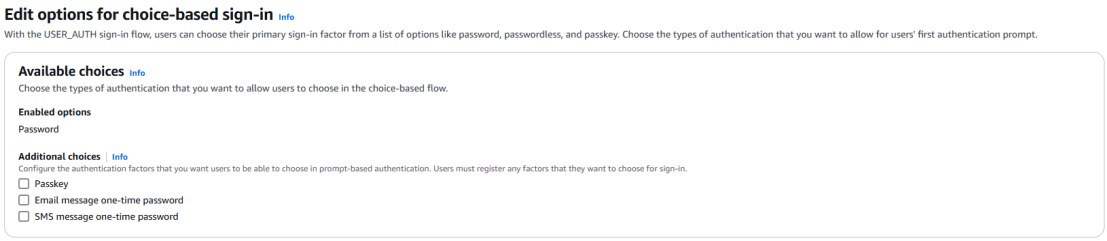
Activate SRP sign-in

Per attivare l'autenticazione basata su client con nome utente e SRP, configura il client dell'app in modo da consentirla. Nella console Amazon Cognito, accedi al menu App client in Applicazioni nella configurazione del pool di utenti. Per consentire l'accesso SRP a un'app mobile o nativa sul lato client, modifica un client dell'app e scegli Accedi con password remota sicura (SRP): ALLOW_USER_SRP_AUTH in Flussi di autenticazione.

Per attivare l'autenticazione basata sulla scelta con nome utente e SRP, modifica il client dell'app e scegli sign-in: ALLOW_USER_AUTH. Choice-based



Per verificare che l'autenticazione SRP sia disponibile nei flussi di autenticazione basata sulla scelta, accedi al menu e consulta la sezione in Opzioni per l'accesso basato sulla scelta. Sign-in Puoi accedere con l'autenticazione SRP se la password è visibile in Scelte disponibili. L'opzione Password include le varianti di autenticazione nome utente-password in chiaro e SRP.



Configura `ExplicitAuthFlows` con le tue opzioni di autenticazione preferite con nome utente e password in una richiesta or. CreateUserPoolClientUpdateUserPoolClient

```
"ExplicitAuthFlows": [
   "ALLOW_USER_SRP_AUTH",
   "ALLOW_USER_AUTH"
]
```

In una CreateUserPoolo più UpdateUserPoolrichieste, esegui la configurazione `Policies` con i flussi di autenticazione basati sulla scelta che desideri supportare. Il `PASSWORD` valore in `AllowedFirstAuthFactors` include sia le opzioni del flusso di autenticazione plaintext-password che quelle del flusso di autenticazione SRP.

```
"Policies": {
   "SignInPolicy": {
      "AllowedFirstAuthFactors": [
```

```
            "PASSWORD",
            "EMAIL_OTP",
            "WEB_AUTHN"
        ]
    }
}
```

Choice-based sign-in with SRP

Per far accedere un utente a un'applicazione con autenticazione nome utente e password con SRP, configura il corpo della tua richiesta come segue. AdminInitiateAuthInitiateAuth Questa richiesta di accesso ha esito positivo o prosegue fino alla sfida successiva se l'utente corrente è idoneo all'autenticazione nome utente-password. In caso contrario, risponde con un elenco di problemi di autenticazione con fattori primari disponibili. Questo set di parametri è il minimo richiesto per l'accesso. Sono disponibili parametri aggiuntivi.

```
{
    "AuthFlow": "USER_AUTH",
    "AuthParameters": {
        "USERNAME" : "testuser",
        "PREFERRED_CHALLENGE" : "PASSWORD_SRP",
        "SRP_A" : "[g^a % N]"
    },
    "ClientId": "1example23456789"
}
```

Puoi anche omettere il `PREFERRED_CHALLENGE` valore e ricevere una risposta che contiene un elenco di fattori di accesso idonei per l'utente.

```
{
    "AuthFlow": "USER_AUTH",
    "AuthParameters": {
        "USERNAME" : "testuser"
    },
    "ClientId": "1example23456789"
}
```

Se non hai inviato una sfida preferita o l'utente inviato non è idoneo per la sua sfida preferita, Amazon Cognito restituisce un elenco di opzioni in. `AvailableChallenges` Se `AvailableChallenges` include una `ChallengeName` di`PASSWORD_SRP`, puoi continuare l'autenticazione con una risposta RespondToAuthChallengeo una risposta alla

AdminRespondToAuthChallengesfida nel formato seguente. È necessario passare un `Session` parametro che associ la risposta alla sfida alla risposta dell'API alla richiesta di accesso iniziale. Questo set di parametri è il minimo richiesto per l'accesso. Sono disponibili parametri aggiuntivi.

```
{
   "ChallengeName": "PASSWORD_SRP",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "SRP_A" : "[g^a % N]"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response"
}
```

Amazon Cognito risponde alle richieste di sfide preferite idonee e `PASSWORD_SRP` alle risposte alle sfide con una sfida. `PASSWORD_VERIFIER` Il tuo cliente deve completare i calcoli SRP e rispondere alla sfida in una richiesta OR. RespondToAuthChallengeAdminRespondToAuthChallenge

```
{
   "ChallengeName": "PASSWORD_VERIFIER",
   "ChallengeResponses": {
      "PASSWORD_CLAIM_SIGNATURE" : "string",
      "PASSWORD_CLAIM_SECRET_BLOCK" : "string",
      "TIMESTAMP" : "string"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response]"
}
```

In caso di risposta riuscita alla `PASSWORD_VERIFIER` sfida, Amazon Cognito emette token o un'altra sfida obbligatoria come l'autenticazione a più fattori (MFA).

Client-based sign-in with SRP

L'autenticazione SRP è più comune all'autenticazione lato client che a quella lato server. Tuttavia, è possibile utilizzare l'autenticazione SRP con e. InitiateAuthAdminInitiateAuth Per far accedere un utente a un'applicazione, configura il corpo della tua `InitiateAuth` `AdminInitiateAuth` richiesta come segue. Questo set di parametri è il minimo richiesto per l'accesso. Sono disponibili parametri aggiuntivi.

Il client genera SRP_A da un generatore il modulo N g elevato alla potenza di un intero casuale segreto a.

```
{
   "AuthFlow": "USER_SRP_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "SRP_A" : "[g^a % N]"
   },
   "ClientId": "1example23456789"
}
```

Amazon Cognito risponde con una richiesta PASSWORD_VERIFIER. Il tuo cliente deve completare i calcoli SRP e rispondere alla sfida in una richiesta RespondToAuthChallengeo AdminRespondToAuthChallenge.

```
{
   "ChallengeName": "PASSWORD_VERIFIER",
   "ChallengeResponses": {
      "PASSWORD_CLAIM_SIGNATURE" : "string",
      "PASSWORD_CLAIM_SECRET_BLOCK" : "string",
      "TIMESTAMP" : "string"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response]"
}
```

In caso di risposta riuscita alla PASSWORD_VERIFIER sfida, Amazon Cognito emette token o un'altra sfida obbligatoria come l'autenticazione a più fattori (MFA).

Accesso senza password con codici monouso

Le password possono essere perse o rubate. Potresti voler verificare solo che i tuoi utenti abbiano accesso a un indirizzo email, a un numero di telefono o a un'app di autenticazione verificati. La soluzione a questo problema è l'accesso senza password. L'applicazione può richiedere agli utenti di inserire il proprio nome utente, indirizzo e-mail o numero di telefono. Amazon Cognito genera quindi una password monouso (OTP), un codice che devono confermare. Un codice corretto completa l'autenticazione.

One-time i flussi di autenticazione tramite password (OTP) non sono compatibili con l'autenticazione a più fattori (MFA) richiesta nel tuo pool di utenti. L'autenticazione con chiave di accesso con verifica utente può soddisfare i requisiti MFA se `FactorConfiguration` impostata su. `MULTI_FACTOR_WITH_USER_VERIFICATION` Se la MFA è facoltativa nel tuo pool di utenti, gli utenti che hanno attivato l'MFA non possono accedere con un primo fattore OTP. Gli utenti che non dispongono di una preferenza MFA in un pool di MFA-optional utenti possono accedere con fattori senza password. Per ulteriori informazioni, consulta Cose da sapere sull'MFA per pool di utenti.

Quando un utente inserisce correttamente un codice ricevuto in un SMS o in un messaggio e-mail come parte dell'autenticazione senza password, oltre ad autenticare l'utente, il pool di utenti contrassegna l'indirizzo e-mail o l'attributo del numero di telefono non verificato dell'utente come verificato. Lo stato dell'utente è cambiato anche da `UNCONFIRMED` a`CONFIRMED`, indipendentemente dal fatto che il pool di utenti sia stato configurato per verificare automaticamente gli indirizzi e-mail o i numeri di telefono.

Nuove opzioni con accesso senza password

Quando attivi l'autenticazione senza password nel tuo pool di utenti, cambia il funzionamento di alcuni flussi di utenti.

1. Gli utenti possono registrarsi senza password e scegliere un fattore senza password al momento dell'accesso. Puoi anche creare utenti senza password come amministratore.
2. Gli utenti importati con un file CSV possono accedere immediatamente senza password. Non sono tenuti a impostare una password prima di accedere.
3. Gli utenti che non dispongono di una password possono inviare richieste ChangePasswordAPI senza il `PreviousPassword` parametro.

Accesso automatico con OTP

Gli utenti che si registrano e confermano i propri account utente tramite e-mail o SMS OTP possono accedere automaticamente con il fattore senza password corrispondente al messaggio di conferma. Nell'interfaccia utente di accesso gestito, gli utenti che confermano i propri account e sono idonei all'accesso OTP con il metodo di invio del codice di conferma procedono automaticamente al primo accesso dopo aver fornito il codice di conferma. Nella tua applicazione personalizzata con un AWS SDK, passa i seguenti parametri a un'operazione or. InitiateAuthAdminInitiateAuth

- Il `Session` parametro della risposta dell'ConfirmSignUpAPI come parametro di richiesta. `Session`
- Un AuthFlowdi`USER_AUTH`.

Puoi passare un PREFERRED_CHALLENGE di `EMAIL_OTP` o`SMS_OTP`, ma non è obbligatorio. Il `Session` parametro fornisce una prova di autenticazione e Amazon Cognito ignora `AuthParameters` quando trasmetti un codice di sessione valido.

L'operazione di accesso restituisce la risposta che indica l'avvenuta autenticazione AuthenticationResult, senza ulteriori problemi se sono vere le seguenti condizioni.

- Il `Session` codice è valido e non è scaduto.
- L'utente è idoneo per il metodo di autenticazione OTP.

Activate passwordless sign-in

Console

Per attivare l'accesso senza password, configura il tuo pool di utenti in modo da consentire l'accesso principale con uno o più tipi di accesso senza password, quindi configura il client dell'app per consentire il flusso. `USER_AUTH` Nella console Amazon Cognito, accedi al Sign-inmenu sotto Autenticazione nella configurazione del tuo pool di utenti. Modifica le opzioni per l'accesso basato sulla scelta e scegli la password monouso per i messaggi di posta elettronica o la password monouso per i messaggi SMS. Puoi attivare entrambe le opzioni. Salvare le modifiche.

Vai al menu App clients e scegli un app client o creane uno nuovo. Seleziona Modifica e scegli Seleziona un tipo di autenticazione al momento dell'accesso: ALLOW_USER_AUTH.

API/SDK

Nell'API dei pool di utenti, configura `SignInPolicy` con le opzioni senza password appropriate in una richiesta or. CreateUserPoolUpdateUserPool

```
"SignInPolicy": {
    "AllowedFirstAuthFactors": [
        "EMAIL_OTP",
        "SMS_OTP"
    ]
}
```

Configura il client dell'app `ExplicitAuthFlows` con l'opzione richiesta in una richiesta CreateUserPoolClientor UpdateUserPoolClient.

```
"ExplicitAuthFlows": [
   "ALLOW_USER_AUTH"
]
```

Sign in with passwordless

L'accesso senza password non dispone di un client `AuthFlow` che puoi specificare in e. InitiateAuthAdminInitiateAuth L'autenticazione OTP è disponibile solo nella versione basata sulla scelta `AuthFlow` diUSER_AUTH, in cui è possibile richiedere un'opzione di accesso preferita o scegliere l'opzione senza password tra quelle di un utente. AvailableChallenges Per far accedere un utente a un'applicazione, configura il corpo della tua richiesta come segue. `InitiateAuth` `AdminInitiateAuth` Questo set di parametri è il minimo richiesto per l'accesso. Sono disponibili parametri aggiuntivi.

In questo esempio, non sappiamo in che modo l'utente desidera accedere. Se aggiungiamo un `PREFERRED_CHALLENGE` parametro e la sfida preferita è disponibile per l'utente, Amazon Cognito risponde con quella sfida.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser"
   },
   "ClientId": "1example23456789"
}
```

Puoi invece aggiungere `"PREFERRED_CHALLENGE": "EMAIL_OTP"` o `"PREFERRED_CHALLENGE": "SMS_OTP"` a `AuthParameters` in questo esempio. Se l'utente è idoneo per il metodo preferito, il pool di utenti invia immediatamente un codice all'indirizzo e-mail o al numero di telefono dell'utente e restituisce `"ChallengeName": "EMAIL_OTP"` o`"ChallengeName": "SMS_OTP"`.

Se non specifichi una sfida preferita, Amazon Cognito risponde con un parametro. `AvailableChallenges`

```
{
   "AvailableChallenges": [
      "EMAIL_OTP",
      "SMS_OTP",
```

```
      "PASSWORD"
    ],
   "Session": "[Session ID]"
}
```

Questo utente è idoneo all'accesso senza password con messaggio di posta elettronica OTP, messaggio SMS OTP e nome-password. L'applicazione può richiedere all'utente di effettuare la selezione o effettuare una selezione in base alla logica interna. Quindi procede con una AdminRespondToAuthChallengerichiesta RespondToAuthChallengeor che seleziona la sfida. Supponiamo che l'utente desideri completare l'autenticazione senza password con un messaggio di posta elettronica OTP.

```
{
   "ChallengeName": "SELECT_CHALLENGE",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "ANSWER" : "EMAIL_OTP"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response]"
}
```

Amazon Cognito risponde con una `EMAIL_OTP` sfida e invia un codice all'indirizzo e-mail verificato dell'utente. La tua applicazione deve quindi rispondere nuovamente a questa sfida.

Questa sarebbe anche la prossima risposta alla sfida se lo richiedessi `EMAIL_OTP` come`PREFERRED_CHALLENGE`.

```
{
   "ChallengeName": "EMAIL_OTP",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "EMAIL_OTP_CODE" : "123456"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response]"
}
```

Accesso senza password con passkey WebAuthn

Le passkey sono sicure e richiedono un livello di impegno relativamente basso agli utenti. L'accesso tramite passkey utilizza autenticatori, dispositivi esterni con cui gli utenti possono autenticarsi. Le password normali espongono gli utenti a vulnerabilità come il phishing, l'indovinazione delle password e il furto di credenziali. Con le passkey, l'applicazione può trarre vantaggio da misure di sicurezza avanzate su telefoni cellulari e altri dispositivi collegati o integrati nei sistemi di informazione. Un flusso di lavoro comune per l'accesso tramite passkey inizia con una chiamata al dispositivo che richiama il gestore della password o delle credenziali, ad esempio il portachiavi iOS o il gestore di password di Google Chrome. Il gestore delle credenziali sul dispositivo richiede loro di selezionare una passkey e di autorizzarla con una credenziale esistente o un meccanismo di sblocco del dispositivo. I telefoni moderni dispongono di scanner facciali, scanner di impronte digitali, schemi di sblocco e altri meccanismi, alcuni dei quali soddisfano contemporaneamente i principi di autenticazione avanzata e qualcosa che conosci. Nel caso dell'autenticazione biometrica tramite passkey, le passkey rappresentano qualcosa che sei.

Potresti voler sostituire le password con l'autenticazione tramite impronta digitale, facciale o chiave di sicurezza. Si tratta di una chiave di accesso o di autenticazione. WebAuthn È normale che gli sviluppatori di applicazioni consentano agli utenti di registrare un dispositivo biometrico dopo il primo accesso con una password. Con i pool di utenti di Amazon Cognito, l'applicazione può configurare questa opzione di accesso per gli utenti. L'autenticazione con chiave di accesso può soddisfare i requisiti di autenticazione a più fattori (MFA) se il pool di utenti è impostato su. `FactorConfiguration` `MULTI_FACTOR_WITH_USER_VERIFICATION` In questa configurazione, l'autenticazione con chiave di accesso con verifica dell'utente viene considerata come autenticazione a più fattori.

One-time i flussi di autenticazione tramite password (OTP) non sono compatibili con l'autenticazione a più fattori (MFA) richiesta nel tuo pool di utenti. L'autenticazione con chiave di accesso con verifica utente può soddisfare i requisiti MFA se `FactorConfiguration` impostata su. `MULTI_FACTOR_WITH_USER_VERIFICATION` Se la MFA è facoltativa nel tuo pool di utenti, gli utenti che hanno attivato l'MFA non possono accedere con un primo fattore OTP. Gli utenti che non dispongono di una preferenza MFA in un pool di MFA-optional utenti possono accedere con fattori senza password. Per ulteriori informazioni, consulta Cose da sapere sull'MFA per pool di utenti.

Cosa sono le passkey?

Le passkey semplificano l'esperienza utente eliminando la necessità di ricordare password complesse o inserire OTP. Le password si basano sugli standard CTAP2 redatti dal World Wide Web

Consortium (W3C) WebAuthn e dalla FIDO (Fast Identity Online) Alliance. I browser e le piattaforme implementano questi standard, forniscono API per applicazioni web o mobili per avviare un processo di registrazione o autenticazione delle passkey e anche un'interfaccia utente che consente all'utente di selezionare e interagire con un autenticatore di passkey.

Quando un utente registra un autenticatore su un sito Web o un'app, l'autenticatore crea una coppia di chiavi pubblica-privata. WebAuthn i browser e le piattaforme inviano la chiave pubblica al backend dell'applicazione del sito Web o dell'app. L'autenticatore conserva la chiave privata, gli ID delle chiavi e i metadati relativi all'utente e all'applicazione. Quando l'utente desidera autenticarsi nell'applicazione registrata con il proprio autenticatore registrato, l'applicazione genera una sfida casuale. La risposta a questa sfida è la firma digitale della sfida generata con la chiave privata dell'autenticatore per l'applicazione e l'utente e i metadati pertinenti. Il browser o la piattaforma applicativa riceve la firma digitale e la trasmette al backend dell'applicazione. L'applicazione convalida quindi la firma con la chiave pubblica memorizzata.

> **Note**
>
> L'applicazione non riceve alcun segreto di autenticazione fornito dagli utenti al proprio autenticatore, né riceve informazioni sulla chiave privata.

Di seguito sono riportati alcuni esempi e funzionalità degli autenticatori attualmente sul mercato. Un autenticatore può soddisfare una o tutte queste categorie.

- Alcuni autenticatori eseguono la verifica dell'utente con fattori come un PIN, un input biometrico con un volto o un'impronta digitale o un codice di accesso prima di concedere l'accesso, garantendo che solo l'utente legittimo possa autorizzare le azioni. Altri autenticatori non dispongono di funzionalità di verifica degli utenti e alcuni possono ignorare la verifica dell'utente quando un'applicazione non la richiede.
- Alcuni autenticatori, ad esempio i token YubiKey hardware, sono portatili. Comunicano con i dispositivi tramite connessioni USB, Bluetooth o NFC. Alcuni autenticatori sono locali e associati a una piattaforma, ad esempio Windows Hello su un PC o Face ID su un iPhone. Un autenticatore legato al dispositivo può essere utilizzato dall'utente se sufficientemente piccolo, come un dispositivo mobile. A volte gli utenti possono connettere il proprio autenticatore hardware a molte piattaforme diverse con comunicazione wireless. Ad esempio, gli utenti dei browser desktop possono utilizzare il proprio smartphone come autenticatore di password quando scansionano un codice QR.

- Alcune passkey legate alla piattaforma si sincronizzano con il cloud in modo da poter essere utilizzate da più postazioni. Ad esempio, le passkey Face ID sugli iPhone sincronizzano i metadati delle passkey con gli account Apple degli utenti nel portachiavi iCloud. Queste passkey garantiscono un'autenticazione senza interruzioni su tutti i dispositivi Apple, anziché richiedere che gli utenti registrino ciascun dispositivo in modo indipendente. Software-based app di autenticazione come 1Password, Dashlane e Bitwarden sincronizzano le passkey su tutte le piattaforme su cui l'utente ha installato l'app.

In WebAuthn termini terminologici, i siti Web e le app sono parti affidabili. Ogni passkey è associata a un ID relying party specifico, un identificatore unificato che rappresenta i siti Web o le app che accettano l'autenticazione tramite passkey. Gli sviluppatori devono selezionare attentamente l'ID del relying party per disporre del giusto ambito di autenticazione. Un tipico ID relying party è il nome di dominio principale di un server web. Una passkey con questa specifica relativa all'ID del relying party può autenticarsi per quel dominio e i sottodomini. I browser e le piattaforme negano l'autenticazione con chiave di accesso quando l'URL del sito Web a cui un utente desidera accedere non corrisponde all'ID del relying party. Analogamente, per le app mobili, una passkey può essere utilizzata solo se il percorso dell'app è presente nei file di `.well-known` associazione che l'applicazione rende disponibili nel percorso indicato dall'ID del relying party.

Le passkey sono individuabili. Possono essere riconosciuti e utilizzati automaticamente da un browser o da una piattaforma senza richiedere all'utente di inserire un nome utente. Quando un utente visita un sito Web o un'app che supporta l'autenticazione con password, può selezionare da un elenco di passkey che il browser o la piattaforma già conoscono oppure può scansionare un codice QR.

In che modo Amazon Cognito implementa l'autenticazione con chiave di accesso?

Le passkey sono una funzionalità opzionale disponibile in tutti i piani di funzionalità ad eccezione di Lite. È disponibile solo nel flusso di autenticazione basato sulla scelta. Con l'accesso gestito, Amazon Cognito gestisce la logica dell'autenticazione tramite passkey. Puoi anche utilizzare l'API dei pool di utenti di Amazon Cognito negli AWS SDK per eseguire l'autenticazione tramite passkey nel back-end dell'applicazione.

Amazon Cognito riconosce le passkey create utilizzando uno dei due algoritmi crittografici asimmetrici, ES256 (-7) e RS256 (-257). La maggior parte degli autenticatori supporta entrambi gli algoritmi. Per impostazione predefinita, gli utenti possono configurare qualsiasi tipo di autenticatore, ad esempio token hardware, smartphone mobili e app di autenticazione software. Amazon Cognito attualmente non supporta l'applicazione degli attestati.

Nel tuo pool di utenti, puoi configurare la verifica degli utenti in modo che sia preferita o obbligatoria. L'impostazione predefinita è preferita nelle richieste API che non forniscono un valore e preferenziale è selezionata per impostazione predefinita nella console Amazon Cognito. Quando imposti la verifica utente su Preferenziale, gli utenti possono configurare autenticatori che non dispongono della funzionalità di verifica dell'utente e le operazioni di registrazione e autenticazione possono avere esito positivo senza la verifica dell'utente. Per imporre la verifica dell'utente nella registrazione e nell'autenticazione delle passkey, modifica questa impostazione in obbligatoria.

L'ID relying party (RP) impostato nella configurazione della passkey è una decisione importante. Se non specifichi diversamente e la versione con marchio del dominio è Managed Login, per impostazione predefinita il pool di utenti prevede il nome del dominio personalizzato come ID RP. Se non disponi di un dominio personalizzato e non specifichi diversamente, il tuo pool di utenti utilizza per impostazione predefinita un ID RP del tuo dominio con prefisso. Puoi anche configurare il tuo ID RP in modo che sia qualsiasi nome di dominio non presente nell'elenco dei suffissi pubblici (PSL). L'immissione dell'ID RP si applica alla registrazione e all'autenticazione della passkey nell'accesso gestito e nell'autenticazione SDK. Passkey funziona nelle applicazioni mobili solo quando Amazon Cognito è in grado di individuare `.well-known` un file di associazione con il tuo ID RP come dominio. Come best practice, determina e imposta il valore dell'ID del relying party prima che il sito Web o l'app siano disponibili al pubblico. Se modifichi il tuo ID RP, gli utenti devono registrarsi nuovamente con il nuovo ID RP.

Ogni utente può registrare fino a 20 passkey. Possono registrare una passkey solo dopo aver effettuato l'accesso al tuo pool di utenti almeno una volta. L'accesso gestito elimina uno sforzo significativo dalla registrazione delle passkey. Quando abiliti l'autenticazione con passkey per un pool di utenti e un client di app, il pool di utenti con un dominio di accesso gestito ricorda agli utenti finali di registrare una passkey dopo aver registrato un nuovo account utente. Puoi anche richiamare i browser degli utenti in qualsiasi momento per indirizzarli a una pagina di accesso gestita per la registrazione della passkey. Gli utenti devono fornire un nome utente prima che Amazon Cognito possa avviare l'autenticazione tramite passkey. L'accesso gestito lo gestisce automaticamente. La pagina di accesso richiede un nome utente, verifica che l'utente abbia almeno una passkey registrata e quindi richiede l'accesso tramite passkey. Analogamente, SDK-based le applicazioni devono richiedere un nome utente e fornirlo nella richiesta di autenticazione.

Quando si configura l'autenticazione del pool di utenti con passkey e si dispone di un dominio personalizzato e di un dominio con prefisso, l'ID RP utilizza per impostazione predefinita il nome di dominio completo (FQDN) del dominio personalizzato. Per impostare un dominio con prefisso come ID RP nella console Amazon Cognito, elimina il dominio personalizzato o inserisci il nome di dominio completo del prefisso come dominio. Third-party

Activate passkey sign-in

Console

Per attivare l'accesso con passkey, configura il tuo pool di utenti in modo da consentire l'accesso principale con uno o più tipi senza password, quindi configura il client dell'app per consentire il flusso. USER_AUTH Nella console Amazon Cognito, accedi al Sign-inmenu sotto Autenticazione nella configurazione del tuo pool di utenti. Modifica le opzioni per l'accesso basato sulla scelta e aggiungi Passkey all'elenco delle scelte disponibili.

Vai al menu Metodi di autenticazione e modifica Passkey.

- La verifica utente è l'impostazione che consente di stabilire se il pool di utenti richiede dispositivi con chiave di accesso che eseguono controlli aggiuntivi per verificare che l'utente corrente sia autorizzato a utilizzare una passkey. Per incoraggiare gli utenti a configurare un dispositivo con la verifica utente, ma non richiederla, seleziona Preferito. Per supportare solo i dispositivi con verifica utente, seleziona Obbligatorio. Per ulteriori informazioni, consulta Verifica utente su w3.org.
- Il Domain for relying party ID è l'identificatore che l'applicazione trasmette nelle richieste di registrazione delle passkey degli utenti. Imposta l'obiettivo della relazione di fiducia con l'emittente delle passkey degli utenti. L'ID del relying party può essere uno dei seguenti:

  Dominio Cognito

  Il dominio del prefisso Amazon Cognito del tuo pool di utenti.

  Dominio personalizzato

  Il dominio personalizzato del tuo pool di utenti.

  Third-party dominio

  Il dominio per le applicazioni che non utilizzano le pagine di accesso gestite dai pool di utenti. Questa impostazione è in genere associata a pool di utenti che non dispongono di un dominio ed eseguono l'autenticazione con un AWS SDK e l'API dei pool di utenti nel backend.

Vai al menu App clients e scegli un app client o creane uno nuovo. Seleziona Modifica e in Flussi di autenticazione, scegli Seleziona un tipo di autenticazione al momento dell'accesso: ALLOW_USER_AUTH.

API/SDK

Nell'API dei pool di utenti, configura `SignInPolicy` con le opzioni di passkey appropriate in una richiesta or. CreateUserPoolUpdateUserPool L'`WEB_AUTHN`opzione per l'autenticazione tramite passkey deve essere accompagnata da almeno un'altra opzione. La registrazione della passkey richiede una sessione di autenticazione esistente.

```
"SignInPolicy": {
    "AllowedFirstAuthFactors": [
        "PASSWORD",
        "WEB_AUTHN"
    ]
}
```

Configura la preferenza di verifica dell'utente e l'ID RP nel `WebAuthnConfiguration` parametro di una richiesta. SetUserPoolMfaConfig L'`RelyingPartyId`obiettivo previsto dell'autenticazione tramite passkey può essere il prefisso del pool di utenti o il dominio personalizzato oppure un dominio di tua scelta.

```
"WebAuthnConfiguration": {
   "RelyingPartyId": "example.auth.us-east-1.amazoncognito.com",
   "UserVerification": "preferred",
   "FactorConfiguration": "SINGLE_FACTOR"
}
```

Configura il client dell'app `ExplicitAuthFlows` con l'opzione richiesta in una CreateUserPoolClientrichiesta or. UpdateUserPoolClient

```
"ExplicitAuthFlows": [
   "ALLOW_USER_AUTH"
]
```

Register a passkey (managed login)

L'accesso gestito gestisce la registrazione delle passkey da parte degli utenti. Quando l'autenticazione con chiave di accesso è attiva nel tuo pool di utenti, Amazon Cognito richiede agli utenti di configurare una passkey quando si registrano per un nuovo account utente.

Amazon Cognito non richiede agli utenti di configurare una passkey se si sono già registrati e non hanno configurato una passkey o se hai creato il loro account come amministratore. Gli utenti in questo stato devono accedere con un altro fattore, come una password o una password OTP senza password, prima di poter registrare una passkey.

Per registrare una passkey

1. Indirizza l'utente alla tua pagina di accesso.

   ```
   https://auth.example.com/oauth2/authorize/?
   client_id=1example23456789&response_type=code&scope=email+openid
   +phone&redirect_uri=https%3A%2F%2Fwww.example.com
   ```

2. Elabora il risultato dell'autenticazione dell'utente. In questo esempio, Amazon Cognito li reindirizza `www.example.com` con un codice di autorizzazione che l'applicazione scambia con token.
3. Indirizza l'utente alla tua pagina register-passkey. L'utente disporrà di un cookie del browser che mantiene la sessione di accesso. L'URL della passkey richiede e parametri. `client_id` `redirect_uri` Amazon Cognito consente solo agli utenti autenticati di accedere a questa pagina. Accedi al tuo utente con una password, un'e-mail OTP o un SMS OTP, quindi richiama un URL che corrisponda allo schema seguente.

   Puoi anche aggiungere altri Endpoint Authorize parametri a questa richiesta, come e. `response_type` `scope`

   ```
   https://auth.example.com/passkeys/add?
   client_id=1example23456789&redirect_uri=https%3A%2F%2Fwww.example.com
   ```

Register a passkey (SDK)

Le credenziali della passkey vengono registrate con i metadati in un oggetto. PublicKeyCreationOptions È possibile generare questo oggetto con le credenziali di un utente connesso e presentarle in una richiesta API all'emittente della passkey. L'emittente restituisce un RegistrationResponse oggetto JSON che conferma la registrazione della passkey.

Per avviare il processo di registrazione della passkey, accedi a un utente con un'opzione di accesso esistente. Autorizza la richiesta StartWebAuthnRegistrationAPI autorizzata dal token con il token di accesso dell'utente corrente. Di seguito è riportato il corpo di una richiesta di esempio. `GetWebAuthnRegistrationOptions`

```
{
   "AccessToken": "eyJra456defEXAMPLE"
}
```

La risposta del pool di utenti contiene l'`PublicKeyCreationOptions`oggetto. Presenta questo oggetto in una richiesta API all'emittente dell'utente. Fornisce informazioni come la chiave pubblica e l'ID del relying party. L'emittente risponde con un oggetto. `RegistrationResponseJSON`

Presenta la risposta di registrazione in una richiesta CompleteWebAuthnRegistrationAPI, nuovamente autorizzata con il token di accesso dell'utente. Quando il pool di utenti risponde con una risposta HTTP 200 con un corpo vuoto, la passkey dell'utente viene registrata.

Sign in with a passkey

Per l'accesso senza password non è possibile specificare nulla `AuthFlow` in e. InitiateAuthAdminInitiateAuth Devi invece dichiarare un no `USER_AUTH` e richiedere un'opzione `AuthFlow` di accesso o scegliere l'opzione senza password dalla risposta del tuo pool di utenti. Per far accedere un utente a un'applicazione, configura il corpo della tua `InitiateAuth` richiesta come segue. `AdminInitiateAuth` Questo set di parametri è il minimo richiesto per l'accesso. Sono disponibili parametri aggiuntivi.

In questo esempio, sappiamo che l'utente desidera accedere con una passkey e aggiungiamo un `PREFERRED_CHALLENGE` parametro.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "PREFERRED_CHALLENGE" : "WEB_AUTHN"
   },
   "ClientId": "1example23456789"
}
```

Amazon Cognito risponde con una richiesta `WEB_AUTHN`. L'applicazione deve rispondere a questa sfida. Avvia una richiesta di accesso con il fornitore di passkey dell'utente. Restituisce un oggetto JSONAuthenticationResponse.

```
{
   "ChallengeName": "WEB_AUTHN",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "CREDENTIAL" : "{AuthenticationResponseJSON}"
   },
```

```
    "ClientId": "1example23456789",
    "Session": "[Session ID from the previous response]"
}
```

MFA dopo l'accesso

È possibile configurare gli utenti che completano l'accesso con un flusso nome utente-password in modo che vengano richieste ulteriori verifiche con una password monouso da un messaggio di posta elettronica, un messaggio SMS o un'applicazione per la generazione di codice. L'autenticazione a più fattori è distinta dall'accesso senza password con password monouso. Tuttavia, le passkey con verifica utente possono soddisfare i requisiti MFA come primo fattore quando si `FactorConfiguration` configura `MULTI_FACTOR_WITH_USER_VERIFICATION` come nel pool di utenti. `WebAuthnConfiguration` Le passkey non possono essere utilizzate come secondo fattore per l'accesso tramite password. Per i flussi basati su password, l'MFA nei pool di utenti è un modello di risposta alla sfida in cui un utente dimostra innanzitutto di conoscere la password, quindi dimostra di avere accesso al proprio dispositivo registrato a secondo fattore.

Risorse per l'implementazione

- Aggiunta dell'autenticazione MFA a un bacino d'utenza

Aggiorna i token

Se desideri mantenere gli utenti connessi senza reinserire le loro credenziali, i token di aggiornamento sono lo strumento di cui dispone l'applicazione per rendere persistente la sessione di un utente. Le applicazioni possono presentare token di aggiornamento al pool di utenti e scambiarli con nuovi ID e token di accesso. Con l'aggiornamento dei token, puoi assicurarti che un utente connesso sia ancora attivo, ottenere informazioni aggiornate sugli attributi e aggiornare i diritti di controllo degli accessi senza l'intervento dell'utente.

Risorse per l'implementazione

- Aggiorna i token

Autenticazione personalizzata

Potresti voler configurare un metodo di autenticazione per i tuoi utenti che non è elencato qui. Puoi farlo con l'autenticazione personalizzata con trigger Lambda. In una sequenza di funzioni Lambda,

Amazon Cognito lancia una sfida, pone una domanda a cui gli utenti devono rispondere, verifica l'accuratezza della risposta, quindi determina se deve essere lanciata un'altra sfida. Le domande e le risposte possono includere domande di sicurezza, richieste a un servizio CAPTCHA, richieste a un'API di servizio MFA esterna o tutte queste in sequenza.

Risorse per l'implementazione

- Trigger Lambda di richieste di autenticazione personalizzate

Flusso di autenticazione personalizzato

I pool di utenti di Amazon Cognito consentono inoltre di utilizzare flussi di autenticazione personalizzati, che possono aiutarti a creare un modello di autenticazione challenge/response basato su trigger. AWS Lambda

Il flusso di autenticazione personalizzato permette cicli personalizzati di richieste e risposte per soddisfare diversi requisiti. Il flusso comincia con una chiamata all'operazione API `InitiateAuth` che indica il tipo di autenticazione da utilizzare e fornisce qualsiasi parametro di autenticazione iniziale. Amazon Cognito risponde alla chiamata `InitiateAuth` con uno dei seguenti tipi di informazione:

- Una sfida all'utente, insieme a una sessione e parametri.
- Un errore se l'utente non si autentica correttamente.
- Token ID, di accesso e di aggiornamento, se i parametri forniti nella chiamata `InitiateAuth` sono sufficienti per l'accesso dell'utente. (In genere l'utente o l'app devono prima rispondere a una sfida, ma è il tuo codice personalizzato a decidere se è il caso.)

Se Amazon Cognito risponde alla chiamata `InitiateAuth` con una sfida, l'app raccoglierà più input e chiamerà l'operazione `RespondToAuthChallenge`. Questa chiamata fornisce le risposte alla sfida e restituisce la sessione. Amazon Cognito risponde alla chiamata `RespondToAuthChallenge` in modo simile alla chiamata `InitiateAuth`. Se l'utente ha effettuato l'accesso, Amazon Cognito fornisce i token, se non ha effettuato l'accesso, Amazon Cognito restituisce un'altra sfida o un errore. Se Amazon Cognito restituisce un'altra sfida, la sequenza si ripete: l'app chiama `RespondToAuthChallenge` fino a quando l'utente non riesce a effettuare l'accesso o non viene restituito un errore. Per ulteriori dettagli sulle operazioni `InitiateAuth` e API `RespondToAuthChallenge`, consulta la documentazione API.

Flusso di autenticazione personalizzato e sfide

Un'app è in grado di avviare un flusso di autenticazione chiamando `InitiateAuth` con `CUSTOM_AUTH` come `Authflow`. Nel caso di un flusso di autenticazione personalizzato, tre trigger Lambda controllano le sfide e la verifica delle risposte.

- Il trigger Lambda `DefineAuthChallenge` usa come input una matrice di sessioni di sfide e risposte precedenti. Quindi genera il nome della sfida successiva e i booleani che indicano se l'utente è autenticato e deve ricevere i token o meno. Questo trigger Lambda è una macchina a stati che controlla il percorso dell'utente attraverso le richieste.
- Il trigger Lambda `CreateAuthChallenge` utilizza il nome di una sfida come input e genera la sfida e i parametri per valutare la risposta. Quando `DefineAuthChallenge` restituisce `CUSTOM_CHALLENGE` come sfida sucessiva, il flusso di autenticazione chiama `CreateAuthChallenge`. Il trigger Lambda `CreateAuthChallenge` passa il tipo di sfida successiva nel parametro di metadati della richiesta.
- La funzione Lambda `VerifyAuthChallengeResponse` valuta la risposta e restituisce un valore booleano per indicare se la risposta era valida.

Un flusso di autenticazione personalizzato può anche utilizzare una combinazione di sfide incorporate, come la verifica della password SRP e MFA tramite SMS. Può utilizzare sfide personalizzate come CAPTCHA o domande segrete.

Utilizzare la verifica della password SRP nel flusso di autenticazione personalizzato

Se intendi includere la verifica SRP in un flusso di autenticazione personalizzato, è necessario partire da essa.

- Per avviare la verifica della password SRP in un flusso personalizzato, l'app chiama `InitiateAuth` con `CUSTOM_AUTH` come `Authflow`. Nella mappa `AuthParameters`, la richiesta dalla tua app include `SRP_A:` (il valore SRP A) e `CHALLENGE_NAME: SRP_A`.
- Il flusso `CUSTOM_AUTH` richiama il trigger Lambda `DefineAuthChallenge` con una sessione iniziale di `challengeName: SRP_A` e `challengeResult: true`. La tua funzione Lambda risponde con `challengeName: PASSWORD_VERIFIER`, `issueTokens: false` e `failAuthentication: false`.
- L'app deve quindi chiamare `RespondToAuthChallenge` con `challengeName: PASSWORD_VERIFIER` e gli altri parametri necessari per SRP nella mappa `challengeResponses`.

- Se Amazon Cognito verifica la password, `RespondToAuthChallenge` richiama il trigger Lambda `DefineAuthChallenge` con una seconda sessione di `challengeName: PASSWORD_VERIFIER` e `challengeResult: true`. A quel punto, il trigger `DefineAuthChallenge` Lambda risponde con `challengeName: CUSTOM_CHALLENGE` per avviare la richiesta personalizzata.
- Se MFA è abilitato per un utente, dopo che Amazon Cognito verifica la password, all'utente viene richiesto di configurare o accedere con MFA.

> **Note**
>
> La pagina Web di accesso ospitata di Amazon Cognito non può attivare i Trigger Lambda di richieste di autenticazione personalizzate.

Per ulteriori informazioni sui trigger lambda, compreso il codice di esempio, consulta Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda.

Flusso di autenticazione per la migrazione degli utenti

Il trigger Lambda per la migrazione degli utenti aiuta a migrare gli utenti da un sistema legacy di gestione degli utenti al bacino d'utenza. Se scegli il flusso di autenticazione `USER_PASSWORD_AUTH`, gli utenti non dovranno reimpostare le password durante la migrazione. Questo flusso invia le password degli utenti al servizio tramite una connessione SSL crittografata durante l'autenticazione.

Una volta completata la migrazione di tutti gli utenti, passa i flussi al flusso SRP più sicuro. Il flusso SRP non invia le password sulla rete.

Per ulteriori informazioni sui trigger Lambda, consulta Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda.

Per ulteriori informazioni sulla migrazione degli utenti con un trigger Lambda, consulta Importazione di utenti con un trigger Lambda per la migrazione di utenti.

Modelli di autorizzazione per l'autenticazione tramite API e SDK

Quando inizi lo sviluppo di un'applicazione con l'autenticazione dei pool di utenti, devi scegliere il modello di autorizzazione API più adatto al tipo di applicazione. Un modello di autorizzazione è un sistema per fornire l'autorizzazione a effettuare richieste con i componenti di autenticazione nei

pool di utenti di Amazon Cognito, nelle integrazioni API e SDK. Amazon Cognito ha tre modelli di autorizzazione: IAM-authorized pubblica e autorizzata tramite token.

Per quanto riguarda IAM-authorized le richieste, l'autorizzazione deriva da una firma mediante un set di credenziali AWS IAM nell'`Authorization`intestazione della richiesta. Per le applicazioni lato server, questa pratica protegge le operazioni di autenticazione con l'autorizzazione IAM. Con le richieste di autenticazione pubbliche (non autenticate), non è richiesta alcuna autorizzazione. È adatto per le applicazioni lato client distribuite agli utenti. Nelle operazioni autorizzate da token, in genere implementate in combinazione con operazioni pubbliche, l'autorizzazione proviene da un token di sessione o da un token di accesso incluso nell'`Authorization`intestazione della richiesta. L'autenticazione di Amazon Cognito richiede in genere l'implementazione di due o più operazioni API in ordine e le operazioni API utilizzate dipendono dalle caratteristiche dell'applicazione. I client pubblici, in cui l'applicazione viene distribuita agli utenti, utilizzano operazioni pubbliche, in cui le richieste di accesso non richiedono l'autorizzazione. Token-authorized le operazioni continuano la sessione degli utenti nelle applicazioni pubbliche. Server-side i client, in cui la logica dell'applicazione è ospitata su un sistema remoto, proteggono le operazioni di autenticazione con l'autorizzazione IAM per le richieste di accesso. Le coppie di operazioni API che seguono e i metodi SDK corrispondenti vengono mappate ai modelli di autorizzazione disponibili.

Ogni operazione di autenticazione pubblica ha una qualche forma di equivalente lato server, ad esempio e. UpdateUserAttributesAdminUpdateUserAttributes Sebbene le operazioni lato client siano avviate dall'utente e richiedano una conferma, le operazioni lato server presuppongono che la modifica sia stata apportata da un amministratore del pool di utenti e che le modifiche abbiano effetto immediato. In questo esempio, Amazon Cognito invia un messaggio con un codice di conferma all'utente e il token di accesso dell'utente autorizza una VerifyUserAttributerichiesta che invia il codice. L'applicazione lato server può impostare immediatamente il valore di qualsiasi attributo, sebbene valgano considerazioni speciali per la modifica del valore degli indirizzi e-mail e dei numeri di telefono quando vengono utilizzati per l'accesso.

Per confrontare l'autenticazione API e visualizzare un elenco completo delle operazioni API e dei relativi modelli di autorizzazione, consulta. Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite

Client-side (public) authentication

Di seguito è riportata una tipica sequenza di richieste in un'applicazione lato client

1. L'InitiateAuthoperazione pubblica invia credenziali primarie come nome utente e password.

2. L'RespondToAuthChallengeoperazione autorizzata dal token invia un token di sessione dalla `InitiateAuth` risposta e dalla risposta a una sfida, ad esempio MFA. L'autorizzazione tramite token di sessione indica le richieste che fanno parte di cicli di autenticazione non ancora completi.
3. L'ConfirmDeviceoperazione autorizzata dal token invia un token di accesso ed esegue l'operazione di scrittura di aggiunta di un dispositivo ricordato al profilo dell'utente. L'autorizzazione tramite token di accesso indica le richieste relative alle operazioni self-service degli utenti dopo che hanno completato l'autenticazione.

Per ulteriori informazioni, consulta Client-side opzioni di autenticazione e Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite.

Server-side authentication

Di seguito è riportata una sequenza tipica di richieste provenienti da un'operazione lato server. Ogni richiesta ha un'intestazione di autorizzazione AWS Signature Version 4 firmata con le credenziali della macchina IAM che sono state emesse al server delle applicazioni.

1. L'AdminInitiateAuthoperazione invia credenziali primarie come nome utente e password.
2. AdminRespondToAuthChallengel'operazione presenta la risposta a una sfida, ad esempio l'MFA.
3. L'AdminUpdateDeviceStatusoperazione imposta la chiave del dispositivo in base alla `AdminInitiateAuth` risposta memorizzata.

Per ulteriori informazioni, consulta Server-side opzioni di autenticazione e Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite.

L'autenticazione di un utente viene effettuata rispondendo a richieste successive fino a che l'autenticazione non riesce o Amazon Cognito emette token per l'utente. Puoi ripetere questi passaggi con Amazon Cognito, in un processo che include richieste di autenticazione di tipo diverso, per supportare qualsiasi flusso di autenticazione personalizzato.

Argomenti

- Server-side opzioni di autenticazione
- Client-side opzioni di autenticazione
- Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite

- Elenco delle operazioni API raggruppate per modello di autorizzazione

Server-side opzioni di autenticazione

Le applicazioni Web e altre applicazioni lato server implementano l'autenticazione su un server remoto che un client carica in un'applicazione di visualizzazione remota come un browser o una sessione SSH. Server-side le applicazioni hanno in genere le seguenti caratteristiche.

- Sono integrate in un'applicazione installata su un server in linguaggi come Java, Ruby o Node.js.
- Si connettono a client di app con pool di utenti che potrebbero avere un client segreto, chiamato client riservati.
- Hanno accesso alle AWS credenziali.
- Richiamano l'accesso gestito per l'autenticazione o utilizzano IAM-authorized le operazioni nell'API dei pool di utenti con un AWS SDK.
- Servono clienti interni e potrebbero servire clienti pubblici.

Server-side le operazioni con l'API dei pool di utenti possono utilizzare password, password monouso o passkey come fattore di accesso principale. Per le app lato server, l'autenticazione dei bacini d'utenza è analoga a quella per le app lato client, con le seguenti eccezioni:

- L'app lato server effettua una richiesta API. AdminInitiateAuth Questa operazione richiede AWS credenziali con autorizzazioni che includono e. `cognito-idp:AdminInitiateAuth` `cognito-idp:AdminRespondToAuthChallenge` L'operazione restituisce il risultato della sfida o dell'autenticazione richiesto.
- Quando l'applicazione riceve una sfida, effettua una richiesta AdminRespondToAuthChallengeAPI. Il funzionamento dell'`AdminRespondToAuthChallenge`API richiede anche AWS credenziali.

Per ulteriori informazioni sulla firma delle richieste API di Amazon Cognito con AWS credenziali, consulta il processo di firma Signature versione 4 nella AWS Guida generale.

Nella risposta di `AdminInitiateAuth` `ChallengeParameters`, l'attributo `USER_ID_FOR_SRP`, se presente, contiene il nome utente reale dell'utente e non un alias (come l'indirizzo e-mail o il numero di telefono). Nella chiamata a `AdminRespondToAuthChallenge`, in `ChallengeResponses`, è necessario passare questo nome utente nel parametro `USERNAME`.

> **Note**
>
> Poiché le implementazioni di amministrazione di back-end utilizzano il flusso di autenticazione dell'amministratore, il flusso non supporta i dispositivi ricordati. Quando abiliti il tracciamento dei dispositivi, l'autenticazione di amministrazione avviene correttamente, ma le chiamate per aggiornare i token di accesso non andranno a buon fine.

Client-side opzioni di autenticazione

Le app mobili e altri tipi di applicazioni lato client vengono installate sui dispositivi degli utenti ed eseguono la logica di autenticazione e l'interfaccia utente localmente. In genere presentano le seguenti caratteristiche.

- Sono costruiti in linguaggi come React native, Flutter e Swift e vengono distribuiti sui dispositivi degli utenti.
- Si connettono a client di app con pool di utenti che non dispongono di un client segreto, chiamati client pubblici.
- Non hanno accesso alle AWS credenziali che autorizzerebbero le richieste IAM-authorized API.
- Richiamano l'accesso gestito per l'autenticazione o utilizzano operazioni pubbliche e autorizzate da token nell'API dei pool di utenti con un SDK. AWS
- Servono clienti pubblici e consentono a chiunque di registrarsi e accedere.

Client-side le operazioni con l'API dei pool di utenti possono utilizzare password, password monouso o password come fattore di accesso principale. Il seguente processo funziona per le app lato client dell'utente create con AWS Amplify o SDK AWS.

1. L'utente inserisce il nome utente e la password nell'app.
2. L'app chiama l'operazione `InitiateAuth` con il nome utente e i dettagli SRP (Secure Remote Password) dell'utente.

   Questa operazione API restituisce i parametri di autenticazione.

> **Note**
>
> L'app genera i dettagli SRP con le funzionalità SRP di Amazon Cognito integrate negli SDK AWS .

3. L'app chiama l'operazione `RespondToAuthChallenge`. Se la chiamata va a buon fine, Amazon Cognito restituisce i token dell'utente e il flusso di autenticazione è completo.

   Se Amazon Cognito richiede un ulteriore fattore di autenticazione, la chiamata a `RespondToAuthChallenge` non restituisce token. Invece, la chiamata restituisce una sessione.

4. Se `RespondToAuthChallenge` restituisce una sessione, l'app chiama di nuovo `RespondToAuthChallenge`, questa volta con la sessione e la risposta alla richiesta di identificazione (ad esempio, il codice MFA).

Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite

I pool di utenti di Amazon Cognito sono una combinazione di diverse tecnologie di autenticazione. Si affidano a provider di identità esterni ()IdPs. Si tratta IdPs di applicazioni che implementano l'autenticazione con gli SDK OpenID Connect (OIDC). Forniscono l'autenticazione come emittenti di token web JSON (JWT) simili all'autenticazione OIDC, ma con metodi API che fanno parte degli SDK. AWS Possono anche essere punti di accesso sicuri alle tue applicazioni.

Quando desideri eseguire la registrazione, l'accesso e la gestione di utenti nel pool di utenti, puoi utilizzare due opzioni.

1. Le tue pagine di accesso gestito e la classica interfaccia utente ospitata includono gli endpoint interattivi con accesso gestito e gli endpoint federativi che gestiscono i ruoli IdP e relying-party. Costituiscono un pacchetto di pagine web pubbliche che vengono attivate da Amazon Cognito quando scegli un dominio per il pool di utenti. Per iniziare rapidamente a utilizzare le funzionalità di autenticazione e autorizzazione dei pool di utenti di Amazon Cognito, comprese le pagine per la registrazione, l'accesso, la gestione delle password e l'autenticazione a più fattori (MFA), utilizza l'interfaccia utente integrata dell'accesso gestito.

   Gli altri endpoint del pool di utenti facilitano l'autenticazione con provider di identità di terze parti (). IdPs I servizi che vengono eseguiti includono quanto segue.

a. Service-provider endpoint di callback per reclami autenticati provenienti da te IdPs, like e. `saml2/idpresponse` `oauth2/idpresponse` Quando Amazon Cognito è un provider di servizi (SP) intermedio tra l'app e l'IdP, gli endpoint di callback rappresentano il servizio.

b. Endpoint che forniscono informazioni sull'ambiente, ad esempio `oauth2/userInfo` e `/.well-known/jwks.json`. La tua app utilizza questi endpoint quando verifica i token o recupera i dati del profilo utente con le librerie per sviluppatori OIDC o OAuth 2.0.

2. L'API dei pool di utenti di Amazon Cognito è un set di strumenti per la tua app web o mobile per autenticare gli utenti dopo aver raccolto le informazioni di accesso nel tuo front-end personalizzato. L'autenticazione API dei pool di utenti produce i seguenti token web JSON.

   a. Un token di identità con richieste di attributi verificabili da parte dell'utente.

   b. Un token di accesso che autorizza l'utente a creare richieste API autorizzate da token a un endpoint di servizio AWS.

   > **Note**
   >
   > Per impostazione predefinita, i token di accesso dell'autenticazione API dei pool di utenti contengono solo l'ambito `aws.cognito.signin.user.admin`. Se desideri generare un token di accesso con altri ambiti, ad esempio per autorizzare una richiesta a un'API di terze parti, richiedi gli ambiti durante l'autenticazione tramite gli endpoint del pool di utenti o aggiungi ambiti personalizzati in un Trigger Lambda di pre-generazione del token. La personalizzazione dei token di accesso aggiunge costi alla bolletta. AWS

   c. Un token di aggiornamento che autorizza le richieste di nuovi ID e token di accesso e aggiorna l'identità dell'utente e le proprietà di controllo degli accessi.

Puoi collegare un utente federato, che normalmente accede tramite gli endpoint dei pool di utenti, con un utente il cui profilo è locale al pool di utenti. Un utente locale esiste esclusivamente nella directory del pool di utenti senza federazione tramite un IdP esterno. Se colleghi la loro identità federata a un utente locale in una richiesta AdminLinkProviderForUserAPI, possono accedere con l'API dei pool di utenti. Per ulteriori informazioni, consulta Collegamento di utenti federati a un profilo utente esistente.

L'API dei pool di utenti di Amazon Cognito ha un doppio scopo.

1. Consente di creare e configurare le risorse dei pool di utenti di Amazon Cognito. Ad esempio, puoi creare pool di utenti, aggiungere AWS Lambda trigger e configurare il dominio del pool di utenti che ospita le tue pagine di accesso gestite.

2. Esegue la registrazione, l'accesso e altre operazioni utente per utenti locali e collegati.

Scenario di esempio con l'API dei pool di utenti di Amazon Cognito

1. L'utente seleziona il pulsante "Crea un account" creato nell'app. Inserisce un indirizzo e-mail e una password.
2. La tua app invia una richiesta SignUpAPI e crea un nuovo utente nel tuo pool di utenti.
3. L'app richiede all'utente un codice di conferma e-mail. Gli utenti inseriscono il codice ricevuto in un messaggio di posta elettronica.
4. L'app invia una richiesta ConfirmSignUpAPI con il codice di conferma dell'utente.
5. L'app richiede all'utente il nome utente e password. Le relative informazioni vengono inserite.
6. L'app invia una richiesta InitiateAuthAPI e memorizza un token ID, un token di accesso e un token di aggiornamento. L'app chiama le librerie OIDC per gestire i token di un utente e mantenere persistente la sessione dell'utente.

Nell'API dei pool di utenti di Amazon Cognito, non è possibile consentire l'accesso agli utenti che eseguono la federazione tramite un IdP. Questi utenti devono essere autenticati tramite gli endpoint del pool di utenti. Per ulteriori informazioni sugli endpoint del pool di utenti che includono l'accesso gestito, consulta. Endpoint del pool di utenti e riferimento per l'accesso gestito

I tuoi utenti federati possono iniziare dall'accesso gestito e selezionare il loro IdP, oppure puoi saltare l'accesso gestito e inviare gli utenti direttamente al tuo IdP per l'accesso. Quando la richiesta API a Endpoint Authorize include un parametro IdP, Amazon Cognito reindirizza automaticamente l'utente alla pagina di accesso IdP.

Scenario di esempio con pagine di accesso gestite

1. L'utente seleziona il pulsante "Crea un account" creato nell'app.
2. L'accesso gestito presenta all'utente un elenco dei provider di identità social a cui sono state registrate le credenziali di sviluppatore. L'utente sceglie Apple.
3. L'app avvia una richiesta a Endpoint Authorize con il nome di provider `SignInWithApple`.
4. Il browser dell'utente apre la pagina di autenticazione Apple. L'utente accede e sceglie di autorizzare Amazon Cognito a leggere le informazioni del proprio profilo.
5. Amazon Cognito conferma il token di accesso Apple ed esegue una query sul profilo Apple dell'utente.

6. L'utente presenta un codice di autorizzazione Amazon Cognito all'app.
7. La libreria OIDC dell'applicazione scambia il codice di autorizzazione con Endpoint Token e memorizza un token ID, un token di accesso e un token di aggiornamento emessi dal pool di utenti. La tua app utilizza le librerie OIDC per gestire i token dell'utente e mantenere una sessione persistente per quell'utente.

L'API dei pool di utenti e le pagine di accesso gestito supportano una varietà di scenari, descritti in questa guida. Nelle sezioni seguenti viene illustrato in che modo l'API dei pool di utenti si divide ulteriormente in classi che supportano i requisiti di registrazione, accesso e gestione delle risorse.

Elenco delle operazioni API raggruppate per modello di autorizzazione

L'API dei pool di utenti di Amazon Cognito, un'interfaccia di gestione delle risorse e un'interfaccia di autenticazione e autorizzazione lato utente, combina i modelli di autorizzazione che seguono le relative operazioni. A seconda dell'operazione API, potrebbe essere necessario fornire l'autorizzazione con credenziali IAM, un token di accesso, un token di sessione, un segreto del client o una combinazione di questi. Per molte operazioni di autenticazione e autorizzazione utente, è possibile scegliere tra versioni autenticate e non autenticate della richiesta. Le operazioni non autenticate sono best practice di sicurezza per le app distribuite agli utenti, come app per dispositivi mobili; non è necessario includere alcun segreto nel codice.

È possibile assegnare autorizzazioni solo nelle policy IAM per IAM-authorized operazioni di gestione e IAM-authorized operazioni degli utenti.

IAM-authorized operazioni di gestione

IAM-authorized le operazioni di gestione modificano e visualizzano il pool di utenti e la configurazione del client dell'app, come faresti in Console di gestione AWS.

Ad esempio, per modificare il pool di utenti in una richiesta UpdateUserPoolAPI, è necessario presentare AWS le credenziali e le autorizzazioni IAM per aggiornare la risorsa.

Per autorizzare queste richieste in AWS Command Line Interface (AWS CLI) o in un AWS SDK, configura il tuo ambiente con variabili di ambiente o una configurazione client che aggiunga credenziali IAM alla tua richiesta. Per ulteriori informazioni, consulta Accesso AWS tramite AWS credenziali in. Riferimenti generali di AWS Puoi anche inviare richieste direttamente agli endpoint del servizio per l'API dei pool di utenti Amazon Cognito. È necessario autorizzare o firmare queste richieste con AWS credenziali da incorporare nell'intestazione della richiesta. Per ulteriori informazioni, consulta Firmare le richieste API. AWS

IAM-authorized operazioni di gestione

AddCustomAttributes

CreateGroup

CreateIdentityProvider

CreateResourceServer

CreateUserImportJob

CreateUserPool

CreateUserPoolClient

CreateUserPoolDomain

DeleteGroup

DeleteIdentityProvider

DeleteResourceServer

DeleteUserPool

DeleteUserPoolClient

DeleteUserPoolDomain

DescribeIdentityProvider

DescribeResourceServer

DescribeRiskConfiguration

DescribeUserImportJob

DescribeUserPool

DescribeUserPoolClient

DescribeUserPoolDomain

IAM-authorized operazioni di gestione

GetCSVHeader

GetGroup

GetIdentityProviderByIdentifier

GetSigningCertificate

GetUICustomization

GetUserPoolMfaConfig

ListGroups

ListIdentityProviders

ListResourceServers

ListTagsForResource

ListUserImportJobs

ListUserPoolClients

ListUserPools

ListUsers

ListUsersInGroup

SetRiskConfiguration

SetUICustomization

SetUserPoolMfaConfig

StartUserImportJob

StopUserImportJob

TagResource

IAM-authorized operazioni di gestione

UntagResource

UpdateGroup

UpdateIdentityProvider

UpdateResourceServer

UpdateUserPool

UpdateUserPoolClient

UpdateUserPoolDomain

IAM-authorized operazioni degli utenti

IAM-authorized operazioni utente: registrazione, accesso, gestione delle credenziali, modifica e visualizzazione degli utenti.

Ad esempio, un livello applicazione lato server potrebbe eseguire il backup di un front-end Web. L'app lato server è un client riservato OAuth considerato affidabile con accesso privilegiato alle risorse Amazon Cognito. Per registrare un utente nell'app, il server può includere AWS credenziali in una richiesta AdminCreateUserAPI. Per ulteriori informazioni sui tipi di client OAuth, consulta la sezione relativa ai tipi di client in The OAuth 2.0 Authorization Framework.

Per autorizzare queste richieste nell'SDK AWS CLI o in un AWS SDK, configura l'ambiente dell'app lato server con variabili di ambiente o una configurazione client che aggiunga credenziali IAM alla tua richiesta. Per ulteriori informazioni, consulta Accesso AWS tramite credenziali in. AWS Riferimenti generali di AWS Puoi anche inviare richieste direttamente agli endpoint del servizio per l'API dei pool di utenti Amazon Cognito. È necessario autorizzare o firmare queste richieste con AWS credenziali da incorporare nell'intestazione della richiesta. Per ulteriori informazioni, consulta Firmare le richieste API. AWS

Se il client dell'app dispone di un segreto del client, devi fornire le tue credenziali IAM e, a seconda dell'operazione, il parametro `SecretHash` o il valore `SECRET_HASH` in `AuthParameters`. Per ulteriori informazioni, consulta Calcolo dei valori SecretHash.

IAM-authorized operazioni utente

AdminAddUserToGroup

AdminConfirmSignUp

AdminCreateUser

AdminDeleteSoftwareToken

AdminDeleteUser

AdminDeleteUserAttributes

AdminDisableProviderForUser

AdminDisableUser

AdminEnableUser

AdminForgetDevice

AdminGetDevice

AdminGetUser

AdminGetUserAuthFactors

AdminInitiateAuth

AdminLinkProviderForUser

AdminListDevices

AdminListGroupsForUser

AdminListUserAuthEvents

AdminRemoveUserFromGroup

AdminResetUserPassword

AdminRespondToAuthChallenge

IAM-authorized operazioni utente

AdminSetUserMFAPreference

AdminSetUserPassword

AdminSetUserSettings

AdminUpdateAuthEventFeedback

AdminUpdateDeviceStatus

AdminUpdateUserAttributes

AdminUserGlobalSignOut

Operazioni utente non autenticate

Operazioni utente non autenticate: registrazione, accesso e avvio del ripristino delle password per gli utenti. Utilizza operazioni API non autenticate, o pubbliche, per consentire a chiunque su Internet di eseguire la registrazione e l'accesso all'app.

Ad esempio, per registrare un utente nell'app, puoi distribuire un client pubblico OAuth che non fornisce alcun accesso privilegiato ai segreti. È possibile registrare questo utente con l'operazione API non autenticata. SignUp

Per inviare queste richieste in un client pubblico sviluppato con un AWS SDK, non è necessario configurare alcuna credenziale. Puoi anche inviare richieste direttamente agli endpoint del servizio per l'API dei pool di utenti Amazon Cognito senza autorizzazione aggiuntiva.

Se il client dell'app dispone di un segreto del client, devi fornire, a seconda dell'operazione, il parametro `SecretHash` o il valore `SECRET_HASH` in `AuthParameters`. Per ulteriori informazioni, consulta Calcolo dei valori SecretHash.

Operazioni utente non autenticate

SignUp

ConfirmSignUp

Operazioni utente non autenticate

ResendConfirmationCode

ForgotPassword

ConfirmForgotPassword

InitiateAuth

Token-authorized operazioni utente

Token-authorized le operazioni degli utenti si disconnettono, gestiscono le credenziali, modificano e visualizzano gli utenti dopo che hanno effettuato l'accesso o iniziato la procedura di accesso. Utilizza le operazioni API autorizzate tramite token quando non desideri distribuire segreti nella tua app e desideri autorizzare le richieste con le credenziali dell'utente. Se l'utente ha completato l'accesso, devi autorizzare la richiesta API autorizzata tramite token con un token di accesso. Se l'utente sta eseguendo un processo di accesso, devi autorizzare la relativa richiesta API autorizzata tramite token con un token di sessione restituito da Amazon Cognito nella risposta alla richiesta precedente.

Ad esempio, in un client pubblico, potrebbe essere necessario aggiornare il profilo di un utente in modo da limitare l'accesso in scrittura solo al profilo dell'utente. Per effettuare questo aggiornamento, il client può includere il token di accesso dell'utente in una richiesta UpdateUserAttributesAPI.

Per inviare queste richieste in un client pubblico sviluppato con un AWS SDK, non è necessario configurare alcuna credenziale. Includi un parametro `AccessToken` o `Session` nella richiesta. Puoi anche inviare richieste direttamente agli endpoint del servizio per l'API dei pool di utenti Amazon Cognito. Per autorizzare una richiesta a un endpoint del servizio, includi il token di accesso o sessione nel corpo POST della richiesta.

Per firmare una richiesta API per un'operazione autorizzata tramite token, includi il token di accesso come un'intestazione `Authorization` nella richiesta, nel formato `Bearer` *`<Base64-encoded access token>`*.

Token-authorized operazioni utente	AccessToken	Sessione
RespondToAuthChallenge		✓

Token-authorized operazioni utente	AccessToken	Sessione
ChangePassword	✓	
GetUser	✓	
StartWebAuthnRegistration	✓	
CompleteWebAuthnRegistration	✓	
DeleteWebAuthnCredential	✓	
ListWebAuthnCredentials	✓	
UpdateUserAttributes	✓	
DeleteUserAttributes	✓	
DeleteUser	✓	
ConfirmDevice	✓	
ForgetDevice	✓	
GetDevice	✓	
ListDevices	✓	
UpdateDeviceStatus	✓	
GetUserAttributeVerificationCode	✓	
VerifyUserAttribute	✓	
SetUserSettings	✓	

Token-authorized operazioni utente	AccessToken	Sessione
SetUserMFAPreference	✓	
GlobalSignOut	✓	
UpdateAuthEventFeedback		✓
AssociateSoftwareToken	✓	✓
VerifySoftwareToken	✓	✓
RevokeToken[1]		
GetTokensFromRefreshToken[1]		

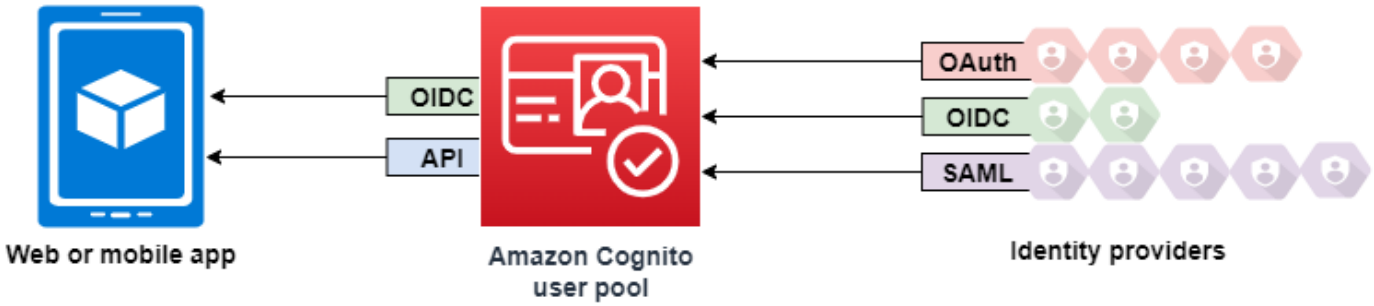
[1] `RevokeToken` e `GetTokensFromRefreshToken` prendi i token di aggiornamento come parametro di autorizzazione. Il token di aggiornamento funge da token di autorizzazione e come risorsa di destinazione.

Accesso al pool di utenti con provider di identità di terze parti

Gli utenti della tua app possono accedere direttamente tramite un pool di utenti oppure possono federarsi tramite un provider di identità (IdP) di terze parti. Il pool di utenti gestisce il sovraccarico di gestione dei token restituiti dall'accesso social tramite Facebook, Google, Amazon e Apple e da OpenID Connect (OIDC) e SAML. IdPs Con l'interfaccia utente web ospitata integrata, Amazon Cognito fornisce la gestione e la gestione dei token per tutti gli utenti autenticati. IdPs In questo modo, i tuoi sistemi di back-end possono standardizzarsi su un set di token per pool di utenti.

Funzionamento dell'accesso federato nei pool di utenti di Amazon Cognito

Sign-in tramite una terza parte (federazione) è disponibile nei pool di utenti di Amazon Cognito. Questa funzione è indipendente dalla federazione tramite pool di identità di Amazon Cognito (identità federate).



Amazon Cognito è una directory utente e un gestore dell'identità digitale (IdP) OAuth 2.0. Quando gli utenti locali effettuano l'accesso alla directory Amazon Cognito, il pool di utenti è un IdP dell'app. Un utente locale esiste esclusivamente nella directory del pool di utenti senza federazione tramite un IdP esterno.

Quando connetti Amazon Cognito a social, SAML o OpenID Connect (OIDC IdPs), il tuo pool di utenti funge da ponte tra più fornitori di servizi e la tua app. Per il gestore dell'identità digitale (IdP) in uso, Amazon Cognito è un fornitore di servizi. Devi IdPs passare un token ID OIDC o un'asserzione SAML ad Amazon Cognito. Amazon Cognito legge le asserzioni relative all'utente nel token o nell'asserzione e mappa tali asserzioni su un nuovo profilo utente nella directory del pool di utenti.

Amazon Cognito crea quindi un profilo utente per l'utente federato nella propria directory. Amazon Cognito aggiunge attributi all'utente in base alle dichiarazioni del tuo IdP e, nel caso di OIDC e provider di identità social, un endpoint pubblico. IdP-operated `userinfo` Gli attributi dell'utente cambiano nel pool di utenti quando un attributo del gestore dell'identità digitale (IdP) mappato cambia. Puoi anche aggiungere altri attributi indipendentemente da quelli del gestore dell'identità digitale (IdP).

Dopo che Amazon Cognito ha creato un profilo per l' utente federato, cambia la sua funzione e si presenta come gestore dell'identità digitale (IdP) all'app, che ora è il fornitore di servizi. Amazon Cognito è una combinazione di gestori dell'identità digitale OIDC e OAuth 2.0. Genera token di accesso, token ID e token di aggiornamento. Per ulteriori informazioni sui token, consulta Comprensione dei token web JSON (JWT) del pool di utenti.

Devi progettare un'app che si integri con Amazon Cognito per autenticare e autorizzare gli utenti, che siano federati o nativi.

Responsabilità di un'app come provider di servizi con Amazon Cognito

Verifica ed elaborazione delle informazioni nei token

Nella maggior parte degli scenari, Amazon Cognito reindirizza l'utente autenticato a un URL dell'app aggiunta con un codice di autorizzazione. L'app scambia il codice per l'accesso, l'ID e l'aggiornamento dei token. Deve quindi verificare la validità dei token e inviare informazioni all'utente in base alle asserzioni contenute nei token.

Risposta agli eventi di autenticazione con le richieste API di Amazon Cognito

L'app deve integrarsi con le API dei pool di utenti di Amazon Cognito e gliendpoint dell'API di autenticazione. L'API di autenticazione effettua l'accesso e la disconnessione dell'utente e gestisce i token. L'API dei pool di utenti dispone di una serie di operazioni che gestiscono il pool di utenti, gli utenti e la sicurezza dell'ambiente di autenticazione. L'app deve sapere cosa fare dopo aver ricevuto una risposta da Amazon Cognito.

Informazioni importanti sull'accesso di terze parti ai pool di utenti di Amazon Cognito

- Se i tuoi utenti accedono con provider federati, devi scegliere un dominio. Questo configura le pagine per l'accesso gestito. Per ulteriori informazioni, consulta Utilizzo del proprio dominio per l'accesso gestito.
- Non puoi accedere a utenti federati con operazioni API come InitiateAuthe AdminInitiateAuth. Quando utilizzi i flussi di autenticazione integrati di Amazon Cognito, gli utenti federati possono accedere solo con il o ilEndpoint Login. Endpoint Authorize Se lo usi`CUSTOM_AUTH`, i trigger controllano il risultato. Sono responsabili se il servizio rilascia un token a qualsiasi utente (federato o meno).
- Endpoint Authorize è un endpoint di reindirizzamento. Se fornisci un `identity_provider` parametro `idp_identifier` or nella tua richiesta, questo reindirizza automaticamente al tuo IdP, ignorando l'accesso gestito. In caso contrario, reindirizza all'accesso gestito. Endpoint Login
- Quando l'accesso gestito reindirizza una sessione a un IdP federato, Amazon Cognito include l'intestazione nella richiesta`user-agent`. `Amazon/Cognito`
- Amazon Cognito deriva l'attributo `username` per un profilo utente federato da una combinazione di un identificatore fisso e il nome del gestore dell'identità digitale (IdP). Per generare un nome utente che soddisfi i tuoi requisiti personalizzati, crea una mappatura all'attributo. `preferred_username` Per ulteriori informazioni, consulta Cose da sapere sulle mappature.

  Ad esempio: `MyIDP_bob@example.com`

- Amazon Cognito crea un gruppo di utenti per ogni OIDC, sAML e IdP social che aggiungi al tuo pool di utenti. Il nome del gruppo è nel formato`[user pool ID]_[IdP name]`, ad esempio o. `us-east-1_EXAMPLE_MYSSO` `us-east-1_EXAMPLE_Google` Ogni profilo utente IdP univoco generato automaticamente viene aggiunto automaticamente a questo gruppo. Gli utenti collegati non vengono aggiunti automaticamente a questo gruppo, ma puoi aggiungere i loro profili al gruppo con un processo separato.
- Amazon Cognito registra le informazioni sull'identità dell'utente federato in un attributo e un'asserzione nel token ID, chiamato `identities`. Questa asserzione contiene il gestore dell'utente e il relativo ID univoco del gestore. Non è possibile modificare l'attributo `identities` direttamente in un profilo utente. Per ulteriori informazioni su come collegare un utente federato, consulta Collegamento di utenti federati a un profilo utente esistente.
- Quando aggiorni il tuo IdP in una richiesta UpdateIdentityProviderAPI, le modifiche possono richiedere fino a un minuto prima che vengano visualizzate nell'accesso gestito.
- Amazon Cognito supporta fino a 20 reindirizzamenti HTTP tra se stesso e l'IdP.
- Quando un utente accede con l'accesso gestito, il suo browser memorizza un cookie di sessione di accesso crittografato che registra il cliente e il provider con cui ha effettuato l'accesso. Se tenta di accedere nuovamente con gli stessi parametri, l'accesso gestito riutilizza qualsiasi sessione esistente non scaduta e l'utente si autentica senza fornire nuovamente le credenziali. Se l'utente accede nuovamente con un IdP diverso, incluso un passaggio da o verso un accesso al pool di utenti locale, deve fornire le credenziali e generare una nuova sessione di accesso.

  Puoi assegnare qualsiasi gruppo di utenti IdPs a qualsiasi client dell'app e gli utenti possono accedere solo con un IdP che hai assegnato al loro client di app.

Argomenti

- Configurazione dei provider di identità per il bacino d'utenza
- Utilizzo di provider di identità social con un pool di utenti
- Utilizzo di provider di identità SAML con un pool di utenti
- Utilizzo di provider di identità OIDC con un pool di utenti
- Mappatura degli attributi IdP su profili e token
- Collegamento di utenti federati a un profilo utente esistente

Configurazione dei provider di identità per il bacino d'utenza

Con i pool di utenti, puoi implementare l'accesso tramite una varietà di provider di identità esterni ()IdPs. Questa sezione della guida contiene istruzioni per configurare questi provider di identità con il tuo pool di utenti nella console Amazon Cognito. In alternativa, puoi utilizzare l'API dei pool di utenti e un AWS SDK per aggiungere in modo programmatico provider di identità per pool di utenti. Per ulteriori informazioni, consulta CreateIdentityProvider.

Le opzioni di provider di identità supportate includono provider di social network come Facebook, Google e Amazon, oltre a provider OpenID Connect (OIDC) e SAML 2.0. Prima di iniziare, configurati con le credenziali amministrative per il tuo IdP. Per ogni tipo di provider, dovrai registrare la tua applicazione, ottenere le credenziali necessarie e quindi configurare i dettagli del provider nel tuo pool di utenti. Gli utenti possono quindi registrarsi e accedere all'applicazione con gli account esistenti dei provider di identità connessi.

Il menu Social e provider esterni in Autenticazione aggiunge e aggiorna il pool di utenti IdPs. Per ulteriori informazioni, consulta Accesso al pool di utenti con provider di identità di terze parti.

Per un IdP OIDC o OAuth, puoi mappare i valori ACR dell'IdP ai livelli di autenticazione di Amazon Cognito con il campo per supportare l'autenticazione avanzata per gli utenti federati. `AcrMapping` Per ulteriori informazioni, consulta Step-up autenticazione per utenti federati.

Argomenti

- Configurazione dell'accesso dell'utente con un IdP social
- Configurazione dell'accesso utente con un IdP OIDC
- Configurazione dell'accesso utente con un IdP SAML

Configurazione dell'accesso dell'utente con un IdP social

Puoi utilizzare la federazione per i bacini d'utenza di Amazon Cognito per integrare i provider di identità social, come Facebook, Google e Login with Amazon.

Per aggiungere un provider di identità social, per prima cosa, crea un account per sviluppatori con il provider di identità. Dopo aver creato un account per sviluppatori, registra l'app con il provider di identità. Il provider di identità crea un'ID e un segreto per l'app, i cui valori devono essere configurati nei bacini d'utenza di Amazon Cognito.

- Piattaforma di identità Google

- Facebook per sviluppatori
- Login with Amazon
- Accedi con Apple

Per integrare l'accesso utente con un IdP social

1. Accedi alla console Amazon Cognito. Se richiesto AWS , inserisci le tue credenziali.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegli il menu Social e provider esterni.
4. Scegli l'opzione Add an identity provider (Aggiungi un provider di identità) oppure seleziona il provider di identità Facebook, Google, Amazon o Apple che hai configurato, individua la voce Identity provider information (Informazioni provider di identità) e seleziona Edit (Modifica). Per ulteriori informazioni su come aggiungere provider di identità social consulta Utilizzo di provider di identità social con un pool di utenti.
5. Inserisci le informazioni del tuo provider di identità social completando uno dei seguenti passaggi, in base alla tua scelta di IdP:

   Per Facebook, Google e Login with Amazon:

   Inserisci l'ID dell'app e il segreto app ricevuti al momento della creazione dell'app client.

   Accedi con Apple

   Inserisci l'ID del servizio fornito ad Apple, nonché l'ID del team, l'ID della chiave e la chiave privata ricevuti quando è stata creato il client dell'app.
6. Nel campo Authorized scopes (Ambiti autorizzati), inserisci i nomi degli ambiti dei provider di identità social che intendi mappare agli attributi del bacino d'utenza. Gli ambiti definiscono gli attributi utente, ad esempio nome ed indirizzo e-mail con cui intendi accedere con l'App. Quando inserisci gli ambiti, usa le seguenti linee guida in base alla tua scelta di IdP:

   - Facebook: ambiti separati da virgole. Ad esempio:

     ```
     public_profile, email
     ```
   - Google, Login with Amazon e Accedi con Apple: ambiti separati da spazi. Ad esempio:
     - Google: `profile email openid`
     - Login with Amazon: `profile postal_code`

- Accedi con Apple: `name email`

> **Note**
>
> Per Accedi con Apple (console), utilizza le caselle di controllo per scegliere gli ambiti.

7. Scegli Save changes (Salva modifiche).
8. Dal menu App client, scegli un client per l'app dall'elenco, quindi seleziona Modifica. Aggiungi il nuovo provider di identità social al client dell'App alla voce Identity providers (Provider di identità).
9. Scegli Save changes (Salva modifiche).

Per ulteriori informazioni sui social IdPs, consultaUtilizzo di provider di identità social con un pool di utenti.

Configurazione dell'accesso utente con un IdP OIDC

Puoi integrare l'accesso degli utenti tramite un provider di identità OpenID Connect (OIDC), ad esempio Salesforce o Ping Identity.

Per aggiungere un provider OIDC a un pool di utenti

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Bacini d'utenza) dal menu di navigazione.
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Social e provider esterni e seleziona Aggiungi un provider di identità.
5. Scegli un provider di identità OpenID Connect.
6. Inserisci un nome univoco nel campo Provider name (Nome provider).
7. Inserisci l'ID client che hai ricevuto dal tuo provider nel campo Client ID (ID client).
8. Inserisci il segreto client che hai ricevuto dal tuo provider nel campo Client secret (Segreto client).
9. Inserisci gli Authorized scopes (Ambiti autorizzati) per questo provider. Gli ambiti definiscono i gruppi di attributi utente (ad esempio `name eemail`) che l'applicazione richiede al provider. Gli ambiti devono essere separati da spazi, secondo la specifica OAuth 2.0.

All'utente viene richiesto il consenso a fornire questi attributi alla tua applicazione.

10. Scegli un Attribute request method (Metodo della richiesta di attributo) per fornire ad Amazon Cognito il metodo HTTP (GET o POST) da utilizzare per recuperare i dettagli dell'utente dall'endpoint userInfo gestito dal tuo provider.
11. Scegli un Setup method (Metodo di impostazione) per recuperare gli endpoint OpenID Connect dall'opzione Auto fill through issuer URL (Riempimento automatico attraverso URL dell'emittente) o da Manual input (Inserimento manuale). Utilizza lo strumento Auto fill through issuer URL (Riempimento automatico dell'URL dell'emittente) quando il tuo provider ha un endpoint pubblico `.well-known/openid-configuration` in cui Amazon Cognito può recuperare gli URL degli endpoint `authorization`, `token`, `userInfo` e `jwks_uri`.
12. Inserisci gli URL dell'emittente o degli endpoint `authorization`, `token`, `userInfo`, e `jwks_uri` dal provider di IdP.

    > **Note**
    >
    > È possibile utilizzare solo i numeri di porta 443 e 80 con URL di rilevamento, compilati automaticamente e inseriti manualmente. Gli accessi utente non riescono se il provider OIDC utilizza porte TCP non standard.
    > L'URL dell'emittente deve iniziare con `https://` e non deve terminare con un carattere `/`. Ad esempio, Salesforce usa questo URL: `https://login.salesforce.com`
    > Il documento `openid-configuration` associato all'URL dell'emittente deve fornire URL HTTPS per i seguenti valori:`authorization_endpoint`, `token_endpoint`, `userinfo_endpoint` e `jwks_uri`. Allo stesso modo, quando scegli Manual input (Inserimento manuale), è possibile inserire solo URL HTTPS.

13. Per impostazione predefinita, la richiesta OIDC sub viene mappata all'attributo del bacino d'utenza Username. Puoi mappare altre richieste OIDC agli attributi del bacino d'utenza. Inserisci l'attestazione OIDC e seleziona l'attributo del pool di utenti corrispondente dall'elenco a discesa. Ad esempio, l'indirizzo e-mail della richiesta viene spesso mappato all'attributo del bacino d'utenza Email (E-mail).
14. Mappa gli attributi aggiuntivi dal provider di identità al bacino d'utenza. Per ulteriori informazioni, consulta la sezione Specificazione di mappature degli attributi del provider di identità per il bacino d'utenza.
15. Scegli Create (Crea).

16. Dal menu App client, seleziona un client di app dall'elenco. Per aggiungere il nuovo provider di identità SAML al client dell'app, vai alla scheda Pagine di accesso e seleziona Modifica nella configurazione delle pagine di accesso gestite.
17. Scegli Save changes (Salva modifiche).

Per ulteriori informazioni su OIDC IdPs, consulta. Utilizzo di provider di identità OIDC con un pool di utenti

Configurazione dell'accesso utente con un IdP SAML

Puoi utilizzare la federazione per il bacino d'utenza di Amazon Cognito per l'integrazione con un provider di identità (IdP) SAML. Puoi fornire un documento di metadati, o caricando il file oppure inserendo un URL di endpoint del documento di metadati. Per informazioni su come ottenere documenti di metadati per IdPs SAML di terze parti, consulta. Configurazione del tuo provider di identità SAML di terze parti

Per configurare un provider di identità SAML 2.0 nel bacino d'utenza

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Social e provider esterno e seleziona Aggiungi un provider di identità.
5. Scegli un provider di identità SAML.
6. Inserisci gli Identifiers (Identificatori) separati da virgole. Un identificatore indirizza Amazon Cognito a controllare l'indirizzo e-mail di accesso dell'utente e poi indirizza l'utente al provider che corrisponde al suo dominio..
7. Scegli Add sign-out flow (Aggiungi flusso di disconnessione) se desideri che Amazon Cognito invii richieste di disconnessione firmate al tuo provider quando un utente si disconnette. Configura il tuo provider di identità SAML 2.0 per inviare risposte di disconnessione all'`https://mydomain.auth.us-east-1.amazoncognito.com/saml2/logout`endpoint creato da Amazon Cognito quando configuri l'accesso gestito. L'endpoint `saml2/logout` utilizza POST vincolanti.

> **Note**
>
> Se selezioni questa opzione e il provider di identità SAML si aspetta una richiesta di disconnessione con firma, è necessario configurare anche il certificato di firma fornito da Amazon Cognito con il tuo IdP SAML.
> L'IdP SAML elabora la richiesta di disconnessione firmata e disconnette l'utente dalla sessione di Amazon Cognito.

8. Scegli una Metadata document source (Fonte del documento di metadati). Se il tuo provider di identità fornisce metadati SAML a un URL pubblico, puoi scegliere l'opzione Metadata document URL (URL del documento di metadati) e inserire l'URL pubblico. In caso contrario, seleziona Upload metadata document (Carica documento di metadati) e seleziona un file di metadati scaricato dal tuo provider in precedenza.

   > **Note**
   >
   > Se il provider ha un endpoint pubblico, suggeriamo di fornire l'URL di un documento di metadati anziché caricare un file. Se utilizzi l'URL, Amazon Cognito aggiorna automaticamente i metadati. In genere, l'aggiornamento dei metadati avviene ogni 6 ore oppure prima della scadenza dei metadati, in base a ciò che avviene prima.

9. Scegli l'opzione Map attributes between your SAML provider and your app (Mappa gli attributi tra il provider SAML e la tua app) per mappare gli attributi del provider SAML al profilo utente nel bacino d'utenza. Includi gli attributi richiesti del bacino d'utenza nella mappa degli attributi.

   Ad esempio, quando seleziona l'`email` dell'Attributo del bacino d'utenza, inserisci il nome dell'attributo SAML così come compare nell'asserzione SAML dal provider di identità. Il tuo provider di identità potrebbe offrire esempi di asserzioni SAML come riferimento. Alcuni provider di identità utilizzano nomi semplici, ad esempio`email`, mentre altri utilizzano URL-formatted nomi di attributi simili a:

   ```
   http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
   ```

10. Scegli Crea.

> **Note**
>
> Se visualizzi `InvalidParameterException` durante la creazione di un provider di identità SAML con un URL di endpoint di metadati HTTPS, assicurati che l'endpoint di metadati abbia l'SSL configurato correttamente e che vi sia associato un certificato SSL valido. Un esempio di tale eccezione potrebbe essere «Errore nel recupero dei metadati da». *<metadata endpoint>*

Per configurare l'IdP SAML per l'aggiunta di un certificato di firma

- Per ottenere il certificato contenente la chiave pubblica utilizzata dall'IdP per verificare la richiesta di disconnessione firmata, procedi come segue:
  1. Vai al menu Social e provider esterni del tuo pool di utenti.
  2. Seleziona il tuo provider SAML,
  3. Scegli Visualizza certificato di firma.

Per ulteriori informazioni su SAML, IdPs consultaUtilizzo di provider di identità SAML con un pool di utenti.

Utilizzo di provider di identità social con un pool di utenti

Gli utenti delle tue app web e mobili possono accedere tramite provider di identità social (IdP) come Google, Facebook, Amazon e Apple. Abilita l'accesso gestito e Amazon Cognito aggiunge un pulsante di accesso per ogni IdP social alle tue pagine di accesso gestite. Amazon Cognito gestisce gli scambi OAuth 2.0 e OpenID Connect con il provider e fornisce all'app un set standard di token per pool di utenti, in modo che i sistemi di back-end possano standardizzare i token del pool di utenti indipendentemente da come l'utente ha effettuato l'accesso. Per ulteriori informazioni sugli endpoint creati da Amazon Cognito per questi scambi, consulta il riferimento all'API di autenticazione dei pool di utenti di Amazon Cognito.

Ogni provider ha la propria console per sviluppatori, i propri valori da inserire e le proprie insidie. Scegli il tuo provider per istruzioni dettagliate:

- Aggiungere Google come fornitore di identità social
- Aggiungi Accedi con Apple come provider di identità social
- Aggiungere Facebook come fornitore di identità social

- Aggiungi Login with Amazon come provider di identità social

> **Note**
>
> Sign-in tramite una terza parte (federazione) è disponibile nei pool di utenti di Amazon Cognito. Questa funzione è indipendente dalla federazione tramite pool di identità di Amazon Cognito (identità federate).

Argomenti

- Aggiungere Google come fornitore di identità social
- Aggiungi Accedi con Apple come provider di identità social
- Aggiungere Facebook come fornitore di identità social
- Aggiungi Login with Amazon come provider di identità social

Aggiungere Google come fornitore di identità social

> **Note**
>
> Ultima verifica sulla console del provider: 2 ottobre 2026. I passaggi della console Google seguono la piattaforma di autenticazione di Google (le sezioni Branding, Audience, Data Access e Clients). Se la console Google è diversa da questi passaggi, consulta la documentazione di Google Auth Platform.

Per consentire ai tuoi utenti di accedere con Google, registri un client OAuth nella piattaforma Google Auth, quindi aggiungi Google come provider di identità (IdP) nel tuo pool di utenti. Devi prima abilitare l'accesso gestito.

Registra un client OAuth nella piattaforma Google Auth

Per registrare un client Google OAuth

1. Accedi alla console Google Cloud e crea o seleziona un progetto.
2. Apri Menu, quindi Google Auth Platform. Se è la prima volta, scegli Inizia nella pagina Panoramica e fornisci il nome dell'app, l'email di assistenza agli utenti, il tipo di pubblico (esterno) e le informazioni di contatto.

3. Scegli Branding. In Domini autorizzati, aggiungi i domini registrabili del tuo pool di utenti, non gli URL completi:

   - `amazoncognito.com`, per il dominio Amazon Cognito predefinito.
   - La radice del tuo dominio personalizzato, ad esempio`example.com`, se ne usi uno.

4. Scegli Accesso ai dati, quindi Aggiungi o rimuovi ambiti. Aggiungi `openid`, `.../auth/userinfo.email` e `.../auth/userinfo.profile`.
5. Scegli Clienti, quindi Crea cliente. Per Tipo di applicazione, scegli Applicazione Web.
6. In JavaScript Origini autorizzate, inserisci il dominio del pool di utenti senza percorso:

   - Dominio Amazon Cognito predefinito: `https://<your-prefix>.auth.<region>.amazoncognito.com`
   - Dominio personalizzato: `https://auth.example.com`

7. In URI di reindirizzamento autorizzati, inserisci l'`/oauth2/idpresponse`endpoint del dominio del tuo pool di utenti:

   - Dominio Amazon Cognito predefinito: `https://<your-prefix>.auth.<region>.amazoncognito.com/oauth2/idpresponse`
   - Dominio personalizzato: `https://auth.example.com/oauth2/idpresponse`

8. Scegli Crea. Archivia in modo sicuro l'ID cliente e il segreto del cliente visualizzati da Google. Inserisci questi valori quando aggiungi Google al tuo pool di utenti.

> **Note**
>
> Per i clienti creati dopo giugno 2025, Google mostra il segreto del client solo una volta. Conservalo immediatamente.

Aggiungi Google al tuo pool di utenti

Per aggiungere Google come IdP nel Console di gestione AWS

1. Nella console Amazon Cognito, scegli il tuo pool di utenti, poi Social e provider esterni, quindi Aggiungi un provider di identità.
2. Seleziona Google. Inserisci l'ID cliente e il segreto del cliente dalla piattaforma Google Auth.
3. Per gli ambiti autorizzati, inserisci`profile email openid`, separati da spazi.

4. Associa gli attributi di Google che desideri agli attributi del tuo pool di utenti. Come minimo, esegui la `email` mappatura da `email` e `email_verified` versoemail_verified. Se non esegui la mappaturaemail_verified, Amazon Cognito considera l'e-mail come non verificata.
5. Scegli Aggiungi provider di identità e abilita Google sul client dell'app.

Prova l'accesso a Google

Apri la pagina di accesso gestito e scegli Continua con Google oppure apri direttamente l'endpoint di autorizzazione:

```
https://<your-prefix>.auth.<region>.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=<app-client-id>&redirect_uri=<your-app-
callback>&identity_provider=Google
```

Checkpoint

Il browser completa la selezione dell'account Google e torna all'app con un codice `/oauth2/idpresponse` di autorizzazione. In Gestione utenti, viene visualizzato un nuovo utente federato con il provider Googleemail, popolato e impostato su. `email_verified` `true` Non dovresti visualizzare alcun errore. `redirect_uri_mismatch`

Google-specific insidie

Limite utente della schermata di consenso non verificato

> Un'app con lo stato di pubblicazione Testing è limitata agli utenti di test che elenchi. Un'app che mostra la schermata dell'app non verificata ha un limite di 100 nuovi utenti per tutta la durata del progetto e il limite non può essere ripristinato. Tuttavia, se l'app richiede solo gli ambiti di accesso di base (`openid`,,`profile`)`email`, non è necessario che gli utenti di test siano elencati, non visualizzino alcun avviso e le loro autorizzazioni non scadono. Rispetta questi ambiti per evitare il limite o verifica la tua app nel Centro di verifica.

Redirect-URI propagazione

> Le modifiche alle origini autorizzate e agli URI di reindirizzamento possono richiedere da cinque minuti a qualche ora per avere effetto. Viene restituita una mancata corrispondenza`redirect_uri_mismatch`; viene restituita un'origine non registrata. `origin_mismatch`

email_verify e collegamento dell'account

Google afferma `email_verified=true` per gli account Gmail e Google Workspace. Amazon Cognito non unisce automaticamente un utente Google con un utente esistente che ha lo stesso indirizzo email. Per collegarli, implementa il collegamento (ad esempio, con un trigger Lambda e prima della registrazione`AdminLinkProviderForUser`) e collega solo su un valore attendibile. `email_verified`

Aggiungi Accedi con Apple come provider di identità social

> **Note**
>
> Ultima verifica sulla console del provider: 2 ottobre 2026. Apple Steps utilizza certificati, identificatori e profili nell'account Apple Developer. Devi avere il ruolo di Titolare dell'account o Amministratore.

L'accesso con Apple richiede quattro elementi: un ID app, un ID Servizi (l'ID cliente Amazon Cognito), una chiave privata e il tuo ID team. Quindi aggiungi Apple come provider di identità (IdP) nel tuo pool di utenti. Devi prima abilitare l'accesso gestito.

Crea i tuoi identificatori e la tua chiave Apple

Per creare gli artefatti Apple

1. In Certificati, identificatori e profili, crea un ID app (identificatori, quindi +, quindi ID app, quindi app) e abilita la funzionalità Accedi con Apple.
2. Crea un ID di servizio (identificatori, quindi +, quindi ID servizi). Inserisci una descrizione e un identificatore di dominio inverso, ad esempio. `com.example.auth.cognito` Questo identificatore è il tuo ID Amazon Cognito Services (ID cliente).
3. Seleziona l'ID dei servizi, abilita Accedi con Apple e scegli Configura. Scegli l'ID dell'app principale, quindi inserisci gli URL web:

   - Domini e sottodomini (solo host, senza schema): `<your-prefix>.auth.<region>.amazoncognito.com` per il dominio Amazon Cognito predefinito o `auth.example.com` per un dominio personalizzato.
   - URL di ritorno (URL completo con`https://`): o `https://auth.example.com/oauth2/idpresponse` per un `https://<your-`

`prefix>.auth.<region>.amazoncognito.com/oauth2/idpresponse` dominio personalizzato.

4. Crea una chiave privata (Keys, poi +), abilita Accedi con Apple e scarica il `.p8` file. Puoi scaricarlo solo una volta. Registra l'ID della chiave.
5. Registra il tuo ID di squadra di 10 caratteri dalla pagina Iscrizione. Inserisci l'ID dei servizi, l'ID del team, l'ID chiave e la chiave privata quando aggiungi Apple al tuo pool di utenti.

Aggiungi Apple al tuo pool di utenti

Per aggiungere Apple come IdP nel Console di gestione AWS

1. Nella console Amazon Cognito, scegli il tuo pool di utenti, quindi Social e provider esterni, quindi Aggiungi un provider di identità, quindi Accedi con Apple.
2. Inserisci l'ID dei servizi, l'ID del team, l'ID della chiave e la chiave privata. Amazon Cognito utilizza la chiave privata per generare il segreto del client per te.
3. Per gli ambiti autorizzati, inserisci. `email name` Utilizzalo solo `email` se non ti serve il nome dell'utente.
4. Esegui `email` il `email` mapping `sub` verso e verso l'attributo del nome utente. Apple restituisce il nome dell'utente solo al primo accesso, quindi acquisiscilo se lo mappi.
5. Scegli Aggiungi provider di identità e abilita Apple sul client dell'app.

Prova Accedi con Apple

Apri la pagina di accesso gestito e scegli Continua con Apple. Prova sia Condividi la mia email che Nascondi la mia email.

Checkpoint

Il browser torna `/oauth2/idpresponse` all'app con un codice di autorizzazione. Un utente appare nel tuo pool di utenti con il SignInWithAppleprovider e un`sub`. L'esecuzione Share My Email mostra l'e-mail reale; l'esecuzione Hide My Email mostra un `@privaterelay.appleid.com` indirizzo. Un `invalid_client` errore significa quasi sempre che il JWT segreto del client è scaduto o che l'ID dei servizi o l'URL di ritorno non corrispondono esattamente. `/oauth2/idpresponse`

Apple-specific insidie

Il segreto del cliente è un JWT che scade

Il segreto del client Apple non è una stringa statica. È un ES256-signed JWT con una durata massima di sei mesi (15.777.000 secondi) e quando scade, l'accesso Apple si interrompe senza alcun avviso. `invalid_client` Nella configurazione descritta qui, fornisci ad Amazon Cognito la chiave privata (insieme a Services ID, Team ID e Key ID) e Amazon Cognito genera il JWT segreto del client da quella chiave, senza incollare un segreto predefinito. Mantieni la chiave privata registrata con Apple valida e aggiornata nella configurazione IdP; se revochi o sostituisci la chiave Apple, aggiorna l'IdP (ad esempio con) con `UpdateIdentityProvider` la nuova chiave.

Collegamento tra email e account con inoltro privato

Quando un utente sceglie Nascondi la mia email, Apple restituisce un `@privaterelay.appleid.com` indirizzo anziché quello reale. Questo indirizzo è diverso dall'indirizzo e-mail che la stessa persona utilizza con altri provider, quindi non collegare gli account tra i diversi provider di identità tramite e-mail quando si tratta di Apple, ma utilizza invece un link fornito dal provider. `sub` Per inviare posta a un indirizzo di inoltro, registra i domini di invio nella sezione Accedi con Apple per le comunicazioni via email.

tipo email_verified e account gestiti

L'`email_verified`affermazione di Apple può arrivare come stringa `"true"` o come `true` booleano: gestiscono entrambi. È sempre vero per gli account Apple consumer. Per gli account Apple at Work & School (gestiti) può essere falso. Preferisci il collegamento `sub` basato e normalizza il valore della stringa prima di qualsiasi logica di fusione.

Aggiungere Facebook come fornitore di identità social

> **Note**
>
> Ultima verifica sulla console del provider: 2 ottobre 2026. Facebook Steps utilizza la console Meta for Developers, che organizza la configurazione delle app in base ai casi d'uso. Le impostazioni di reindirizzamento OAuth si trovano in Casi d'uso, Personalizza, Impostazioni.

Per consentire ai tuoi utenti di accedere con Facebook, crei un'app Meta con il caso d'uso Facebook Login, quindi aggiungi Facebook come provider di identità (IdP) nel tuo pool di utenti. Devi prima abilitare l'accesso gestito.

Crea un'app Meta per Facebook Login

Per creare e configurare un'app Meta

1. Accedi a Meta for Developers, scegli Le mie app, quindi Crea app.
2. Per il caso d'uso, scegli Autentica e richiedi dati agli utenti con Facebook Login. Completa la creazione dell'app. Le nuove app iniziano in modalità Sviluppo.
3. Da Impostazioni dell'app, quindi di base, registra l'ID dell'app e il segreto dell'app. Imposta App Domains sull'host del dominio del tuo pool di utenti (`<your-prefix>.auth.<region>.amazoncognito.comoauth.example.com`) e aggiungi la piattaforma Website con l'URL di accesso gestito come URL del sito.
4. Vai a Casi d'uso, scegli il caso d'uso di Facebook Login, quindi Personalizza. Conferma l'`public_profile`autorizzazione (richiesta) e aggiungi`email`.
5. Nelle impostazioni dello use case, imposta gli URI di reindirizzamento OAuth validi sull'`/oauth2/idpresponse`endpoint del dominio del tuo pool di utenti:
   - Dominio Amazon Cognito predefinito: `https://<your-prefix>.auth.<region>.amazoncognito.com/oauth2/idpresponse`
   - Dominio personalizzato: `https://auth.example.com/oauth2/idpresponse`

Aggiungi Facebook al tuo pool di utenti

Per aggiungere Facebook come IdP nel Console di gestione AWS

1. Nella console Amazon Cognito, scegli il tuo pool di utenti, poi Social e provider esterni, quindi Aggiungi un provider di identità, quindi Facebook.
2. Inserisci l'ID dell'app e il segreto dell'app e scegli una versione dell'API di Facebook. Scegli l'ultima versione disponibile, poiché ogni versione ha una data di sospensione e gli ambiti possono variare tra le versioni.
3. Per gli ambiti autorizzati, inserisci, separati da `public_profile,email` virgole.
4. Mappa avanti `email` e `email` indietro. `name` `name`
5. Scegli Aggiungi provider di identità e abilita Facebook sul client dell'app.

Prova l'accesso a Facebook

Mentre l'app è in modalità Sviluppo, aggiungi il tuo account Facebook come ruolo dell'app (amministratore, sviluppatore o tester). Quindi apri la pagina di accesso gestito e scegli Continua con Facebook.

Checkpoint

Facebook torna `/oauth2/idpresponse` alla tua app con un codice di autorizzazione e un nuovo utente federato appare nel tuo pool di utenti `email` e `name` lo compila. Quindi passa l'app alla modalità Live e conferma che un account senza ruolo può accedere. Un account senza ruolo che visualizza «app non attiva» o «in modalità di sviluppo» significa che l'app è ancora in modalità Sviluppo o non dispone di accesso avanzato.

Facebook-specific insidie

modalità di sviluppo e revisione dell'app

Una nuova app è in modalità Sviluppo e solo gli utenti con ruolo di app (Amministratore, Sviluppatore, Tester) possono accedere. Per consentire l'accesso al pubblico generico, passa l'app alla modalità Live, completa la verifica aziendale e ottieni l'accesso avanzato per `email` (e un accesso migliorato per). `public_profile` First-time i tester spesso cliccano su «app non attiva» perché eseguono il test con un account diverso dal ruolo.

Nessun reclamo con email_verified

Facebook non restituisce alcun `email_verified` reclamo, quindi non c'è nulla da mappare e Amazon Cognito considera l'e-mail di Facebook come non verificata. Non dare per scontato che si tratti di un'e-mail verificata proveniente da Facebook e collega gli account tramite e-mail solo tramite una logica deliberata e affidabile anziché automaticamente.

Mantieni gli oscilloscopi piccoli

Sign-in il completamento scende bruscamente oltre le quattro autorizzazioni circa. Conserva la richiesta a e. `public_profile` `email` La federazione di Amazon Cognito non è inoltre compatibile con Facebook Limited Login su iOS.

Aggiungi Login with Amazon come provider di identità social

> **Note**
>
> Ultima verifica sulla console del provider: 2 ottobre 2026. Login with Amazon Steps utilizza la console Amazon Developer (non la AWS console).

Per consentire ai tuoi utenti di accedere con il loro account Amazon, devi registrare un profilo di sicurezza Login with Amazon, quindi aggiungere Login with Amazon come provider di identità (IdP) nel tuo pool di utenti. Devi prima abilitare l'accesso gestito.

Registra un profilo di sicurezza Login with Amazon

Per creare e configurare un profilo di sicurezza

1. Accedi alla console Amazon Developer o vai su App e servizi, quindi Accedi con Amazon.
2. Scegli Crea un nuovo profilo di sicurezza. Inserisci un nome (mostrato nella schermata di consenso), una descrizione e l'URL dell'informativa sulla privacy per il consenso. Selezionare Salva.
3. Rivela l'ID cliente e il segreto del cliente per il profilo di sicurezza. Li inserisci quando aggiungi Login with Amazon al tuo pool di utenti.
4. Apri le Impostazioni Web per il profilo e scegli Modifica. Inserisci i valori seguenti, quindi Salva.

   - Origini consentite (HTTPS, nessun percorso): `https://<your-prefix>.auth.<region>.amazoncognito.com` o`https://auth.example.com`.
   - URL di ritorno consentiti (l'`/oauth2/idpresponse`endpoint): `https://<your-prefix>.auth.<region>.amazoncognito.com/oauth2/idpresponse` o. `https://auth.example.com/oauth2/idpresponse` Amazon Cognito utilizza il flusso di reindirizzamento, quindi l'URL di ritorno è il valore da cui dipende.

Aggiungi Login with Amazon al tuo pool di utenti

Per aggiungere Login with Amazon come IdP nel Console di gestione AWS

1. Nella console Amazon Cognito, scegli il tuo pool di utenti, poi Social e provider esterni, quindi Aggiungi un provider di identità, quindi Login with Amazon.
2. Inserisci l'ID cliente e il segreto del cliente dal profilo di sicurezza.

3. Per gli ambiti autorizzati, inserisci`profile`. Ciò restituisce il nome, l'e-mail e l'ID utente dell'utente. (Aggiungi `postal_code` solo se hai bisogno del CAP o del codice postale).
4. Esegui `email` il `email` mapping `name` verso`name`, verso e `user_id` verso l'attributo del tuo nome utente.
5. Scegli Aggiungi provider di identità e abilita Login with Amazon sul client dell'app.

Prova il Login with Amazon

Apri la tua pagina di accesso gestito e scegli Continua con Amazon, quindi autentica e accetta la schermata di consenso.

Checkpoint

Il browser torna `/oauth2/idpresponse` all'app con un codice di autorizzazione e l'utente appare nel tuo pool di utenti come utente federato il cui nome utente è preceduto da un prefisso. `LoginWithAmazon_` Il token ID contiene i file `email` mappati e. `name` Nota che `email_verified` legge `false` a meno che non lo imposti con un trigger (vedi insidie).

Login with Amazon pitfalls

Nessun reclamo verificato tramite email

L'`profile`ambito restituisce solo `email``name`, e `user_id` non è possibile `email_verified` rivendicare la mappatura. Le e-mail mappate non sono verificate per impostazione predefinita, quindi gli utenti di Login with Amazon atterrano con `email_verified` set to. `false`

Effetto sul collegamento degli account

Poiché l'e-mail non è verificata, Amazon Cognito non unirà automaticamente un utente Login with Amazon a un utente esistente verificato tramite e-mail. Per collegare o contrassegnare l'e-mail come verificata, utilizza un trigger Lambda prima della registrazione (`AdminLinkProviderForUser`and) e solo quando il tuo modello di fiducia giustifica l'attendibilità dell'e-mail. Amazon-asserted

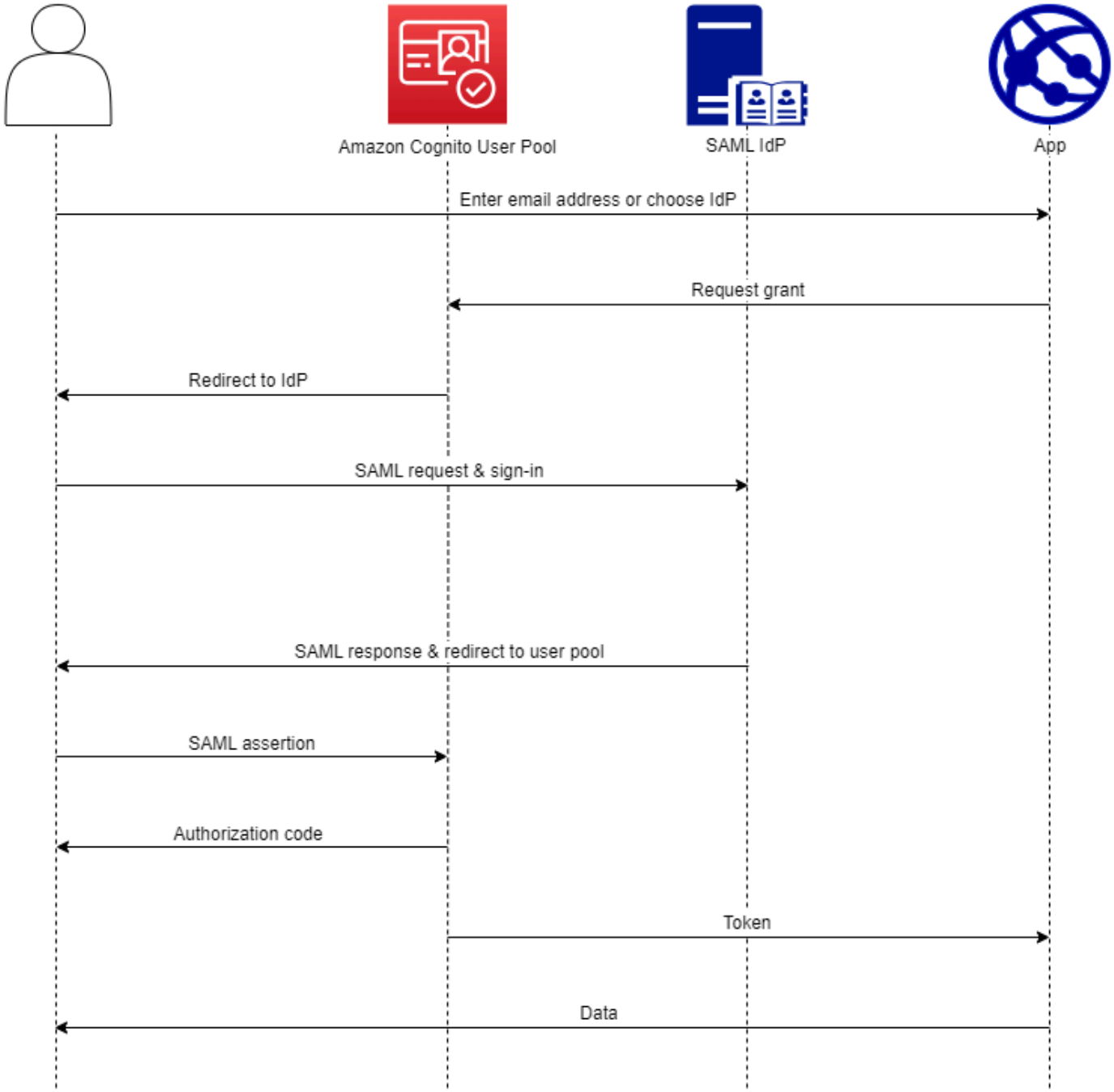
Gli attributi mappati devono essere scrivibili

Qualsiasi attributo personalizzato mappato deve essere modificabile e scrivibile dal client dell'app, altrimenti l'accesso non riesce o l'attributo viene ignorato automaticamente. Standard e vanno bene. `email` `name`

Utilizzo di provider di identità SAML con un pool di utenti

Puoi scegliere di fare in modo che gli utenti delle tue app web e mobili accedano tramite un provider di identità SAML (IdP) come Microsoft Active Directory Federation Services (ADFS) o Shibboleth. È necessario selezionare un provider di identità SAML che supporti lo standard SAML 2.0.

Con l'accesso gestito, Amazon Cognito autentica gli utenti IdP locali e di terze parti ed emette token web JSON (JWT). Con i token emessi da Amazon Cognito, puoi consolidare più fonti di identità in uno standard OpenID Connect (OIDC) universale per tutte le tue app. Amazon Cognito può elaborare le asserzioni SAML dei tuoi provider di terze parti in quello standard SSO. Puoi creare e gestire un IdP SAML nell'API Console di gestione AWS dei pool di utenti di Amazon Cognito, tramite o con AWS CLI l'API dei pool di utenti. Per creare il tuo primo IdP SAML in, Console di gestione AWS consulta. Aggiungere e gestire i provider di identità SAML in un pool di utenti



Note

La federazione con accesso tramite un IdP di terze parti è una funzionalità dei pool di utenti di Amazon Cognito. I pool di identità di Amazon Cognito, a volte chiamati identità federate di Amazon Cognito, sono un'implementazione della federazione che devi configurare

separatamente in ogni pool di identità. Un pool di utenti può essere un IdP di terze parti per un pool di identità. Per ulteriori informazioni, consulta Pool di identità di Amazon Cognito.

Riferimento rapido per la configurazione IdP

Devi configurare il tuo IdP SAML per accettare richieste e inviare risposte al tuo pool di utenti. La documentazione per il tuo IdP SAML contiene informazioni su come aggiungere il tuo pool di utenti come relying party o applicazione per il tuo IdP SAML 2.0. La documentazione che segue fornisce i valori da fornire per l'ID dell'entità SP e l'URL dell'assertion consumer service (ACS).

Riferimento rapido ai valori SAML del pool di utenti

ID dell'entità SP

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

URL ACS

```
https://Your user pool domain/saml2/idpresponse
```

È necessario configurare il pool di utenti per supportare il proprio provider di identità. I passaggi di alto livello per aggiungere un IdP SAML esterno sono i seguenti.

1. Scarica i metadati SAML dal tuo IdP o recupera l'URL del tuo endpoint di metadati. Consulta Configurazione del tuo provider di identità SAML di terze parti.
2. Aggiungi un nuovo IdP al tuo pool di utenti. Carica i metadati SAML o fornisci l'URL dei metadati. Consulta Aggiungere e gestire i provider di identità SAML in un pool di utenti.
3. Assegna l'IdP ai client della tua app. Consulta Application-specific impostazioni con i client dell'app.

Argomenti

- Cose da sapere su SAML IdPs nei pool di utenti di Amazon Cognito
- Distinzione tra maiuscole e minuscole dei nomi utente SAML
- Configurazione del tuo provider di identità SAML di terze parti
- Aggiungere e gestire i provider di identità SAML in un pool di utenti

- Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito
- Disconnessione degli utenti SAML con accesso singolo
- Firma e crittografia SAML
- Nomi e identificatori dei provider di identità SAML

Cose da sapere su SAML IdPs nei pool di utenti di Amazon Cognito

L'implementazione di un IdP SAML 2.0 comporta alcuni requisiti e restrizioni. Fai riferimento a questa sezione quando implementi il tuo IdP. Troverai anche informazioni utili per la risoluzione degli errori durante la federazione SAML con un pool di utenti.

Amazon Cognito elabora le asserzioni SAML per te

I pool di utenti di Amazon Cognito supportano la federazione SAML 2.0 con endpoint. POST-binding In questo modo, l'app non avrà bisogno di recuperare o analizzare le risposte dell'asserzione SAML poiché il bacino d'utenza riceve la risposta SAML direttamente dal provider di identità tramite un agente utente. Il bacino d'utenza si comporta da provider di servizi (SP) per conto della tua applicazione. Amazon Cognito supporta SP-initiated il IdP-initiated single sign-on (SSO) come descritto nelle sezioni 5.1.2 e 5.1.4 della panoramica tecnica SAML. V2.0

Fornisci un certificato di firma IdP valido

Il certificato di firma nei metadati del tuo provider SAML non deve essere scaduto quando configuri l'IdP SAML nel tuo pool di utenti.

I pool di utenti supportano più certificati di firma

Quando l'IdP SAML include più di un certificato di firma nei metadati SAML, al momento dell'accesso il pool di utenti determina che l'asserzione SAML è valida se corrisponde a qualsiasi certificato nei metadati SAML. Ogni certificato di firma non deve contenere più di 4.096 caratteri.

Mantenere il parametro dello stato del relè

Amazon Cognito e l'IdP SAML gestiscono le informazioni sulla sessione con un parametro `relayState`.

1. Amazon Cognito supporta i valori `relayState` superiori a 80 byte. Mentre le specifiche SAML affermano che il valore `relayState` "non deve superare 80 byte di lunghezza", la pratica attuale del settore spesso si discosta da questo comportamento. Di conseguenza, il rifiuto `relayState` di valori superiori a 80 byte interromperà molte integrazioni di provider SAML standard.

2. Il `relayState` token è un riferimento opaco alle informazioni sullo stato gestite da Amazon Cognito. Amazon Cognito non garantisce i contenuti del parametro `relayState`. Non analizza i contenuti in modo tale che l'app dipende dal risultato. Per ulteriori informazioni consulta le Specifiche SAML 2.0.

Identifica l'endpoint ACS

Il provider di identità SAML richiede che venga impostato un endpoint assertion consumer. L'IdP reindirizza gli utenti a questo endpoint con la relativa asserzione SAML. Configura il seguente endpoint nel dominio del pool di utenti per il binding SAML 2.0 POST nel gestore dell'identità digitale SAML.

```
https://Your user pool domain/saml2/idpresponse
With an Amazon Cognito domain:
https://mydomain.auth.us-east-1.amazoncognito.com/saml2/idpresponse
With a custom domain:
https://auth.example.com/saml2/idpresponse
```

Per ulteriori informazioni sui domini del pool di utenti, consulta l'articolo Configurazione di un dominio di bacino d'utenza.

Nessuna asserzione ripetuta

Non puoi ripetere né riprodurre un'asserzione SAML sul tuo endpoint `saml2/idpresponse` Amazon Cognito. Un'asserzione SAML riprodotta dispone di un ID asserzione che duplica l'ID di una risposta IdP precedente.

L'ID del pool di utenti è l'ID dell'entità SP

È necessario fornire all'IdP l'ID del pool di utenti nel service provider (SP)`urn`, chiamato anche URI del pubblico o ID dell'entità SP. Il formato dell'URI audience per il pool di utenti è il seguente.

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

Puoi trovare l'ID del tuo pool di utenti nella sezione Panoramica del pool di utenti nella console Amazon Cognito.

Mappa tutti gli attributi obbligatori

Configura l'IdP SAML per fornire i valori per tutti gli attributi impostati come obbligatori nel pool di utenti. Ad esempio, `email` è un attributo obbligatorio comune per i pool di utenti. Prima che

gli utenti possano effettuare l'accesso, le asserzioni IdP SAML devono includere un'attestazione mappata all'attributo pool di utenti `email`. Per ulteriori informazioni sulla mappatura attributi, consultareMappatura degli attributi IdP su profili e token.

Il formato di asserzione ha requisiti specifici

Il tuo IdP SAML deve includere le seguenti affermazioni nell'asserzione SAML.

- Un reclamo. `NameID` Amazon Cognito associa un'asserzione SAML all'utente di destinazione tramite. `NameID` In caso di `NameID` modifiche, Amazon Cognito considera l'affermazione come riferita a un nuovo utente. L'attributo impostato `NameID` nella configurazione IdP deve avere un valore persistente. Per assegnare agli utenti SAML un profilo utente coerente nel tuo pool di utenti, assegna la tua `NameID` dichiarazione in base a un attributo con un valore che non cambia.

  ```
  <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:persistent">
    carlos
  </saml2:NameID>
  ```

  Una `Format` `NameID` dichiarazione del tuo IdP `urn:oasis:names:tc:SAML:1.1:nameid-format:persistent` indica che il tuo IdP sta passando un valore invariato. Amazon Cognito non richiede questa dichiarazione di formato e assegna un formato se `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified` il tuo IdP non specifica un formato del claim. `NameID` Questo comportamento è conforme alla sezione 2.2.2 Complex Type NameIDType, della specifica SAML 2.0.
- Una richiesta `AudienceRestriction` con un valore `Audience` che imposta l'ID dell'entità SP del pool di utenti come destinazione della risposta.

  ```
  <saml:AudienceRestriction>
    <saml:Audience> urn:amazon:cognito:sp:us-east-1_EXAMPLE
  </saml:AudienceRestriction>
  ```
- Per il SP-initiated single sign-on, un `Response` elemento con un valore dell'ID della richiesta SAML originale`InResponseTo`.

  ```
  <saml2p:Response Destination="https://mydomain.auth.us-east-1.amazoncognito.com/
  saml2/idpresponse" ID="id123" InResponseTo="_dd0a3436-bc64-4679-
  a0c2-cb4454f04184" IssueInstant="Date-time stamp" Version="2.0"
   xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:xs="http://
  www.w3.org/2001/XMLSchema">
  ```

> **Note**
>
> IdP-initiated Le asserzioni SAML non devono contenere un valore. `InResponseTo`

- Un `SubjectConfirmationData` elemento con un `Recipient` valore dell'`saml2/idpresponse`endpoint del pool di utenti e, per SP-initiated SAML, un `InResponseTo` valore che corrisponde all'ID della richiesta SAML originale.

```
<saml2:SubjectConfirmationData InResponseTo="_dd0a3436-bc64-4679-a0c2-
cb4454f04184" NotOnOrAfter="Date-time stamp" Recipient="https://mydomain.auth.us-
east-1.amazoncognito.com/saml2/idpresponse"/>
```

SP-initiated richieste di accesso

Quando Endpoint Authorize reindirizza l'utente alla pagina di accesso dell'IdP, Amazon Cognito include una richiesta SAML in un parametro URL della richiesta `HTTP GET`. Una richiesta SAML contiene informazioni sul tuo pool di utenti, incluso l'endpoint ACS. Facoltativamente, puoi applicare una firma crittografica a queste richieste.

Firma le richieste e crittografa le risposte

Ogni pool di utenti con un provider SAML genera una coppia di key pair asimmetrica e un certificato di firma per una firma digitale che Amazon Cognito assegna alle richieste SAML. Ogni IdP SAML esterno che configuri per supportare la risposta SAML crittografata fa sì che Amazon Cognito generi una nuova coppia di chiavi e un nuovo certificato di crittografia per quel provider. Per visualizzare e scaricare i certificati con la chiave pubblica, scegli il tuo IdP nel menu Social e provider esterni della console Amazon Cognito.

Per stabilire un rapporto di fiducia con le richieste SAML provenienti dal tuo pool di utenti, fornisci al tuo IdP una copia del certificato di firma SAML 2.0 del tuo pool di utenti. Il tuo IdP potrebbe ignorare le richieste SAML firmate dal tuo pool di utenti se non configuri l'IdP per accettare le richieste firmate.

1. Amazon Cognito applica una firma digitale alle richieste SAML che l'utente trasmette al tuo IdP. Il tuo pool di utenti firma tutte le richieste di single logout (SLO) e puoi configurare il tuo pool di utenti per firmare le richieste Single Sign-On (SSO) per qualsiasi IdP esterno SAML. Quando fornisci una copia del certificato, il tuo IdP può verificare l'integrità delle richieste SAML degli utenti.
2. Il tuo IdP SAML può crittografare le risposte SAML con il certificato di crittografia. Quando configuri un IdP con crittografia SAML, l'IdP deve inviare solo risposte crittografate.

Codifica caratteri non alfanumerici

Amazon Cognito non accetta UTF-8 caratteri a 4 byte come o # che il # tuo IdP trasmette come valore di attributo. Puoi codificare il carattere in Base64, passarlo come testo e quindi decodificarlo nell'app.

Nell'esempio seguente, la dichiarazione relativa all'attributo non è accettata:

```
<saml2:Attribute Name="Name" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
  <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="xsd:string">#</saml2:AttributeValue>
</saml2:Attribute>
```

A differenza dell'esempio precedente, è accettata la seguente dichiarazione di attributo:

```
<saml2:Attribute Name="Name" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
  <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="xsd:string">8J+YkA==</saml2:AttributeValue>
</saml2:Attribute>
```

L'endpoint di metadati deve disporre di una protezione valida a livello di trasporto

Se `InvalidParameterException` durante la creazione di un IdP SAML con un URL dell'endpoint di metadati HTTPS vedi, ad esempio, «Errore nel recupero dei metadati *<metadata endpoint>* da», assicurati che l'endpoint di metadati abbia SSL impostato correttamente e che sia associato un certificato SSL valido. Per ulteriori informazioni sulla convalida dei certificati, consulta Che cos'è un certificato? SSL/TLS .

L'endpoint dei metadati deve essere su una porta TCP standard per HTTP o HTTPS

Amazon Cognito accetta solo URL di metadati per i provider SAML sulle porte TCP standard 80 per HTTP e 443 per HTTPS. Come best practice di sicurezza, ospita i metadati SAML su un URL con il prefisso. TLS-encrypted `https://` Inserisci gli URL dei metadati nel formato o. *`http://www.example.com/saml2/metadata.xml`* *`https://www.example.com/saml2/metadata.xml`* La console Amazon Cognito accetta URL di metadati solo con il prefisso. `https://` Puoi anche configurare i metadati IdP con e. CreateIdentityProviderUpdateIdentityProvider

I client di app con IdP-initiated SAML possono accedere solo con SAML

Quando attivi il supporto per un IdP SAML 2.0 che IdP-initiated supporta l'accesso a un app client, puoi aggiungere solo altro SAML IdPs 2.0 a quell'app client. Ti viene impedito di aggiungere la directory utente nel pool di utenti e tutti i provider di identità esterni non SAML a un client di app configurato in questo modo.

Le risposte di disconnessione devono utilizzare l'associazione POST

L'`/saml2/logout`endpoint accetta richieste `LogoutResponse` come`HTTP POST`. I pool di utenti non accettano risposte di disconnessione con `HTTP GET` associazione.

Rotazione dei certificati di firma dei metadati

Amazon Cognito memorizza nella cache i metadati SAML per un massimo di sei ore quando fornisci metadati con un URL. Quando esegui una rotazione dei certificati di firma dei metadati, configura la fonte di metadati per pubblicare sia i certificati originali che quelli nuovi per almeno sei ore. Quando Amazon Cognito aggiorna la cache dall'URL dei metadati, considera ogni certificato come valido e il tuo IdP SAML può iniziare a firmare le asserzioni SAML con il nuovo certificato. Trascorso questo periodo, puoi rimuovere il certificato originale dai metadati pubblicati.

Distinzione tra maiuscole e minuscole dei nomi utente SAML

Quando un utente federato tenta di accedere, il provider di identità SAML (IdP) trasmette un messaggio univoco ad Amazon `NameId` Cognito nell'asserzione SAML dell'utente. Amazon Cognito identifica un SAML-federated utente in base alla sua dichiarazione. `NameId` Indipendentemente dalle impostazioni di distinzione tra maiuscole e minuscole del tuo pool di utenti, Amazon Cognito riconosce un utente federato di ritorno da un IdP SAML quando trasmette la sua dichiarazione unica e con distinzione tra maiuscole e minuscole. `NameId` Se mappi un attributo come `email` a `NameId` e l'utente modifica il proprio indirizzo e-mail, non può accedere alla tua app.

Mappa `NameId` nelle asserzioni SAML da un attributo del provider di identità con valori che non cambiano.

Ad esempio, Carlos ha un profilo utente nel bacino d'utenza senza distinzione tra maiuscole e minuscole da un'asserzione SAML di Active Directory Federation Services (ADFS) che ha passato un valore `NameId` di `Carlos@example.com`. La prossima volta che Carlos tenta di accedere, il provider di identità ADFS passa un valore `NameId` di `carlos@example.com`. Poiché `NameId` deve rispettare esattamente la distinzione tra maiuscole e minuscole, l'accesso non ha esito positivo.

Se i tuoi utenti non riescono ad accedere dopo le `NameID` modifiche, elimina i loro profili utente dal tuo pool di utenti. Amazon Cognito crea nuovi profili utente al successivo accesso.

Argomenti

- Configurazione del tuo provider di identità SAML di terze parti
- Aggiungere e gestire i provider di identità SAML in un pool di utenti
- Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito
- Disconnessione degli utenti SAML con accesso singolo
- Firma e crittografia SAML
- Nomi e identificatori dei provider di identità SAML

Configurazione del tuo provider di identità SAML di terze parti

Quando desideri aggiungere un provider di identità SAML (IdP) al tuo pool di utenti, devi apportare alcuni aggiornamenti di configurazione nell'interfaccia di gestione del tuo IdP. Questa sezione descrive come formattare i valori che devi fornire al tuo IdP. Puoi anche scoprire come recuperare il documento di metadati statico o Active-URL che identifica l'IdP e le relative attestazioni SAML nel tuo pool di utenti.

Per configurare soluzioni di provider di identità (IdP) SAML 2.0 di terze parti in modo che funzionino con la federazione per i pool di utenti di Amazon Cognito, devi configurare il tuo IdP SAML per reindirizzare al seguente URL di Assertion Consumer Service (ACS):. `https://`*`mydomain.auth.us-east-1.amazoncognito.com`*`/saml2/idpresponse` Se il tuo pool di utenti ha un dominio Amazon Cognito, puoi trovare il percorso del dominio del pool di utenti nel menu Dominio del tuo pool di utenti nella console Amazon Cognito.

Alcuni SAML IdPs richiedono che tu fornisca il`urn`, chiamato anche URI del pubblico o ID dell'entità SP, nel modulo. `urn:amazon:cognito:sp:`*`us-east-1_EXAMPLE`* Puoi trovare l'ID del tuo pool di utenti nella sezione Panoramica del pool di utenti nella console Amazon Cognito.

Devi anche configurare il tuo IdP SAML per fornire valori per tutti gli attributi che hai designato come attributi obbligatori nel tuo pool di utenti. In genere, `email` è un attributo obbligatorio per i pool di utenti, nel qual caso l'IdP SAML deve fornire `email` una qualche forma di attestazione nella propria asserzione SAML e tu devi mappare l'attestazione all'attributo di quel provider.

Le seguenti informazioni di configurazione per le soluzioni IdP SAML 2.0 di terze parti sono un buon punto di partenza per configurare la federazione con i pool di utenti di Amazon Cognito. Per le informazioni più aggiornate, consulta direttamente la documentazione del tuo provider.

Per firmare le richieste SAML, devi configurare il tuo IdP in modo che consideri attendibili le richieste firmate dal certificato di firma del tuo pool di utenti. Per accettare risposte SAML crittografate, devi configurare il tuo IdP per crittografare tutte le risposte SAML nel tuo pool di utenti. Il tuo provider dispone della documentazione sulla configurazione di queste funzionalità. Per un esempio di Microsoft, vedi Configurare la crittografia del token SAML di Microsoft Entra.

> **Note**
>
> Amazon Cognito richiede solo il documento di metadati del provider di identità. Il tuo provider potrebbe anche offrire informazioni di configurazione personalizzate per la federazione SAML 2.0 con IAM o. Centro identità AWS IAM Per scoprire come configurare l'integrazione con Amazon Cognito, consulta le istruzioni generali per recuperare il documento di metadati e gestire il resto della configurazione nel tuo pool di utenti.

Soluzione	Ulteriori informazioni
ID Microsoft Entra	Metadati federativi
Okta	Come scaricare i metadati IdP e i certificati di firma SAML per l'integrazione di un'app SAML
Auth0	Configura Auth0 come provider di identità SAML
Ping Identity () PingFederate	Esportazione di metadati SAML da PingFederate
JumpCloud	Note di configurazione SAML
SecureAuth	Integrazione delle applicazioni SAML

Aggiungere e gestire i provider di identità SAML in un pool di utenti

Dopo aver configurato il tuo provider di identità per l'utilizzo con Amazon Cognito, puoi aggiungerlo ai tuoi pool di utenti e ai client delle app. Le seguenti procedure mostrano come creare, modificare ed eliminare i provider SAML in un pool di utenti Amazon Cognito.

Console di gestione AWS

Puoi utilizzare il Console di gestione AWS per creare ed eliminare i provider di identità SAML (). IdPs

Prima di creare un IdP SAML, devi disporre del documento di metadati SAML che ottieni dall'IdP di terze parti. Per istruzioni su come ricevere o generare il documento di metadati SAML richiesto, consulta Configurazione del tuo provider di identità SAML di terze parti.

Per configurare un provider di identità SAML 2.0 nel bacino d'utenza

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS .
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Social e provider esterni, quindi seleziona Aggiungi un provider di identità.
5. Scegli un provider di identità SAML.
6. Inserisci il nome del provider. È possibile passare questo nome descrittivo in un parametro di `identity_provider` richiesta aEndpoint Authorize.
7. Inserisci gli Identifiers (Identificatori) separati da virgole. Un identificatore indica ad Amazon Cognito che dovrebbe controllare l'indirizzo e-mail inserito dall'utente al momento dell'accesso e quindi indirizzarlo al provider che corrisponde al proprio dominio.
8. Scegli Add sign-out flow (Aggiungi flusso di disconnessione) se desideri che Amazon Cognito invii richieste di disconnessione firmate al tuo provider quando un utente si disconnette. Devi configurare il tuo IdP SAML 2.0 per inviare risposte di disconnessione all'endpoint creato quando `https://`*`mydomain.auth.us-east-1.amazoncognito.com`*`/saml2/logout` configuri l'accesso gestito. L'endpoint `saml2/logout` utilizza POST vincolanti.

   > **Note**
   >
   > Se questa opzione è selezionata e il tuo IdP SAML prevede una richiesta di disconnessione firmata, devi anche fornire al tuo IdP SAML il certificato di firma dal tuo pool di utenti.
   > L'IdP SAML elabora la richiesta di disconnessione firmata e disconnette l'utente dalla sessione di Amazon Cognito.

9. Scegli la tua IdP-initiated configurazione di accesso SAML. Come best practice di sicurezza, scegli Accetta solo le asserzioni SP-initiated SAML. Se hai preparato il tuo ambiente per

accettare in modo sicuro sessioni di accesso SAML non richieste, scegli Accetta e asserzioni SAML. SP-initiated IdP-initiated Per ulteriori informazioni, consulta Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito.

10. Scegli una Metadata document source (Fonte del documento di metadati). Se il provider di identità offre metadati SAML a un URL pubblico, puoi scegliere l'opzione Metadata document URL (URL del documento di metadati) e inserire l'URL pubblico. In caso contrario, seleziona Upload metadata document (Carica documento di metadati) e seleziona un file di metadati scaricato dal tuo provider in precedenza.

    > **Note**
    >
    > Ti consigliamo di inserire l'URL di un documento di metadati se il tuo provider ha un endpoint pubblico anziché caricare un file. Amazon Cognito aggiorna automaticamente i metadati dall'URL dei metadati. In genere, l'aggiornamento dei metadati avviene ogni 6 ore oppure prima della scadenza dei metadati, in base a ciò che avviene prima.

11. Mappa gli attributi tra il tuo provider SAML e il tuo pool di utenti per mappare gli attributi del provider SAML al profilo utente nel tuo pool di utenti. Includi gli attributi richiesti del bacino d'utenza nella mappa degli attributi.

    Ad esempio, quando scegli User pool attribute (Attributo del bacino d'utenza) `email`, inserisci il nome dell'attributo SAML così come compare nell'asserzione SAML del provider di identità. Il provider di identità può offrire asserzioni SAML di esempio che possono essere di aiuto nell'individuare il nome. Alcuni IdPs usano nomi semplici, come`email`, mentre altri usano nomi come i seguenti.

    ```
    http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
    ```

12. Scegli Crea.

API/CLI

Utilizza i comandi seguenti per creare e gestire un provider di identità SAML.

Creazione di un provider di identità e caricamento di un documento di metadati

- AWS CLI: `aws cognito-idp create-identity-provider`

Esempio con file di metadati: `aws cognito-idp create-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1 --provider-type SAML --provider-details file:///details.json --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

Dove `details.json` contiene:

```
"ProviderDetails": {
      "MetadataFile": "<SAML metadata XML>",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

> **Note**
>
> Se *<SAML metadata XML>* contiene delle istanze del personaggio", devi aggiungere \ come carattere di escape:\".

Esempio con URL di metadati: `aws cognito-idp create-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1 --provider-type SAML --provider-details MetadataURL=https://myidp.example.com/sso/saml/metadata --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

- AWS API: CreateIdentityProvider

Caricamento di un nuovo documento di metadati per un provider di identità

- AWS CLI: `aws cognito-idp update-identity-provider`

  Esempio con file di metadati: `aws cognito-idp update-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1 --provider-details file:///details.json --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

Dove `details.json` contiene:

```
"ProviderDetails": {
      "MetadataFile": "<SAML metadata XML>",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

> **Note**
>
> Se *<SAML metadata XML>* contiene delle istanze del personaggio", devi aggiungere \ come carattere di escape:\".

Esempio con URL di metadati: `aws cognito-idp update-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1 --provider-details MetadataURL=https://myidp.example.com/sso/saml/metadata --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

- AWS API: UpdateIdentityProvider

Recupero delle informazioni su un provider di identità specifico

- AWS CLI: `aws cognito-idp describe-identity-provider`

  `aws cognito-idp describe-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1`

- AWS API: DescribeIdentityProvider

Per elencare informazioni su tutti IdPs

- AWS CLI: `aws cognito-idp list-identity-providers`

  Ad esempio: `aws cognito-idp list-identity-providers --user-pool-id us-east-1_EXAMPLE --max-results 3`

- AWS API: ListIdentityProviders

Eliminazione di un IdP

- AWS CLI: `aws cognito-idp delete-identity-provider`

```
aws cognito-idp delete-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1
```

- AWS API: DeleteIdentityProvider

Per configurare l'IdP SAML per l'aggiunta di un bacino d'utenza come relying party

- L'URN del provider di servizi del bacino d'utenza è: `urn:amazon:cognito:sp:us-east-1_EXAMPLE`. Amazon Cognito richiede un valore di restrizione del pubblico che corrisponda a questo URN nella risposta SAML. Configura il tuo IdP per utilizzare il seguente endpoint di binding POST per il IdP-to-SP messaggio di risposta.

```
https://mydomain.auth.us-east-1.amazoncognito.com/saml2/idpresponse
```

- Il tuo IdP SAML deve `NameID` compilare tutti gli attributi obbligatori per il tuo pool di utenti nell'asserzione SAML. `NameID`viene utilizzato per identificare in modo univoco l'utente federato SAML nel pool di utenti. Il tuo IdP deve trasmettere l'ID del nome SAML di ogni utente in un formato coerente con distinzione tra maiuscole e minuscole. Qualsiasi variazione del valore dell'ID del nome utente crea un nuovo profilo utente.

Fornire un certificato di firma all'IDP SAML 2.0

- Per scaricare una copia della chiave pubblica da Amazon Cognito che il tuo IdP può utilizzare per convalidare le richieste di disconnessione SAML, scegli il menu Social e provider esterni del tuo pool di utenti, seleziona il tuo IdP e, in Visualizza certificato di firma, seleziona Scarica come .crt.

Puoi eliminare qualsiasi provider SAML configurato nel tuo bacino d'utenza con la console Amazon Cognito.

Come eliminare un provider SAML

1. Accedi alla console Amazon Cognito.

2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegli il menu Social e provider esterni.
4. Seleziona il pulsante di opzione accanto al SAML IdPs che desideri eliminare.
5. Quando viene richiesto di confermare l'opzione Delete identity provider (Elimina provider di identità), inserisci il nome del provider SAML per confermare l'eliminazione, quindi scegli Delete (Elimina).

Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito

Amazon Cognito supporta Single Sign-On (SSO) e SSO (Service Provider-InitiatedSP-initiated). IdP-initiated Come migliore pratica di sicurezza, implementa l'SSO nel tuo pool di utenti. SP-initiated La sezione 5.1.2 della panoramica V2.0 tecnica SAML descrive l'SSO. SP-initiated Amazon Cognito è il provider di identità della tua app. L'app è il provider di servizi che recupera i token per gli utenti autenticati. Tuttavia, quando utilizzi un provider di identità di terze parti per autenticare gli utenti, Amazon Cognito diventa il provider di servizi. Quando gli utenti SAML 2.0 si autenticano con un SP-initiated flusso, devono sempre prima effettuare una richiesta ad Amazon Cognito e reindirizzare all'IdP per l'autenticazione.

Per alcuni casi d'uso aziendali, l'accesso alle applicazioni interne inizia da un segnalibro in un dashboard ospitato dal provider di identità aziendale. Quando un utente seleziona un segnalibro, il provider di identità genera una risposta SAML e la invia al provider di servizi per autenticare l'utente con l'applicazione.

Puoi configurare un IdP SAML nel tuo pool di utenti per IdP-initiated supportare l'SSO. Quando supporti IdP-initiated l'autenticazione, Amazon Cognito non può verificare di aver richiesto la risposta SAML che riceve perché Amazon Cognito non avvia l'autenticazione con una richiesta SAML. In SP-initiated SSO, Amazon Cognito imposta parametri di stato che convalidano una risposta SAML rispetto alla richiesta originale. Con SP-initiated l'accesso puoi anche proteggerti dalla falsificazione delle richieste tra siti (CSRF).

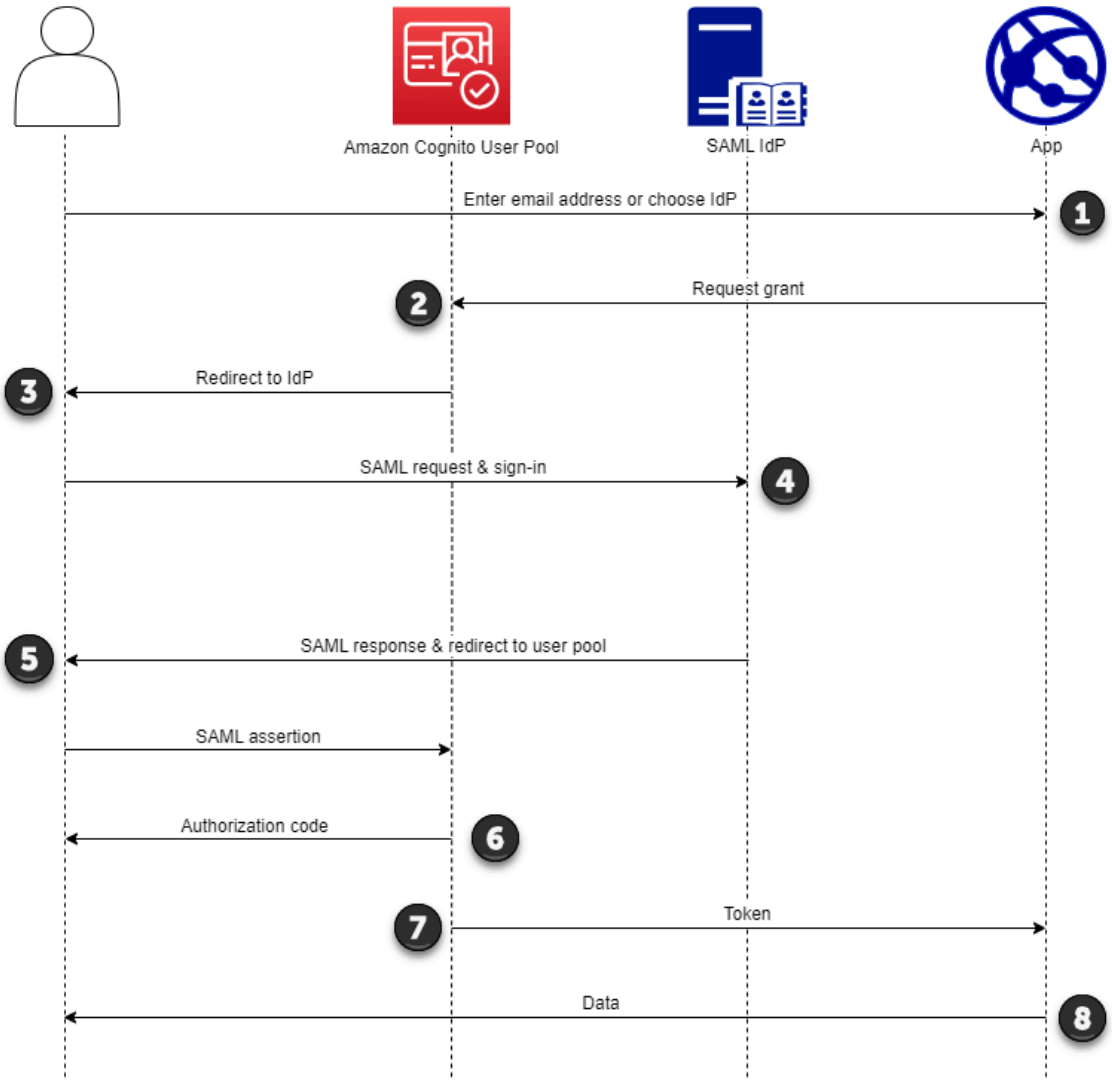
Argomenti

- Implementa l' SP-initated accesso SAML
- Implementa l' IdP-initiated accesso SAML

Implementa l' SP-initated accesso SAML

Come best practice, implementa l'accesso avviato dal fornitore di servizi () SP-initiated nel tuo pool di utenti. Amazon Cognito avvia la sessione dell'utente e lo reindirizza al tuo IdP. Con questo metodo, hai il massimo controllo su chi presenta le richieste di accesso. Puoi anche consentire IdP-initiated l'accesso a determinate condizioni.

La procedura seguente mostra come gli utenti completano l' SP-initiated accesso al tuo pool di utenti tramite un provider SAML.



1. L'utente inserisce il proprio indirizzo e-mail in una pagina di accesso. Per determinare il reindirizzamento dell'utente al suo IdP, puoi raccogliere il suo indirizzo e-mail in un'applicazione personalizzata o richiamare l'accesso gestito nella visualizzazione web.

Puoi configurare le tue pagine di accesso gestite per visualizzare un elenco IdPs o richiedere un indirizzo e-mail e abbinarlo all'identificatore del tuo IdP SAML. Per richiedere un indirizzo e-mail, modifica lo stile di branding dell'accesso gestito e, in Foundation, individua il comportamento di autenticazione e, in Provider display, imposta Stile di visualizzazione su Domain search input.

2. L'app richiama l'endpoint di reindirizzamento del pool di utenti e richiede una sessione con l'ID client che corrisponde all'app e l'ID IdP che corrisponde all'utente.
3. Amazon Cognito reindirizza l'utente all'IdP con una richiesta SAML, facoltativamente firmata, in un elemento. `AuthnRequest`
4. L'IdP autentica l'utente in modo interattivo o con una sessione memorizzata in un cookie del browser.
5. L'IdP reindirizza l'utente all'endpoint di risposta SAML del pool di utenti con l'asserzione SAML crittografata facoltativamente nel payload POST.

> **Note**
>
> Amazon Cognito annulla le sessioni che non ricevono una risposta entro 5 minuti e reindirizza l'utente all'accesso gestito. Quando il tuo utente riscontra questo risultato, riceve un messaggio di errore. `Something went wrong`

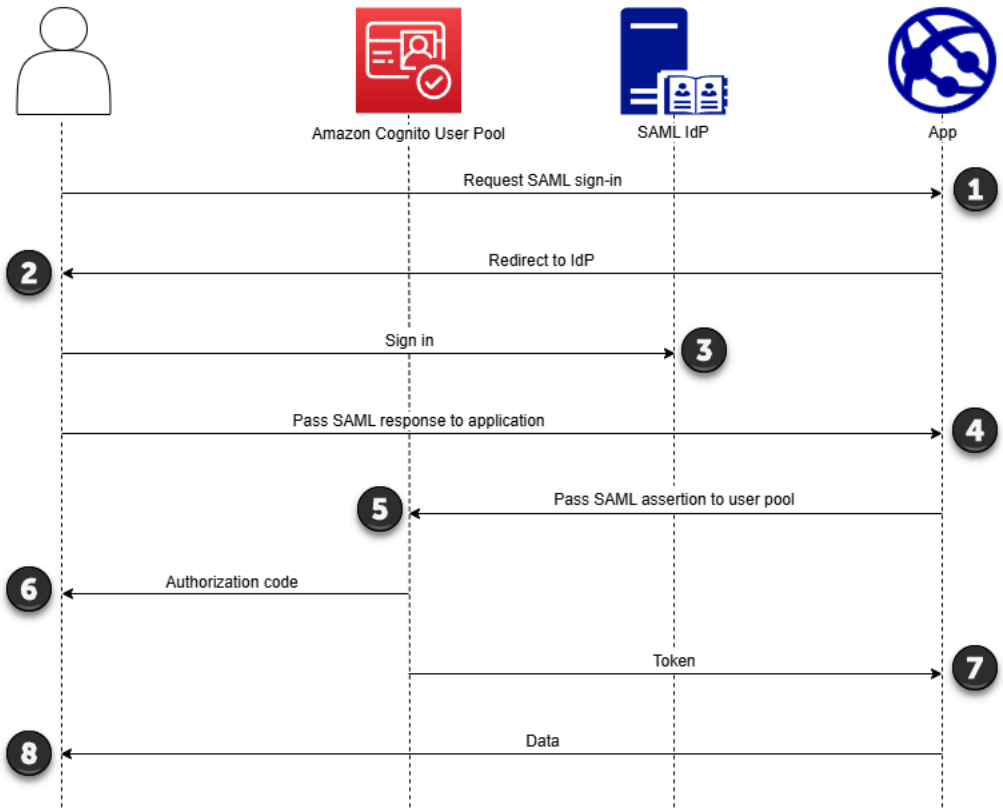
6. Dopo aver verificato l'asserzione SAML e aver mappato gli attributi utente dalle affermazioni nella risposta, Amazon Cognito crea o aggiorna internamente il profilo dell'utente nel pool di utenti. In genere, il tuo pool di utenti restituisce un codice di autorizzazione alla sessione del browser dell'utente.
7. L'utente presenta il codice di autorizzazione all'app, che lo scambia con token web JSON (JWT).
8. L'app accetta ed elabora il token ID dell'utente come autenticazione, genera richieste autorizzate alle risorse con il relativo token di accesso e archivia il relativo token di aggiornamento.

Quando un utente si autentica e riceve una concessione di codice di autorizzazione, il pool di utenti restituisce ID, accesso e token di aggiornamento. Il token ID è un oggetto di autenticazione per la gestione delle identità. OIDC-based Il token di accesso è un oggetto di autorizzazione con ambiti OAuth 2.0. Il token di aggiornamento è un oggetto che genera nuovi ID e token di accesso quando i token correnti dell'utente sono scaduti. Puoi configurare la durata dei token degli utenti nel client dell'app del pool di utenti.

Puoi anche scegliere la durata dei token di aggiornamento. Dopo la scadenza del token di aggiornamento, l'utente deve effettuare nuovamente l'accesso. Se si sono autenticati tramite un IdP SAML, la durata della sessione degli utenti è impostata dalla scadenza dei loro token, non dalla scadenza della sessione con il loro IdP. L'app deve archiviare il token di aggiornamento di ogni utente e rinnovare la sessione alla scadenza. L'accesso gestito mantiene le sessioni utente in un cookie del browser valido per 1 ora.

Implementa l' IdP-initiated accesso SAML

Quando configuri il tuo provider di identità per l'accesso a IdP-initiated SAML 2.0, puoi presentare asserzioni SAML all'`saml2/idpresponse`endpoint nel dominio del tuo pool di utenti senza la necessità di avviare la sessione presso. Endpoint Authorize Un pool di utenti con questa configurazione accetta asserzioni IdP-initiated SAML da un provider di identità esterno del pool di utenti supportato dal client dell'app richiesto.



1. Un utente richiede l'accesso SAML con la tua applicazione.
2. L'applicazione richiama un browser o reindirizza l'utente alla pagina di accesso del suo provider SAML.

3. L'IdP autentica l'utente in modo interattivo o con una sessione memorizzata in un cookie del browser.
4. L'IdP reindirizza l'utente all'applicazione con l'asserzione, o risposta, SAML nel corpo POST.
5. L'applicazione aggiunge l'asserzione SAML al corpo POST di una richiesta all'endpoint del pool di utenti. `saml2/idpresponse`
6. Amazon Cognito emette un codice di autorizzazione per l'utente.
7. L'utente presenta il codice di autorizzazione all'app, che lo scambia con token web JSON (JWT).
8. L'applicazione accetta ed elabora il token ID dell'utente come autenticazione, genera richieste autorizzate alle risorse con il relativo token di accesso e archivia il relativo token di aggiornamento.

I passaggi seguenti descrivono il processo generale di configurazione e accesso con un provider IdP-initiated SAML 2.0.

1. Crea o designa un pool di utenti e un client per l'app.
2. Crea un IdP SAML 2.0 nel tuo pool di utenti.
3. Configura il tuo IdP per supportare l'avvio dell'IdP. IdP-initiated SAML introduce considerazioni sulla sicurezza a cui gli altri provider SSO non sono soggetti. Per questo motivo, non puoi aggiungere elementi non SAML IdPs, incluso il pool di utenti stesso, a nessun client di app che utilizza un provider SAML con accesso. IdP-initiated
4. Associa il tuo provider IdP-initiated SAML a un client di app nel tuo pool di utenti.
5. Indirizza l'utente alla pagina di accesso del tuo IdP SAML e recupera un'asserzione SAML.
6. Indirizza il tuo utente all'endpoint del tuo pool di utenti con la sua asserzione SAML`saml2/idpresponse`.
7. Ricevi token web JSON (JWT).

Per accettare asserzioni SAML non richieste nel tuo pool di utenti, devi considerarne l'effetto sulla sicurezza dell'app. Quando si accettano le richieste, è probabile che si verifichino tentativi di contraffazione delle richieste e tentativi di CSRF. IdP-initiated Sebbene il tuo pool di utenti non sia in grado di verificare una sessione di IdP-initiated accesso, Amazon Cognito convalida i parametri di richiesta e le asserzioni SAML.

Inoltre, l'asserzione SAML non deve contenere un `InResponseTo` reclamo e deve essere stata emessa nei 6 minuti precedenti.

Devi inviare richieste con IdP-initiated SAML al tuo. `/saml2/idpresponse` Per gestire SP-initiated le richieste di autorizzazione all'accesso, devi fornire parametri che identifichino il client dell'app richiesta, gli ambiti, l'URI di reindirizzamento e altri dettagli come parametri della stringa di query nelle richieste. `HTTP GET` Per le asserzioni IdP-initiated SAML, tuttavia, i dettagli della richiesta devono essere formattati come `RelayState` parametri nel corpo di una richiesta. `HTTP POST` Il corpo della richiesta deve contenere anche l'asserzione SAML come parametro. `SAMLResponse`

Di seguito è riportato un esempio di richiesta e risposta per un provider IdP-initiated SAML.

```
POST /saml2/idpresponse HTTP/1.1
User-Agent: USER_AGENT
Accept: */*
Host: example.auth.us-east-1.amazoncognito.com
Content-Type: application/x-www-form-urlencoded

SAMLResponse=[Base64-encoded SAML assertion]&RelayState=identity_provider
%3DMySAMLIdP%26client_id%3D1example23456789%26redirect_uri%3Dhttps%3A%2F
%2Fwww.example.com%26response_type%3Dcode%26scope%3Demail%2Bopenid%2Bphone

HTTP/1.1 302 Found
Date: Wed, 06 Dec 2023 00:15:29 GMT
Content-Length: 0
x-amz-cognito-request-id: 8aba6eb5-fb54-4bc6-9368-c3878434f0fb
Location: https://www.example.com?code=[Authorization code]
```

Console di gestione AWS

Per configurare un IdP per SAML IdP-initiated

1. Crea un pool di utenti, un client di app e un provider di identità SAML.
2. Dissocia tutti i provider di identità social e OIDC dal client dell'app, se ne sono associati.
3. Vai al menu Provider social ed esterni del tuo pool di utenti.
4. Modifica o aggiungi un provider SAML.
5. In Accesso IdP-initiated SAML, scegli Accetta SP-initiated e IdP-initiated asserzioni SAML.
6. Scegli Save changes (Salva modifiche).

API/CLI

Per configurare un IdP per SAML IdP-initiated

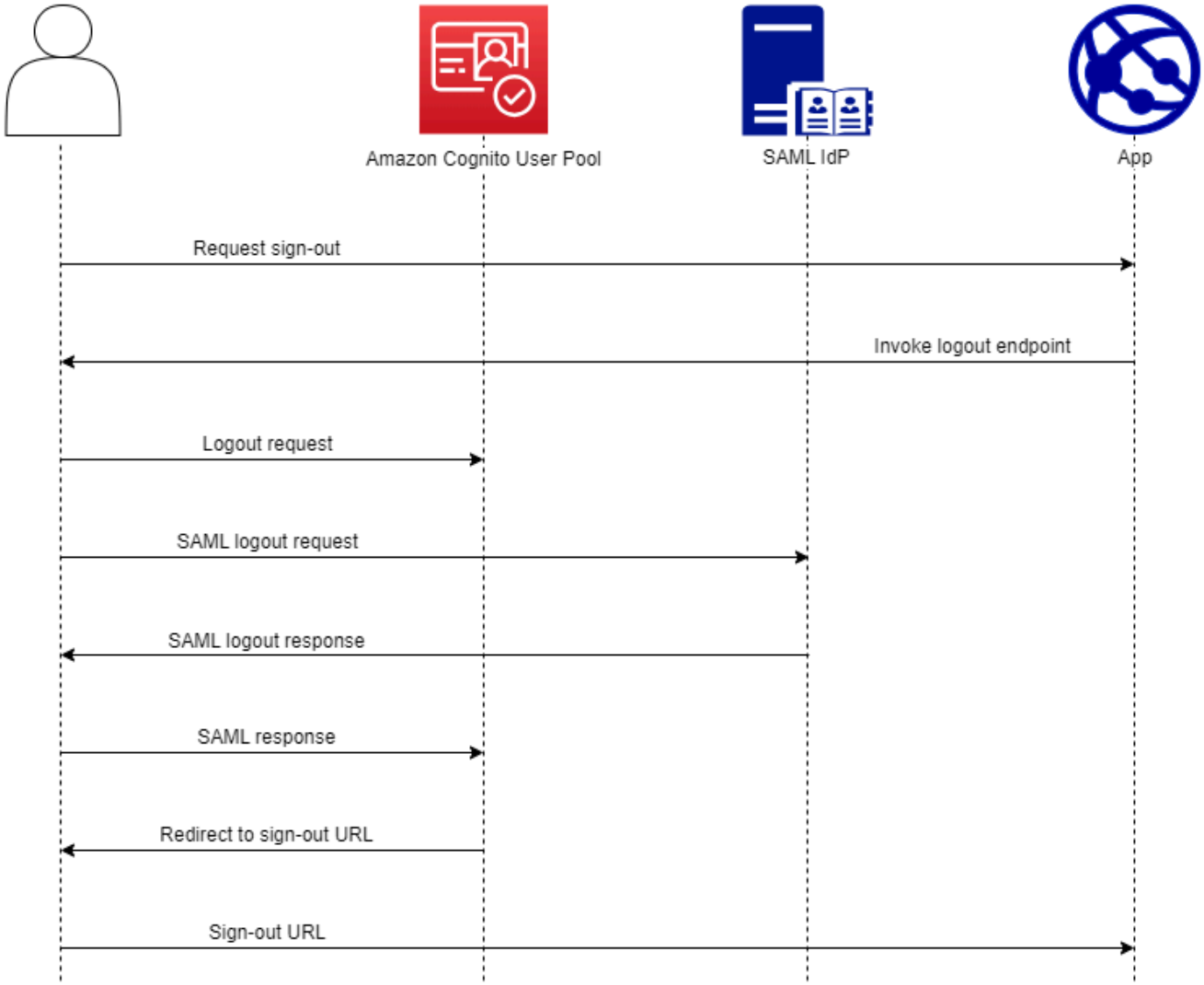
Configura IdP-initiated SAML con il `IDPInit` parametro in una richiesta CreateIdentityProvidero UpdateIdentityProviderAPI. Di seguito è riportato un esempio `ProviderDetails` di IdP che supporta IdP-initiated SAML.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Disconnessione degli utenti SAML con accesso singolo

Amazon Cognito supporta il single logout (SLO) SAML 2.0. Con SLO, la tua applicazione può disconnettere gli utenti dai loro provider di identità SAML (IdPs) quando si disconnettono dal tuo pool di utenti. In questo modo, quando gli utenti vogliono accedere nuovamente alla tua applicazione, devono autenticarsi con il proprio IdP SAML. In caso contrario, potrebbero disporre di cookie del browser IdP o del pool di utenti che li trasmettono all'applicazione senza la necessità di fornire credenziali.

Quando configuri il tuo IdP SAML per Sign-out supportare il flusso, Amazon Cognito reindirizza l'utente con una richiesta di disconnessione SAML firmata al tuo IdP. Amazon Cognito determina la posizione di reindirizzamento dall'`SingleLogoutService`URL nei metadati del tuo IdP. Amazon Cognito firma la richiesta di disconnessione con il certificato di firma del tuo pool di utenti.



Quando indirizzi un utente con una sessione SAML all'`/logout`endpoint del tuo pool di utenti, Amazon Cognito reindirizza l'utente SAML con la seguente richiesta all'endpoint SLO specificato nei metadati IdP.

```
https://[SingleLogoutService endpoint]?
SAMLRequest=[encoded SAML request]&
RelayState=[RelayState]&
SigAlg=http://www.w3.org/2001/04/xmldsig-more#rsa-sha256&
Signature=[User pool RSA signature]
```

L'utente torna quindi al tuo `saml2/logout` endpoint con un messaggio del suo `LogoutResponse` IdP. Il tuo IdP deve inviare una `HTTP POST` richiesta. `LogoutResponse` Amazon Cognito li reindirizza quindi alla destinazione di reindirizzamento indicata nella richiesta di disconnessione iniziale.

Il tuo provider SAML potrebbe inviare un messaggio contenente più `LogoutResponse` di uno. `AuthnStatement` Il `sessionIndex` valore inserito per primo `AuthnStatement` in una risposta di questo tipo deve corrispondere a quello contenuto `sessionIndex` nella risposta SAML che originariamente ha autenticato l'utente. Se si `sessionIndex` trova in un'altra`AuthnStatement`, Amazon Cognito non riconoscerà la sessione e l'utente non verrà disconnesso.

Console di gestione AWS

Per configurare la disconnessione SAML

1. Crea un pool di utenti, un client di app e un IdP SAML.
2. Quando crei o modifichi il tuo provider di identità SAML, in Informazioni sul provider di identità, seleziona la casella con il titolo Aggiungi flusso di disconnessione.
3. Dal menu Social e provider esterni del tuo pool di utenti, scegli il tuo IdP e individua il certificato di firma.
4. Scegli Scarica come .crt.
5. Configura il tuo provider SAML per supportare il single logout SAML e la firma delle richieste, quindi carica il certificato di firma del pool di utenti. Il tuo IdP deve reindirizzare verso il dominio del tuo `/saml2/logout` pool di utenti.

API/CLI

Per configurare la disconnessione SAML

Configura il logout singolo con il `IDPSignout` parametro di una richiesta CreateIdentityProvidero UpdateIdentityProviderAPI. Di seguito è riportato un esempio `ProviderDetails` di IdP che supporta il single logout SAML.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",,
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
```

```
}
```

Firma e crittografia SAML

L'accesso a SAML 2.0 si basa sull'utente di un'applicazione come portatore di richieste e risposte nel flusso di autenticazione. Potresti voler assicurarti che gli utenti non leggano o modifichino questi documenti SAML in transito. A tale scopo, aggiungi la firma e la crittografia SAML ai provider di identità SAML (IdPs) nel tuo pool di utenti. Con la firma SAML, i tuoi pool di utenti aggiungono una firma alle richieste di accesso e disconnessione SAML. Con la chiave pubblica del tuo pool di utenti, il tuo IdP può verificare che stia ricevendo richieste SAML non modificate. Quindi, quando il tuo IdP risponde e trasmette le asserzioni SAML alle sessioni del browser degli utenti, l'IdP può crittografare la risposta in modo che l'utente non possa controllare i propri attributi e autorizzazioni.

Con la firma e la crittografia SAML, tutte le operazioni crittografiche durante le operazioni SAML del pool di utenti devono generare firme e testo cifrato con chiavi fornite dal pool di utenti generate da Amazon Cognito. Al momento, non è possibile configurare un pool di utenti per firmare richieste o accettare asserzioni crittografate con una chiave esterna.

> **Note**
>
> I certificati del pool di utenti sono validi per 10 anni. Una volta all'anno, Amazon Cognito genera nuovi certificati di firma e crittografia per il tuo pool di utenti. Amazon Cognito restituisce il certificato più recente quando richiedi il certificato di firma e firma le richieste con il certificato di firma più recente. Il tuo IdP può crittografare le asserzioni SAML con qualsiasi certificato di crittografia del pool di utenti non scaduto. I certificati precedenti continuano a essere validi per tutta la loro durata e la chiave pubblica non cambia tra i certificati. Come best practice, aggiorna annualmente il certificato nella configurazione del provider.

Argomenti

- Accettazione di risposte SAML crittografate dal tuo IdP
- Firma delle richieste SAML

Accettazione di risposte SAML crittografate dal tuo IdP

Amazon Cognito e il tuo IdP possono stabilire la riservatezza nelle risposte SAML quando gli utenti accedono e si disconnettono. Amazon Cognito assegna una coppia di chiavi RSA pubblica-privata

e un certificato a ogni provider SAML esterno che configuri nel tuo pool di utenti. Quando abiliti la crittografia delle risposte per il provider SAML del tuo pool di utenti, devi caricare il certificato su un IdP che supporti le risposte SAML crittografate. La connessione del pool di utenti al tuo IdP SAML non funziona prima che l'IdP inizi a crittografare tutte le asserzioni SAML con la chiave fornita.

Di seguito è riportata una panoramica del flusso di un accesso SAML crittografato.

1. L'utente avvia l'accesso e sceglie il proprio IdP SAML.
2. Il tuo pool di utenti Endpoint Authorize reindirizza l'utente al suo IdP SAML con una richiesta di accesso SAML. Il tuo pool di utenti può facoltativamente accompagnare questa richiesta con una firma che abilita la verifica dell'integrità da parte dell'IdP. Quando desideri firmare richieste SAML, devi configurare il tuo IdP per accettare le richieste che il tuo pool di utenti ha firmato con la chiave pubblica nel certificato di firma.
3. L'IdP SAML accede al tuo utente e genera una risposta SAML. L'IdP crittografa la risposta con la chiave pubblica e reindirizza l'utente all'endpoint del pool di utenti. `/saml2/idpresponse` L'IdP deve crittografare la risposta come definito dalla specifica SAML 2.0. Per ulteriori informazioni, consulta `Element <EncryptedAssertion>` la sezione Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML). V2.0
4. Il tuo pool di utenti decrittografa il testo cifrato nella risposta SAML con la chiave privata e accede all'utente.

> **Important**
>
> Quando abiliti la crittografia delle risposte per un IdP SAML nel tuo pool di utenti, l'IdP deve crittografare tutte le risposte con una chiave pubblica specifica per il provider. Amazon Cognito non accetta risposte SAML non crittografate da un IdP esterno SAML configurato per supportare la crittografia.

Qualsiasi IdP SAML esterno nel tuo pool di utenti può supportare la crittografia delle risposte e ogni IdP riceve la propria coppia di chiavi.

Console di gestione AWS

Per configurare la crittografia delle risposte SAML

1. Crea un pool di utenti, un client di app e un IdP SAML.

2. Quando crei o modifichi il tuo provider di identità SAML, in Firma le richieste e crittografa le risposte, seleziona la casella con il titolo Richiedi asserzioni SAML crittografate da questo provider.
3. Dal menu Provider social ed esterni del tuo pool di utenti, seleziona il tuo IdP SAML e scegli Visualizza certificato di crittografia.
4. Scegli Scarica come .crt e fornisci il file scaricato al tuo IdP SAML. Configura il tuo IdP SAML per crittografare le risposte SAML con la chiave nel certificato.

API/CLI

Per configurare la crittografia delle risposte SAML

Configura la crittografia delle risposte con il `EncryptedResponses` parametro di una richiesta CreateIdentityProvidero UpdateIdentityProviderAPI. Di seguito è riportato un esempio `ProviderDetails` di IdP che supporta la firma delle richieste.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Per ottenere il certificato di crittografia dal tuo pool di utenti, effettua una richiesta DescribeIdentityProviderAPI e recupera il valore di `ActiveEncryptionCertificate` nel parametro di risposta. `ProviderDetails` Salva questo certificato e forniscilo al tuo IdP come certificato di crittografia per le richieste di accesso dal tuo pool di utenti.

Firma delle richieste SAML

La capacità di dimostrare l'integrità delle richieste SAML 2.0 al tuo IdP è un vantaggio in termini di sicurezza dell'accesso SAML di Amazon SP-initiated Cognito. Ogni pool di utenti con un dominio riceve un certificato di firma del pool di utenti. X.509 Con la chiave pubblica in questo certificato, i pool di utenti applicano una firma crittografica alle richieste di disconnessione che il pool di utenti genera quando gli utenti selezionano un IdP SAML. Facoltativamente, puoi configurare il client dell'app per firmare le richieste di accesso SAML. Quando firmi le tue richieste SAML, il tuo IdP

può verificare che la firma nei metadati XML delle tue richieste corrisponda alla chiave pubblica nel certificato del pool di utenti che fornisci.

Console di gestione AWS

Per configurare la firma delle richieste SAML

1. Crea un pool di utenti, un client di app e un IdP SAML.
2. Quando crei o modifichi il tuo provider di identità SAML, in Firma richieste e risposte crittografate, seleziona la casella con il titolo Firma le richieste SAML a questo provider.
3. Dal menu Social e provider esterni del tuo pool di utenti, scegli Visualizza certificato di firma.
4. Scegli Scarica come .crt e fornisci il file scaricato al tuo IdP SAML. Configura il tuo IdP SAML per verificare la firma delle richieste SAML in entrata.

API/CLI

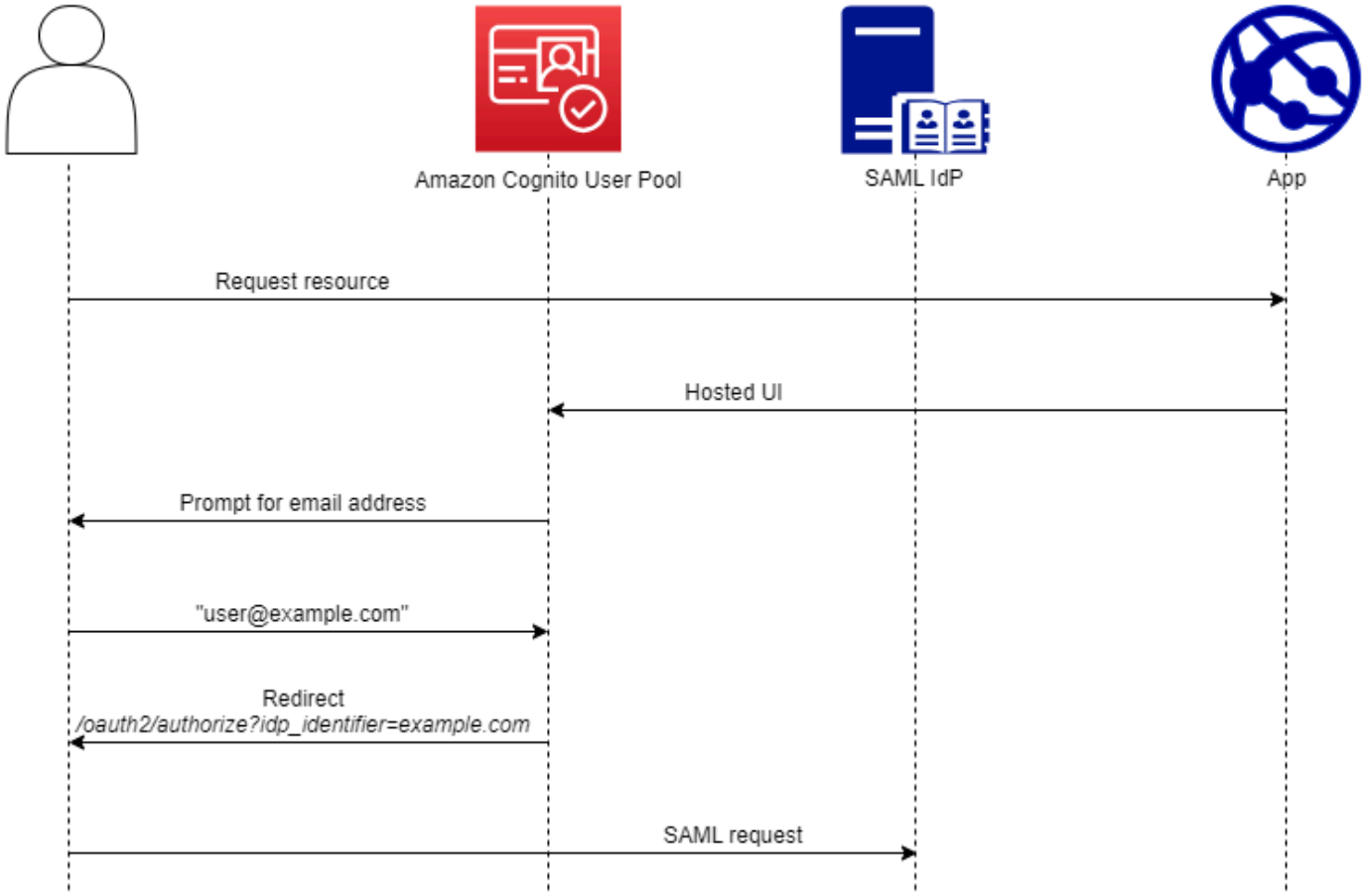
Per configurare la firma delle richieste SAML

Configura la firma delle richieste con il `RequestSigningAlgorithm` parametro di una richiesta CreateIdentityProvidero UpdateIdentityProviderAPI. Di seguito è riportato un esempio `ProviderDetails` di IdP che supporta la firma delle richieste.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Nomi e identificatori dei provider di identità SAML

Quando dai un nome ai tuoi provider di identità SAML (IdPs) e assegni identificatori IdP, puoi automatizzare il flusso di richieste di SP-initiated accesso e disconnessione a quel provider. Per informazioni sui vincoli di stringa al nome del provider, consulta la proprietà di. `ProviderName` CreateIdentityProvider



Puoi anche scegliere fino a 50 identificatori per i tuoi provider SAML. Un identificatore è un nome descrittivo per un IdP nel pool di utenti e deve essere univoco all'interno del pool di utenti. Se gli identificatori SAML corrispondono ai domini e-mail degli utenti, l'accesso gestito richiede l'indirizzo e-mail di ciascun utente, valuta il dominio nel relativo indirizzo e-mail e lo reindirizza all'IdP corrispondente al loro dominio. Poiché la stessa organizzazione può possedere più domini, un singolo IdP può avere più identificatori.

Indipendentemente dal fatto che utilizzi o meno gli identificatori di dominio e-mail, puoi utilizzare gli identificatori in un'app multi-tenant per reindirizzare gli utenti all'IdP corretto. Se desideri ignorare completamente l'accesso gestito, puoi personalizzare i link che presenti agli utenti in modo che reindirizzino Endpoint Authorize direttamente al loro IdP. Per accedere ai tuoi utenti con un identificatore e reindirizzarli al loro IdP, includi l'identificatore nel formato `idp_identifier=`*`myidp.example.com`* nei parametri di richiesta della loro richiesta di autorizzazione iniziale.

Un altro metodo per passare un utente al tuo IdP consiste nel compilare il parametro `identity_provider` con il nome del tuo IdP nel seguente formato URL.

```
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
identity_provider=MySAMLIdP&
client_id=1example23456789&
redirect_uri=https://www.example.com
```

Dopo che un utente ha effettuato l'accesso con il tuo IdP SAML, il tuo IdP lo reindirizza con una risposta SAML nel corpo dell'utente all'endpoint. `HTTP POST /saml2/idpresponse` Amazon Cognito elabora l'asserzione SAML e, se le affermazioni nella risposta soddisfano le aspettative, reindirizza all'URL di callback del client dell'app. Dopo aver completato l'autenticazione in questo modo, l'utente ha interagito con le pagine Web solo per il tuo IdP e la tua app.

Con gli identificatori IdP in un formato di dominio, l'accesso gestito richiede gli indirizzi e-mail al momento dell'accesso e quindi, quando il dominio e-mail corrisponde a un identificatore IdP, reindirizza gli utenti alla pagina di accesso del loro IdP. Ad esempio, crei un'app che richiede l'accesso da parte di dipendenti di due aziende diverse. La prima azienda, AnyCompany A, possiede `exampleA.com` `eexampleA.co.uk`. La seconda società, AnyCompany B, possiede`exampleB.com`. Per questo esempio, ne hai impostate due IdPs, una per ogni società, come segue:

- Per il provider di identità A, puoi definire gli identificatori `exampleA.com` e `exampleA.co.uk`.
- Per il provider di identità B, puoi definire l'identificatore `exampleB.com`.

Nell'app, richiama l'accesso gestito per il client dell'app per richiedere a ciascun utente di inserire il proprio indirizzo e-mail. Amazon Cognito ricava il dominio dall'indirizzo e-mail, correla il dominio a un IdP con un identificatore di dominio e reindirizza l'utente all'IdP corretto con una richiesta a quello che contiene un parametro di richiesta. Endpoint Authorize `idp_identifier` Ad esempio, se un utente entrabob`@exampleA.co.uk`, la pagina successiva con cui interagisce è la pagina di accesso IdP all'indirizzo. `https://auth.exampleA.co.uk/sso/saml`

Puoi anche implementare la stessa logica in modo indipendente. Nella tua app, puoi creare un modulo personalizzato che raccoglie l'input dell'utente e lo correla all'IdP corretto secondo la tua logica. Puoi generare portali personalizzati per ciascuno dei tenant dell'app, ognuno dei quali si collega all'endpoint di autorizzazione con l'identificatore del tenant nei parametri di richiesta.

Per raccogliere un indirizzo email e analizzare il dominio nell'accesso gestito, assegna almeno un identificatore a ciascun IdP SAML che hai assegnato al client dell'app. Per impostazione predefinita, la schermata di accesso gestito mostra un pulsante per ciascuno dei pulsanti assegnati al client IdPs dell'app. Tuttavia, se gli identificatori sono stati assegnati correttamente, la pagina di accesso all'interfaccia utente ospitata classica è simile all'immagine seguente.
Una pagina di accesso gestito da Amazon Cognito che mostra l'accesso dell'utente locale e una richiesta a un utente federato di inserire un indirizzo e-mail.

> **Note**
>
> Nella classica interfaccia utente ospitata, la pagina di accesso per il client dell'app richiede automaticamente un indirizzo e-mail quando assegni gli identificatori al tuo. IdPs Nell'esperienza di accesso gestito, devi abilitare questo comportamento nell'editor di branding. Nella categoria Impostazioni del comportamento di autenticazione, seleziona Domain search input sotto la voce Provider display.

L'analisi del dominio nell'accesso gestito richiede l'utilizzo di domini come identificatori IdP. Se assegni un identificatore di qualsiasi tipo a ciascun client SAML IdPs per un'app, l'accesso gestito per quell'app non visualizza più i pulsanti. IdP-selection Aggiungi identificatori IdP per SAML quando intendi utilizzare l'analisi delle e-mail o la logica personalizzata per generare reindirizzamenti.
Se desideri generare reindirizzamenti silenziosi e desideri che le tue pagine di accesso gestite visualizzino un elenco di IdPs, non assegnare identificatori e utilizza il parametro di richiesta nelle tue richieste di autorizzazione. `identity_provider`

- Se assegni un solo IdP SAML al client dell'app, la pagina di accesso gestito mostra un pulsante per accedere con quell'IdP.
- Se assegni un identificatore a ogni IdP SAML che attivi per il client dell'app, nella pagina di accesso gestito viene visualizzato un messaggio che richiede l'immissione di un indirizzo e-mail da parte dell'utente.
- Se ne hai più di uno IdPs e non assegni un identificatore a tutti, la pagina di accesso gestito mostra un pulsante per accedere con ogni IdP assegnato.
- Se hai assegnato degli identificatori al tuo IdPs e desideri che le tue pagine di accesso gestite mostrino una selezione di pulsanti IdP, aggiungi un nuovo IdP privo di identificatore al client dell'app o crea un nuovo app client. Puoi anche eliminare un IdP esistente e aggiungerlo di nuovo senza un identificatore. Se crei un nuovo IdP, i tuoi utenti SAML creano nuovi profili utente. Questa

duplicazione di utenti attivi potrebbe avere un impatto sulla fatturazione nel mese in cui modifichi la configurazione dell'IdP.

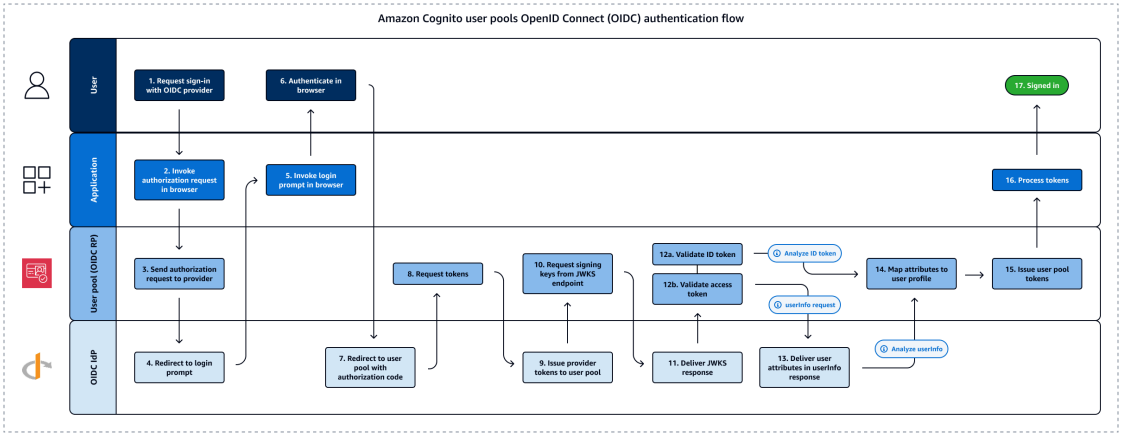
Per ulteriori informazioni sulla configurazione degli IdP, consulta Configurazione dei provider di identità per il bacino d'utenza.

Utilizzo di provider di identità OIDC con un pool di utenti

Gli utenti possono accedere all'applicazione utilizzando gli account esistenti dei provider di identità OpenID Connect (OIDC) (). IdPs Con i provider OIDC, gli utenti di sistemi single sign-on indipendenti possono fornire le credenziali esistenti mentre l'applicazione riceve i token OIDC nel formato condiviso di pool di utenti. Per configurare un IdP OIDC, configura il tuo IdP in modo che gestisca il tuo pool di utenti come RP e configura l'applicazione per gestire il tuo pool di utenti come IdP. Amazon Cognito funge da passaggio intermedio tra più IdPs OIDC e le tue applicazioni. Il tuo pool di utenti applica le regole di mappatura degli attributi alle affermazioni contenute nell'ID e nei token di accesso che il provider trasmette direttamente al tuo pool di utenti. Amazon Cognito emette quindi nuovi token in base agli attributi utente mappati e a eventuali modifiche aggiuntive apportate al flusso di autenticazione con i trigger Lambda.

Agli utenti che accedono con un IdP OIDC non è richiesto di fornire nuove credenziali o informazioni per accedere all'applicazione del pool di utenti. La tua applicazione può reindirizzarli silenziosamente al loro IdP per l'accesso, con un pool di utenti come strumento in background che standardizza il formato del token per la tua applicazione. Per ulteriori informazioni sul reindirizzamento IdP, consulta. Endpoint Authorize

Come con altri provider di identità di terze parti, devi registrare la tua applicazione presso il provider OIDC e ottenere informazioni sull'applicazione IdP che desideri connettere al tuo pool di utenti. Un pool di utenti OIDC IdP richiede un ID client, un client secret, gli ambiti da richiedere e informazioni sugli endpoint del servizio del provider. Il tuo pool di utenti può scoprire gli endpoint OIDC del provider da un endpoint di rilevamento oppure puoi inserirli manualmente. È inoltre necessario esaminare i token ID del provider e creare mappature degli attributi tra l'IdP e gli attributi del pool di utenti.



Vedi Flusso di autenticazione IdP del bacino d'utenza OIDC per maggiori dettagli su questo flusso di autenticazione.

> **Note**
>
> Sign-in tramite una terza parte (federazione) è disponibile nei pool di utenti di Amazon Cognito. Questa funzionalità è indipendente dalla federazione OIDC con i pool di identità di Amazon Cognito.

Puoi aggiungere un IdP OIDC al tuo pool di utenti nel Console di gestione AWS, tramite o con il metodo AWS CLI API del pool di utenti. CreateIdentityProvider

Argomenti

- Prerequisiti
- Registrare un'applicazione con un IdP OIDC
- Aggiunta di un IdP OIDC al bacino d'utenza
- Test della configurazione dell'IdP OIDC
- Flusso di autenticazione IdP del bacino d'utenza OIDC
- Step-up autenticazione per utenti federati

Prerequisiti

Prima di iniziare, è necessario:

- Un bacino d'utenza con un client di applicazioni e dominio per il bacino d'utenza. Per ulteriori informazioni, consulta Creazione di un bacino d'utenza.
- Un IdP OIDC con la seguente configurazione:
  - Supporta l'autenticazione client `client_secret_post`. Amazon Cognito non controlla l'attestazione `token_endpoint_auth_methods_supported` nell'endpoint di rilevamento OIDC per il tuo IdP. Amazon Cognito non supporta l'autenticazione client `client_secret_basic`. Per ulteriori informazioni sull'autenticazione client, consulta Autenticazione client nella documentazione di OpenID Connect.
  - Utilizza HTTPS solo per endpoint OIDC come `openid_configuration`, `userInfo` e `jwks_uri`.
  - Utilizza solo le porte TCP 80 e 443 per gli endpoint OIDC.
  - Firma i token ID solo con algoritmi HMAC-SHA ECDSA o RSA.
  - Pubblica un'attestazione `kid` dell'ID chiave presso il relativo `jwks_uri` e include un'attestazione `kid` nei rispettivi token.
  - Presenta una chiave pubblica non scaduta con una catena di fiducia CA root valida.

Registrare un'applicazione con un IdP OIDC

Prima di aggiungere un IdP OIDC alla configurazione del pool di utenti e assegnarlo ai client delle app, devi configurare un'applicazione client OIDC nel tuo IdP. Il pool di utenti è l'applicazione relying-party che gestisce l'autenticazione con il tuo IdP.

Registrazione con un IdP OIDC

1. Crea un account sviluppatore con l'IdP OIDC.

   Collegamenti a OIDC IdPs

IdP OIDC	Come installare	URL di rilevamento OIDC
Salesforce	Salesforce come provider di identità OpenID Connect	`https://MyDomainName.my.salesforce.com/.well-known/openid-configuration`
OneLogin	Connect un'app abilitata per OIDC	`https://your-domain.onelogin.com/`

IdP OIDC	Come installare	URL di rilevamento OIDC
		`oidc/2/.well-known/openid-configuration`
JumpCloud	SSO con OIDC	`https://oauth.id.jumpcloud.com/.well-known/openid-configuration`
Okta	Installa un provider di identità Okta	`https://`*`Your Okta subdomain`*`.okta.com/.well-known/openid-configuration`
ID Microsoft Entra	OpenID Connect sulla piattaforma di identità Microsoft	`https://login.microsoftonline.com/`*`{tenant}`*`/v2.0` I valori di `tenant` possono includere un ID tenant, `commonorganizations` , o. `consumers`

2. Registra l'URL del dominio del bacino d'utenza con l'`/oauth2/idpresponse`endpoint presso l'IdP OIDC. In questo modo verrà accettato dall'IdP OIDC durante l'autenticazione degli utenti da parte di Amazon Cognito.

   ```
   https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/idpresponse
   ```

3. Seleziona gli ambiti che desideri che la tua directory utente condivida con il tuo pool di utenti. L'ambito openid è necessario per consentire a OIDC di offrire qualsiasi informazione IdPs all'utente. L'`email`ambito è necessario per concedere l'accesso ai reclami e. `email``email_verified` Gli ambiti aggiuntivi nella specifica OIDC si riferiscono a tutti `profile` gli attributi utente e a ephone. `phone_number` `phone_number_verified`

4. L'IdP OIDC fornisce un ID client e un segreto client. Annota questi valori e aggiungili alla configurazione dell'IdP OIDC che aggiungerai successivamente al tuo pool di utenti.

Esempio: utilizza Salesforce come IdP OIDC con il tuo bacino d'utenza

Utilizzi un IdP OIDC quando desideri stabilire un rapporto di fiducia tra un OIDC-compatible IdP come Salesforce e il tuo pool di utenti.

1. Creazione di un account sul sito Web di Salesforce Developers.
2. Accedi tramite il tuo account sviluppatore impostato nella fase precedente.
3. Dalla pagina di Salesforce, procedi in una delle modalità seguenti:
   - Se utilizzi Lightning Experience, scegli l'icona a forma di ingranaggio delle impostazioni, quindi scegli Setup Home (Imposta home).
   - Se utilizzi Salesforce Classic e vedi la voce Setup (Impostazioni) nell'intestazione dell'interfaccia utente, sceglila.
   - Se utilizzi Salesforce Classic e non vedi Configurazione nell'intestazione, scegli il tuo nome dalla barra di navigazione in alto e scegli Configurazione dall'elenco a discesa.
4. Nella barra di navigazione a sinistra, scegli Company Settings (Impostazioni azienda).
5. Nella barra di navigazione, scegli Domain (Dominio), inserisci un dominio e seleziona Create (Crea).
6. Nella barra di navigazione a sinistra, vai a Platform Tools Strumenti piattaforma e scegli App.
7. Scegli App Manager Gestore app.
8. a. Scegli New connected app (Nuova app connessa).
   b. Completa i campi obbligatori.

      In Start URL (URL di avvio), inserisci un URL all'endpoint `/authorize` per il dominio del bacino d'utenza che accede con il provider di identità Salesforce. Quando gli utenti accedono all'app connessa, Salesforce li indirizza a questo URL per completare l'accesso. Quindi Salesforce reindirizza gli utenti all'URL di callback associato al client di app.

      ```
      https://mydomain.auth.us-east-1.amazoncognito.com/authorize?
      response_type=code&client_id=<your_client_id>&redirect_uri=https://
      www.example.com&identity_provider=CorpSalesforce
      ```

   c. Abilita OAuth settings (Impostazioni OAuth) e inserisci l'URL dell'endpoint `/oauth2/idpresponse` per il dominio del bacino d'utenza in Callback URL (URL di callback). Questo è l'URL da cui Salesforce emette il codice di autorizzazione che Amazon Cognito scambia con un token OAuth.

```
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/idpresponse
```

9. Seleziona i tuoi scopes (ambiti). È necessario includere openid per l'ambito. Per concedere l'accesso alle attestazioni email e email_verified aggiungi l'ambito email. Separa gli ambiti con gli spazi.
10. Scegli Create (Crea).

    In Salesforce, l'ID client viene chiamato Consumer Key (Chiave consumatore), mentre il segreto client è un Consumer Secret (Segreto consumatore). Prendi nota dell'ID client e del segreto client. Li utilizzerai nella sezione successiva.

Aggiunta di un IdP OIDC al bacino d'utenza

Dopo aver configurato l'IdP, puoi configurare il pool di utenti per gestire le richieste di autenticazione con un IdP OIDC.

Amazon Cognito console

Aggiungi un IdP OIDC nella console

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS .
2. Scegli User Pools (Bacini d'utenza) dal menu di navigazione.
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Social e provider esterni, quindi seleziona Aggiungi un provider di identità.
5. Scegli un provider di identità OpenID Connect.
6. Inserisci un nome di provider univoco.
7. Inserisci l'ID client IdP. Questo è l'ID del client applicativo che crei nel tuo IdP OIDC. L'ID client che fornisci deve essere un provider OIDC che hai configurato con un URL di callback di. `https://[your user pool domain]/oauth2/idpresponse`
8. Inserisci il segreto del client IdP. Questo deve essere il segreto del client per la stessa applicazione client del passaggio precedente.
9. Inserisci gli Authorized scopes (Ambiti autorizzati) per questo provider. Gli ambiti definiscono i gruppi di attributi utente (ad esempio `name` e`email`) che l'applicazione richiede al provider. Gli ambiti devono essere separati da spazi, secondo la specifica OAuth 2.0.

Il tuo IdP potrebbe richiedere agli utenti di acconsentire a fornire questi attributi alla tua applicazione al momento dell'accesso.

10. Scegli un metodo di richiesta degli attributi. IdPspotrebbe richiedere che le richieste ai rispettivi `userInfo` endpoint siano formattate come uno `GET` dei seguenti formati. `POST` L'`userInfo`endpoint Amazon Cognito richiede `HTTP GET` richieste, ad esempio.
11. Scegli un metodo di configurazione per il modo in cui desideri che il pool di utenti determini il percorso verso gli OIDC-federation endpoint chiave del tuo IdP. In genere, IdPs ospita un `/well-known/openid-configuration` endpoint presso un URL di base dell'emittente. Se questo è il caso del tuo provider, l'opzione Compilazione automatica attraverso l'URL dell'emittente richiede l'URL di base, tenta di accedere al `/well-known/openid-configuration` percorso da lì e legge gli endpoint ivi elencati. Potresti avere percorsi di endpoint non tipici o voler passare le richieste a uno o più endpoint tramite un proxy alternativo. In questo caso, seleziona Input manuale e specifica i percorsi per`authorization`,`token`, `userInfo` e endpoint. `jwks_uri`

    > **Note**
    >
    > L'URL deve iniziare con `https://` e non deve finire con una barra `/`. Solo i numeri di porta 443 e 80 possono essere utilizzati con questo URL. Ad esempio, Salesforce usa questo URL:
    > `https://login.salesforce.com`
    > Scegliendo il riempimento automatico, il documento di individuazione deve utilizzare HTTPS per i seguenti valori: `authorization_endpoint`, `token_endpoint`, `userinfo_endpoint`, e `jwks_uri`. In caso contrario, l'accesso avrà esito negativo.

12. Configura le tue regole di mappatura degli attributi in Map attributes tra il tuo provider OpenID Connect e il tuo pool di utenti. L'attributo del pool di utenti è l'attributo di destinazione nel profilo utente di Amazon Cognito e l'attributo OpenID Connect è l'attributo di origine che desideri che Amazon Cognito trovi in un claim o in una risposta. ID-token `userInfo` Amazon Cognito mappa automaticamente il claim sub OIDC al `username` profilo utente di destinazione.

    Per ulteriori informazioni, consulta Mappatura degli attributi IdP su profili e token.
13. Scegli Aggiungi provider di identità.

14. Dal menu App client, seleziona un client dell'app dall'elenco. Vai alla scheda Pagine di accesso e in Configurazione delle pagine di accesso gestite, seleziona Modifica. Individua i provider di identità e aggiungi il tuo nuovo IdP OIDC.
15. Scegli Save changes (Salva modifiche).

API/CLI

Vedi la configurazione OIDC nel secondo esempio su. CreateIdentityProvider È possibile modificare questa sintassi e utilizzarla come corpo della richiesta o come file di `--cli-input-json` input per `CreateIdentityProvider` create-identity-provider.

Test della configurazione dell'IdP OIDC

Nella tua applicazione, devi richiamare un browser nel client dell'utente in modo che possa accedere con il suo provider OIDC. Prova l'accesso con il tuo provider dopo aver completato le procedure di configurazione nelle sezioni precedenti. L'URL di esempio seguente carica la pagina di accesso per il tuo pool di utenti con un dominio prefisso.

```
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

Questo link è la pagina a cui Amazon Cognito ti indirizza quando accedi al menu App client, selezioni un client per l'app, accedi alla scheda Pagine di accesso e selezioni Visualizza pagina di accesso. Per ulteriori informazioni sui domini del pool di utenti, consulta. Configurazione di un dominio di bacino d'utenza Per ulteriori informazioni sui client delle app, inclusi gli ID client e gli URL di callback, consulta. Application-specific impostazioni con i client dell'app

Il seguente collegamento di esempio imposta il reindirizzamento silenzioso al `MyOIDCIdP` provider dal parametro Endpoint Authorize con un `identity_provider` parametro di query. Questo URL ignora l'accesso interattivo al pool di utenti con accesso gestito e va direttamente alla pagina di accesso IdP.

```
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
identity_provider=MyOIDCIdP&response_type=code&client_id=1example23456789&redirect_uri=https://
www.example.com
```

Flusso di autenticazione IdP del bacino d'utenza OIDC

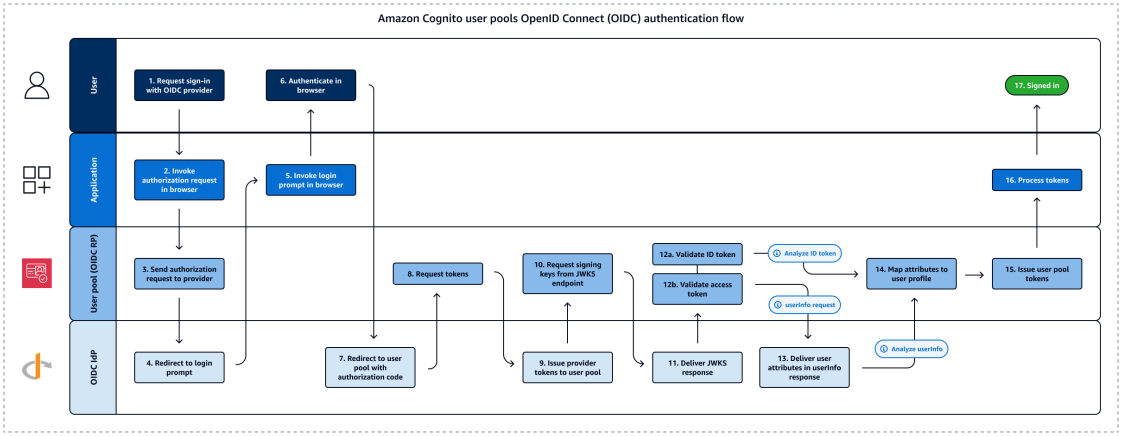
Con l'accesso a OpenID Connect (OIDC), il tuo pool di utenti automatizza un flusso di accesso con codice di autorizzazione con il tuo provider di identità (IdP). Dopo che l'utente ha completato l'accesso con il proprio IdP, Amazon Cognito raccoglie il codice sull'endpoint del provider esterno. `oauth2/idpresponse` Con il token di accesso risultante, il pool di utenti interroga l'endpoint `userInfo` IdP per recuperare gli attributi dell'utente. Il pool di utenti confronta quindi gli attributi ricevuti con le regole di mappatura degli attributi che hai impostato e compila di conseguenza il profilo e il token ID dell'utente.

Gli ambiti OAuth 2.0 richiesti nella configurazione del provider OIDC definiscono gli attributi utente che l'IdP fornisce ad Amazon Cognito. Come migliore pratica di sicurezza, richiedi solo gli ambiti che corrispondono agli attributi che desideri mappare al tuo pool di utenti. Ad esempio, se il tuo pool di utenti lo richiedeo`openid profile`, otterrai tutti gli attributi possibili, ma se lo richiedi `openid email phone_number` otterrai solo l'indirizzo email e il numero di telefono dell'utente. Puoi configurare gli ambiti richiesti a OIDC in modo che IdPs differiscano da quelli che autorizzi e richiedi nella richiesta di autenticazione del client dell'app e del pool di utenti.

Quando l'utente accede all'applicazione utilizzando un IdP OIDC, il pool di utenti esegue il seguente flusso di autenticazione.

1. Un utente accede alla tua pagina di accesso gestito e sceglie di accedere con il proprio IdP OIDC.
2. L'applicazione indirizza il browser dell'utente all'endpoint di autorizzazione del pool di utenti.
3. Il pool di utenti reindirizza la richiesta all'endpoint di autorizzazione dell'IdP OIDC.
4. Il tuo IdP visualizza una richiesta di accesso.
5. Nell'applicazione, la sessione dell'utente visualizza una richiesta di accesso per l'IdP OIDC.
6. L'utente inserisce le proprie credenziali per l'IdP o presenta un cookie per una sessione già autenticata.
7. Dopo l'autenticazione dell'utente, l'IdP OIDC reindirizza ad Amazon Cognito con un codice di autorizzazione.
8. Il tuo pool di utenti scambia il codice di autorizzazione con ID e token di accesso. Amazon Cognito riceve token di accesso quando configuri il tuo IdP con gli ambiti. `openid` Le affermazioni nel token ID e la `userInfo` risposta sono determinate da ambiti aggiuntivi della configurazione IdP, ad esempio `profile` e. `email`
9. Il tuo IdP emette i token richiesti.

10. Il tuo pool di utenti determina il percorso dell'`jwks_uri`endpoint IdP dagli URL dell'emittente nella configurazione dell'IdP e richiede le chiavi di firma del token dall'endpoint JSON web key set (JWKS).
11. L'IdP restituisce le chiavi di firma dall'endpoint JWKS.
12. Il tuo pool di utenti convalida i token IdP in base ai dati di firma e scadenza contenuti nei token.
13. Il tuo pool di utenti autorizza una richiesta all'endpoint `userInfo` IdP con il token di accesso. L'IdP risponde con i dati dell'utente in base agli ambiti del token di accesso.
14. Il tuo pool di utenti confronta il token ID e la `userInfo` risposta dell'IdP con le regole di mappatura degli attributi nel tuo pool di utenti. Scrive gli attributi IdP mappati negli attributi del profilo del pool di utenti.
15. Amazon Cognito genera i token di connessione dell'applicazione, che potrebbero includere i token di identità, accesso e aggiornamento.
16. L'applicazione elabora i token del pool di utenti e accede all'utente.



> **Note**
>
> Amazon Cognito annulla le richieste di autenticazione che non vengono completate entro 5 minuti e reindirizza l'utente all'accesso gestito. Viene visualizzato il messaggio di errore `Something went wrong` nella pagina.

OIDC è un livello di identità che si aggiunge a OAuth 2.0, che specifica i token di identità JSON-formatted (JWT) emessi dalle app client OIDC (relying party). IdPs Per informazioni su come aggiungere Amazon Cognito come relying party OIDC, consultare la documentazione dell'IdP OIDC.

Quando un utente esegue l'autenticazione con una concessione del codice di autorizzazione, il pool di utenti restituisce i token ID, di accesso e di aggiornamento. Il token ID è un token OIDC standard per la gestione delle identità, mentre il token di accesso è un token OAuth 2.0 standard. Per ulteriori informazioni sui tipi di concessione che il client app del pool di utenti può supportare, consulta Endpoint Authorize.

In che modo un pool di utenti elabora le richieste di un provider OIDC

Quando l'utente completa l'accesso con un provider OIDC di terze parti, l'accesso gestito recupera un codice di autorizzazione dall'IdP. Il pool di utenti scambia il codice di autorizzazione per i token di accesso e ID con l'endpoint `token` del gestore dell'identità digitale. Il pool di utenti non passa questi token all'utente o all'app, ma li utilizza per creare un profilo utente con i dati resi disponibili nelle richieste nei propri token.

Amazon Cognito non convalida in modo indipendente il token di accesso. Richiede invece informazioni sugli attributi utente dall'endpoint `userInfo` del provider e si aspetta che la richiesta venga rifiutata se il token non è valido.

Amazon Cognito convalida il token ID del provider con i seguenti controlli:

1. Verifica che il provider abbia firmato il token con un algoritmo del seguente set: RSA, HMAC, crittografia a curva ellittica.
2. Se il provider ha firmato il token con un algoritmo di firma asimmetrico, verifica che l'ID della chiave di firma nella richiesta `kid` del token sia presente nell'endpoint `jwks_uri` del provider. Amazon Cognito aggiorna la chiave di firma dall'endpoint JWKS nella configurazione IdP per ogni token ID IdP che elabora.
3. Confronta la firma del token ID con la firma prevista in base ai metadati del provider.
4. Confronta la richiesta `iss` con l'emittente OIDC configurata per il gestore dell'identità digitale.
5. Verifica che la richiesta `aud` corrisponda all'ID client configurato nel gestore dell'identità digitale o che contenga l'ID client configurato se sono presenti più valori nella richiesta `aud`.
6. Verifica che il timestamp nella richiesta `exp` non sia anteriore all'ora corrente.

Il pool di utenti convalida il token ID, quindi tenta una richiesta all'endpoint `userInfo` del provider con il token di accesso del provider. Recupera tutte le informazioni sul profilo utente che gli ambiti definiti nel token di accesso autorizzano a leggere. Il pool di utenti cerca quindi gli attributi utente impostati come obbligatori nel pool di utenti. Per gli attributi obbligatori è necessario creare mappature degli attributi nella configurazione del provider. Il pool di utenti controlla il token ID del

provider e la risposta `userInfo`. Il pool di utenti scrive tutte le richieste corrispondenti alle regole di mappatura negli attributi utente sul profilo utente del pool di utenti. Il pool di utenti ignora gli attributi che corrispondono a una regola di mappatura ma che non sono obbligatori e non sono presenti nelle richieste del provider.

Step-up autenticazione per utenti federati

Per gli utenti federati che accedono tramite un provider di identità (IdP) OpenID Connect (OIDC), Amazon Cognito utilizza i valori ACR e AMR dell'IdP OIDC anziché elaborarli autonomamente. L'IdP OIDC possiede l'autenticazione per questi utenti federati. Step-up l'autenticazione per un utente federato avviene tramite l'IdP OIDC: Amazon Cognito reindirizza all'IdP con. `acr_values` Per ulteriori informazioni sui livelli ACR e sui valori AMR, consulta. the section called “Livelli di autenticazione con attestazioni ACR e AMR”

Puoi mappare i valori ACR dell'IdP OIDC ai livelli di Amazon Cognito con `AcrMapping` il campo sull'IdP. Configura questo campo con l'operazione or. CreateIdentityProviderUpdateIdentityProvider Amazon Cognito mappa il valore ACR dell'IdP OIDC a un livello di Amazon Cognito, quindi timbra il nome del livello corrispondente che hai configurato per il tuo pool di utenti sul token Amazon Cognito. L'esempio seguente configura un per un provider OIDC. `AcrMapping`

```
{
  "ProviderName": "MyOktaIdP",
  "ProviderType": "OIDC",
  "AcrMapping": {
    "Level1": "urn:okta:loa:1",
    "Level2": "urn:okta:loa:2",
    "Level3": "urn:okta:loa:3",
    "Level4": "urn:okta:loa:4"
  }
}
```

L'impostazione `AcrMapping` del provider di identità è disponibile in tutti i piani di funzionalità. Non è limitato al piano di funzionalità Essentials o Plus.

Il seguente comportamento si applica agli utenti federati che accedono tramite un IdP OIDC.

- Se l'IdP OIDC restituisce un valore ACR che non è nella configurazione, Amazon `AcrMapping` Cognito imposta la `acr` dichiarazione del token al livello più basso (il nome predefinito `urn:cognito:loa:1` o il nome di livello 1 che hai configurato per il tuo pool di utenti). Questo non è un errore.

- Amazon Cognito elimina qualsiasi valore AMR che non riconosce e mantiene solo i valori noti.
- Se l'IdP OIDC non supporta ACR e AMR o non restituisce valori AMR riconosciuti da Amazon Cognito, Amazon Cognito imposta come valore AMR segnaposto predefinito. `fed`
- Amazon Cognito si fida dei `amr` valori `acr` e dei valori che l'IdP OIDC afferma, soggetti a mappatura e filtraggio. Non convalida le sfide effettivamente eseguite dall'IdP.

In un flusso federativo, l'applicazione reindirizza all'`/oauth2/authorize`endpoint con il livello o i livelli di destinazione espressi nei nomi del pool di utenti. Amazon Cognito traduce i valori nei valori che l'IdP OIDC si aspetta con, `AcrMapping` reindirizza all'IdP, riceve i valori ACR e AMR dell'IdP, li mappa ed emette token Amazon Cognito. Per ulteriori informazioni sulla configurazione di un IdP, consulta Configurazione dei provider di identità.

Mappatura degli attributi IdP su profili e token

I servizi di provider di identità (IdP), incluso Amazon Cognito, in genere possono registrare più informazioni su un utente. Potresti voler sapere per quale azienda lavorano, come contattarli e altre informazioni identificative. Tuttavia, il formato utilizzato da questi attributi presenta variazioni tra i provider. Ad esempio, configura tre IdPs fornitori diversi con il tuo pool di utenti ed esamina un esempio di asserzione SAML, token ID o `userInfo` payload di ciascuno. Uno rappresenterà l'indirizzo e-mail dell'utente come`email`, un altro come e il terzo come`emailaddress`. `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

Uno dei principali vantaggi derivanti dal consolidamento IdPs con un pool di utenti è la capacità di mappare la varietà di nomi di attributi in un unico schema di token OIDC con nomi di attributi coerenti, prevedibili e condivisi. In questo modo, gli sviluppatori non sono tenuti a mantenere la logica per l'elaborazione di una varietà complessa di eventi Single Sign-On. Questo consolidamento del formato è la mappatura degli attributi. La mappatura degli attributi del pool di utenti assegna i nomi degli attributi IdP ai nomi degli attributi del pool di utenti corrispondenti. Ad esempio, puoi configurare il tuo pool di utenti per scrivere il valore di un `emailaddress` claim nell'attributo standard del pool di utenti. `email`

Ogni IdP del pool di utenti ha uno schema di mappatura degli attributi separato. Per specificare le mappature degli attributi per il tuo IdP, configura un provider di identità del pool di utenti nella console Amazon Cognito, in un AWS SDK o nell'API REST dei pool di utenti.

Cose da sapere sulle mappature

Prima di iniziare a configurare la mappatura degli attributi utente, esamina i seguenti dettagli importanti.

- Prima di poter mappare un claim IdP a un attributo personalizzato, tale attributo personalizzato deve già esistere nello schema del pool di utenti. Gli attributi personalizzati vengono definiti quando si crea il pool di utenti o li si aggiunge in seguito e non è possibile mappare un attributo IdP a un attributo personalizzato che non è stato ancora aggiunto allo schema. Per ulteriori informazioni sulla definizione degli attributi nel pool, vedere. Utilizzo degli attributi utente
- Quando un utente federato effettua l'accesso all'applicazione, per ogni attributo del pool di utenti richiesto deve essere presente una mappatura. Ad esempio, se il pool di utenti richiede un attributo `email` per l'accesso, mappa questo attributo al suo equivalente del gestore dell'identità digitale (IdP).
- Di default, gli indirizzi di posta elettronica mappati non sono verificati. Non è possibile verificare un indirizzo e-mail mappato utilizzando un codice una tantum. Tuttavia, puoi mappare un attributo del provider di identità per ottenere lo stato di verifica. Ad esempio, Google e la maggior parte dei provider OIDC includono l'attributo `email_verified`.
- Puoi mappare i token del gestore dell'identità digitale agli attributi personalizzati nel pool di utenti. I provider social visualizzano un token di accesso, mentre i provider OIDC visualizzano un token di accesso e ID. Per mappare un token, aggiungi un attributo personalizzato con una lunghezza massima di 2.048 caratteri, concedi al client dell'app l'accesso in scrittura all'attributo e mappa `access_token` o `id_token` dal gestore dell'identità digitale all'attributo personalizzato.
- Per ogni attributo del pool di utenti mappato, la lunghezza massima del valore (2.048 caratteri) deve essere sufficiente per il valore recuperato da Amazon Cognito dal gestore dell'identità digitale. In caso contrario, Amazon Cognito genera un errore quando gli utenti accedono alla tua applicazione. Amazon Cognito non supporta la mappatura tra token del gestore dell'identità digitale e attributi personalizzati quando i token hanno una lunghezza superiore a 2.048 caratteri.
- Amazon Cognito ricava l'`username`attributo nel profilo di un utente federato da affermazioni specifiche approvate dal tuo IdP federato, come mostrato nella tabella seguente. Amazon Cognito, ad esempio, aggiunge questo valore di attributo al nome del tuo IdP. `MyOIDCIdP_[sub]` Se desideri che i tuoi utenti federati abbiano un attributo che corrisponda esattamente a un attributo nella tua directory utente esterna, mappalo a un attributo di accesso di Amazon Cognito come. `preferred_username`

Gestore dell'identità digitale	Attributo di origine **`username`**
Facebook	`id`
Google	`sub`
Login with Amazon	`user_id`
Accedi con Apple	`sub`
Provider SAML	`NameID`
Provider OpenID Connect (OIDC)	`sub`

- Quando un pool di utenti non fa distinzione tra maiuscole e minuscole, Amazon Cognito converte l'intero nome utente generato automaticamente di un utente federato in lettere minuscole, incluso il prefisso del nome IdP. Di seguito è riportato un esempio di nome utente per un pool di utenti con distinzione tra maiuscole e minuscole:. `MySAML_TestUser@example.com` Quello che segue è lo stesso nome utente per un pool di utenti senza distinzione tra maiuscole e minuscole:. `mysaml_testuser@example.com`

  Nei pool di utenti che non fanno distinzione tra maiuscole e minuscole, i trigger Lambda che elaborano il nome utente devono tenere conto delle modifiche all'intero nome utente. Queste modifiche includono tutti i valori misti degli attributi di origine del nome utente mappati dal tuo IdP. Per collegare il tuo IdP a un pool di utenti con un'impostazione di distinzione tra maiuscole e minuscole diversa rispetto al pool corrente, crea un nuovo pool di utenti.
- Amazon Cognito deve essere in grado di aggiornare gli attributi mappati del bacino d'utenza quando gli utenti accedono alla tua applicazione. Quando un utente accede tramite un provider di identità, Amazon Cognito aggiorna gli attributi mappati con le informazioni più recenti del provider di identità. Amazon Cognito aggiorna gli attributi mappati solo quando i loro valori cambiano. Per garantire che Amazon Cognito possa aggiornare gli attributi, controlla i seguenti requisiti:
  - Tutti gli attributi personalizzati del pool di utenti mappati dal gestore dell'identità digitale (IdP) devono essere mutabili. Puoi aggiornare gli attributi personalizzati modificabili in qualsiasi momento. Al contrario, puoi impostare un valore per un attributo personalizzato immutabile di un utente solo quando il profilo utente viene creato. Per creare un attributo personalizzato modificabile nella console Amazon Cognito, attiva la casella di controllo Mutabile per l'attributo che aggiungi quando selezioni Aggiungi attributi personalizzati nel menu. Sign-up Oppure, se

crei il tuo pool di utenti utilizzando l'operazione CreateUserPoolAPI, puoi impostare il `Mutable` parametro per ciascuno di questi attributi su. `true` Se il tuo IdP invia un valore per un attributo immutabile mappato, Amazon Cognito restituisce un errore e l'accesso non riesce.

- Nelle impostazioni del client di applicazioni, gli attributi mappati devono essere scrivibili. Puoi impostare gli attributi scrivibili nella pagina Client di applicazioni nella console Amazon Cognito. In alternativa, se crei il client dell'app utilizzando l'operazione API `CreateUserPoolClient`, puoi aggiungere questi attributi alla matrice `WriteAttributes`. Se il tuo IdP invia un valore per un attributo non scrivibile mappato, Amazon Cognito non imposta il valore dell'attributo e procede con l'autenticazione.

- Quando gli attributi IdP contengono più valori, Amazon Cognito appiattisce tutti i valori in un'unica stringa delimitata da virgole racchiusa tra parentesi quadre e. `[ ]` Il modulo URL di Amazon Cognito codifica i valori contenenti caratteri non alfanumerici ad eccezione di,, e. `. - * _` È necessario decodificare e analizzare i singoli valori prima di utilizzarli nell'app.
- L'attributo di destinazione conserva qualsiasi valore che le regole di mappatura degli attributi gli assegnano, a meno che non venga modificato da un'azione amministrativa o di accesso. Amazon Cognito non rimuove gli attributi dagli utenti quando l'attributo di origine non viene più inviato nel token del provider o nell'asserzione SAML. Le seguenti azioni rimuovono il valore di un attributo dal profilo di un pool di utenti per un utente federato:
  1. L'IdP invia un valore vuoto per l'attributo di origine e una regola di mappatura applica il valore vuoto all'attributo di destinazione.
  2. Si cancella il valore dell'attributo mappato con una DeleteUserAttributesrichiesta or. AdminDeleteUserAttributes

Specificare le mappature degli attributi del provider di identità per il pool di utenti (Console di gestione AWS)

È possibile utilizzare Console di gestione AWS per specificare le mappature degli attributi per l'IdP nel pool di utenti.

> **Note**
>
> Amazon Cognito associa le attestazioni in entrata agli attributi del pool di utenti solo se le attestazioni esistono nel token in entrata. Se una registrazione mappata in precedenza non esiste più nel token in ingresso, non verrà eliminata o modificata. Se la tua applicazione richiede la mappatura delle attestazioni eliminate, puoi utilizzare il trigger Pre-Authentication

Lambda per eliminare l'attributo personalizzato durante l'autenticazione e consentire a questi attributi di ripopolarsi dal token in entrata.

Specificazione della mappatura degli attributi di provider di identità social

1. Accedi alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegli il menu Social e provider esterni.
4. Scegli l'opzione Add an identity provider (Aggiungi un provider di identità) oppure scegli il provider di identità Facebook, Google, Amazon o Apple che hai configurato. Individua Attribute mapping (Mappatura degli attributi) e scegli Edit (Modifica).

   Per ulteriori informazioni su come aggiungere un provider di identità social, consulta Utilizzo di provider di identità social con un pool di utenti.
5. Esegui i seguenti passaggi per ciascun attributo per cui intendi eseguire la mappatura:

   a. Seleziona un attributo dalla colonna User pool attribute (Attributo bacino d'utenza). Questo è l'attributo assegnato al profilo utente nel bacino d'utenza. Gli attributi personalizzati vengono elencati dopo gli attributi standard.
   b. Seleziona un attributo dalla colonna ***<provider>***degli attributi. Questo è l'attributo passato dalla directory del provider. Gli attributi noti del provider social sono forniti in un elenco a discesa.
   c. Per mappare attributi aggiuntivi tra il tuo IdP e Amazon Cognito, scegli Add another attribute (Aggiungi un altro attributo).
6. Scegli Save changes (Salva modifiche).

Specificazione di mappatura di attributo del provider SAML

1. Accedi alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegli il menu Social e provider esterni.
4. Scegli l'opzione Add an identity provider (Aggiungi un provider di identità) oppure seleziona il provider di identità SAML configurato. Individua Attribute mapping (Mappatura degli attributi) e

scegli Edit (Modifica). Per ulteriori informazioni su come aggiungere un provider di identità SAML, consulta Utilizzo di provider di identità SAML con un pool di utenti.

5. Esegui i seguenti passaggi per ciascun attributo per cui intendi eseguire la mappatura:

   a. Seleziona un attributo dalla colonna User pool attribute (Attributo bacino d'utenza). Questo è l'attributo assegnato al profilo utente nel bacino d'utenza. Gli attributi personalizzati vengono elencati dopo gli attributi standard.

   b. Seleziona un attributo dalla colonna SAML attribute (attributo SAML). Questo è l'attributo passato dalla directory del provider.

      Il provider di identità potrebbe offrire esempi di asserzioni SAML come riferimento. Alcuni IdPs usano nomi semplici, come`email`, mentre altri usano nomi di URL-formatted attributi simili a:

      ```
      http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
      ```

   c. Per mappare attributi aggiuntivi tra il tuo IdP e Amazon Cognito, scegli Add another attribute (Aggiungi un altro attributo).

6. Scegli Save changes (Salva modifiche).

Specificare le mappature degli attributi del provider di identità per il pool di utenti (AWS CLI and AWS API)

Il seguente corpo di richiesta per CreateIdentityProvidero UpdateIdentityProvidermappa gli attributi `emailaddress` "MyIdP» del provider SAML e `phone` gli attributi del pool di utenti e `email` `birthdate``phone_number`, in quest'ordine. `birthdate` Si tratta di un corpo di richiesta completo per un provider SAML 2.0: il corpo della richiesta varia a seconda del tipo di IdP e dei dettagli specifici. La mappatura degli attributi si trova nel parametro. `AttributeMapping`

```
{
   "AttributeMapping": {
      "email" : "emailaddress",
      "birthdate" : "birthdate",
      "phone_number" : "phone"
   },
   "IdpIdentifiers": [
      "IdP1",
      "pdxsaml"
   ],
```

```
   "ProviderDetails": {
      "IDPInit": "true",
      "IDPSignout": "true",
      "EncryptedResponses" : "true",
      "MetadataURL": "https://auth.example.com/sso/saml/metadata",
      "RequestSigningAlgorithm": "rsa-sha256"
   },
   "ProviderName": "MyIdP",
   "ProviderType": "SAML",
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Utilizza i seguenti comandi per specificare le mappature degli attributi del provider di identità per il bacino d'utenza.

Specificazione di mappature degli attributi al momento della creazione del provider

- AWS CLI: `aws cognito-idp create-identity-provider`

  Esempio con file di metadati: `aws cognito-idp create-identity-provider --user-pool-id` *`<user_pool_id>`* `--provider-name=SAML_provider_1 --provider-type SAML --provider-details file:///details.json --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

  Dove `details.json` contiene:

  ```
  {
      "MetadataFile": "<SAML metadata XML>"
  }
  ```

  > **Note**
  >
  > Se *`<SAML metadata XML>`* contiene delle virgolette ("), queste devono essere escluse (). `\"`

  Esempio con URL di metadati:

  ```
  aws cognito-idp create-identity-provider \
  --user-pool-id us-east-1_EXAMPLE \
  --provider-name=SAML_provider_1 \
  ```

```
--provider-type SAML \
--provider-details MetadataURL=https://myidp.example.com/saml/metadata \
--attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
emailaddress
```

- API/SDK: CreateIdentityProvider

Specificazione delle mappature degli attributi per un provider di identità esistente

- AWS CLI: `aws cognito-idp update-identity-provider`

  Ad esempio: `aws cognito-idp update-identity-provider --user-pool-id <user_pool_id> --provider-name <provider_name> --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`
- API/SDK: UpdateIdentityProvider

Acquisizione di informazioni sulla mappatura degli attributi per un provider di identità specifico

- AWS CLI: `aws cognito-idp describe-identity-provider`

  Ad esempio: `aws cognito-idp describe-identity-provider --user-pool-id <user_pool_id> --provider-name <provider_name>`
- API/SDK: DescribeIdentityProvider

Collegamento di utenti federati a un profilo utente esistente

Spesso, lo stesso utente ha un profilo con più provider di identità (IdPs) che hai collegato al tuo pool di utenti. Amazon Cognito può collegare ogni occorrenza di un utente allo stesso profilo utente nella directory. In questo modo, una persona con più utenti IdP può avere un'esperienza coerente con la tua app. AdminLinkProviderForUserindica ad Amazon Cognito di riconoscere l'ID univoco di un utente nella tua directory federata come utente del pool di utenti. Un utente nel pool di utenti conta come un utente attivo mensile (MAU) ai fini della fatturazione quando hai zero o più identità federate associate al profilo utente.

Quando un utente federato accede al tuo pool di utenti per la prima volta, Amazon Cognito cerca un profilo locale che hai collegato alla sua identità. Se non esiste alcun profilo collegato, il tuo pool di utenti crea un nuovo profilo. Puoi creare un profilo locale e collegarlo al tuo utente federato in qualsiasi momento prima del primo accesso, in una richiesta `AdminLinkProviderForUser`

API, in un'attività di preconfigurazione pianificata o in una. Trigger Lambda di pre-registrazione Dopo che l'utente ha effettuato l'accesso e Amazon Cognito ha rilevato un profilo locale collegato, il pool di utenti legge le dichiarazioni dell'utente e le confronta con le regole di mappatura per l'IdP. Il tuo pool di utenti aggiorna quindi il profilo locale collegato con le attestazioni mappate in base al loro accesso. In questo modo, puoi configurare il profilo locale con le attestazioni di accesso e mantenere aggiornate le relative dichiarazioni di identità con il tuo provider. Dopo che Amazon Cognito ha associato il tuo utente federato a un profilo collegato, quest'ultimo accede sempre a quel profilo. Puoi quindi collegare più identità di provider del tuo utente allo stesso profilo, offrendo a un cliente un'esperienza coerente con la tua app. Per collegare un utente federato che ha precedentemente effettuato l'accesso, devi prima eliminare il suo profilo esistente. Puoi identificare i profili esistenti in base al loro formato: *`[Provider name]`*`_identifier`. Ad esempio, `LoginWithAmazon_amzn1.account.AFAEXAMPLE`. Un utente che hai creato e poi collegato a un'identità utente di terze parti ha lo stesso nome utente con cui è stato creato e un `identities` attributo che contiene i dettagli delle identità collegate.

> **Important**
>
> Poiché `AdminLinkProviderForUser` consente a un utente con un'identità federata esterna di accedere come utente esistente nel pool di utenti, è fondamentale che venga utilizzato solo con attributi esterni IdPs e del provider considerati attendibili dal proprietario dell'applicazione.

Ad esempio, se sei un fornitore di servizi gestiti (MSP) con un'app che condividi con più clienti. Ciascun cliente accede all'app tramite Active Directory Federation Services (ADFS). L'amministratore IT, Carlos, ha un account in ciascuno dei domini dei clienti. Vuoi che Carlos venga riconosciuto come amministratore dell'app ogni volta che accede, indipendentemente dal gestore dell'identità digitale (IdP).

Il tuo ADFS IdPs presenta l'indirizzo e-mail di Carlos `msp_carlos@example.com` nel `email` reclamo delle affermazioni SAML di Carlos ad Amazon Cognito. Crei un utente nel tuo pool di utenti con il nome utente. `Carlos` I seguenti comandi AWS Command Line Interface (AWS CLI) collegano le identità di Carlos da IdPs ADFS1, ADFS2 e ADFS3.

> **Note**
>
> È possibile collegare un utente in base alle asserzioni di attributi specifici. Questa capacità è esclusiva di OIDC e SAML. IdPs Per altri tipi di gestori, è necessario collegarsi in base a

un attributo di origine fisso. Per ulteriori informazioni, consulta AdminLinkProviderForUser. È necessario impostare `ProviderAttributeName` su `Cognito_Subject` quando colleghi un IdP social a un profilo utente. `ProviderAttributeValue` deve essere l'identificatore univoco dell'utente con l'IdP.

```
aws cognito-idp admin-link-provider-for-user \
--user-pool-id us-east-1_EXAMPLE \
--destination-user ProviderAttributeValue=Carlos,ProviderName=Cognito \
--source-user
 ProviderName=ADFS1,ProviderAttributeName=email,ProviderAttributeValue=msp_carlos@example.com

aws cognito-idp admin-link-provider-for-user \
--user-pool-id us-east-1_EXAMPLE \
--destination-user ProviderAttributeValue=Carlos,ProviderName=Cognito \
--source-user
 ProviderName=ADFS2,ProviderAttributeName=email,ProviderAttributeValue=msp_carlos@example.com

aws cognito-idp admin-link-provider-for-user \
--user-pool-id us-east-1_EXAMPLE \
--destination-user ProviderAttributeValue=Carlos,ProviderName=Cognito \
--source-user
 ProviderName=ADFS3,ProviderAttributeName=email,ProviderAttributeValue=msp_carlos@example.com
```

Il profilo utente `Carlos` nel pool di utenti ora dispone del seguente attributo `identities`.

```
[{
    "userId": "msp_carlos@example.com",
    "providerName": "ADFS1",
    "providerType": "SAML",
    "issuer": "http://auth.example.com",
    "primary": false,
    "dateCreated": 111111111111111
}, {
    "userId": "msp_carlos@example.com",
    "providerName": "ADFS2",
    "providerType": "SAML",
    "issuer": "http://auth2.example.com",
    "primary": false,
    "dateCreated": 111111111111111
}, {
    "userId": "msp_carlos@example.com",
```

```
    "providerName": "ADFS3",
    "providerType": "SAML",
    "issuer": "http://auth3.example.com",
    "primary": false,
    "dateCreated": 111111111111111
}]
```

Informazioni importanti sul collegamento di utenti federati

- È possibile collegare fino a cinque utenti federati a ciascun profilo utente.
- Puoi collegare gli utenti a ciascun IdP da un massimo di cinque attestati di attributo IdP, come definito dal `ProviderAttributeName` parametro di `SourceUser` in una richiesta API. `AdminLinkProviderForUser` Ad esempio, se hai collegato almeno un utente agli attributi di origine`email`,`phone`, `department` `given_name`, `location`e puoi collegare solo altri utenti su uno di questi cinque attributi.
- È possibile collegare utenti federati a un profilo utente federato esistente o a un utente locale.
- Non è possibile collegare i provider ai profili utente in Console di gestione AWS.
- Il token ID dell'utente contiene tutti i gestori associati nell'asserzione `identities`.
- È possibile impostare una password per il profilo utente federato creato automaticamente in una richiesta AdminSetUserPasswordAPI. Lo stato dell'utente cambia quindi da `EXTERNAL_PROVIDER` a `CONFIRMED`. Un utente in questo stato può accedere come utente federato e avviare flussi di autenticazione nell'API come un utente locale collegato. Possono anche modificare la password e gli attributi nelle richieste API autenticate da token come e. ChangePasswordUpdateUserAttributes Come best practice di sicurezza e per mantenere gli utenti sincronizzati con il tuo IdP esterno, non impostare la password per i profili di utenti federati. Collega invece gli utenti ai profili locali con `AdminLinkProviderForUser`.
- Amazon Cognito compila gli attributi utente in un profilo utente locale collegato quando l'utente accede tramite il proprio gestore dell'identità digitale. Amazon Cognito elabora le dichiarazioni di identità nel token ID di un IdP OIDC e controlla anche l'endpoint `userInfo` dei provider OAuth 2.0 e OIDC. Amazon Cognito assegna la priorità alle informazioni in un token ID rispetto alle informazioni di `userInfo`.

Quando scopri che il tuo utente non utilizza più un account utente esterno che hai collegato al suo profilo, puoi dissociare quell'account utente dal tuo utente del pool di utenti. Quando hai collegato il tuo utente, hai fornito il nome dell'attributo utente, il valore dell'attributo e il nome del provider

nella richiesta. Per rimuovere un profilo di cui l'utente non ha più bisogno, effettua una richiesta AdminDisableProviderForUserAPI con parametri equivalenti.

Vedi AdminLinkProviderForUserla sintassi dei comandi e gli esempi aggiuntivi negli AWS SDK.

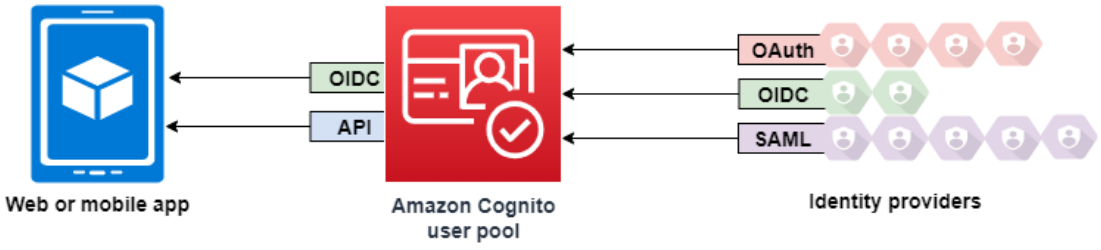
Accesso alle risorse dopo aver effettuato correttamente l'accesso

Gli utenti della tua app possono accedere direttamente tramite un pool di utenti oppure possono federarsi tramite un provider di identità (IdP) di terze parti. Il pool di utenti gestisce il sovraccarico di gestione dei token restituiti dall'accesso social tramite Facebook, Google, Amazon e Apple e da OpenID Connect (OIDC) e SAML. IdPs Per ulteriori informazioni, consulta Comprensione dei token web JSON (JWT) del pool di utenti.

Dopo un'autenticazione riuscita, l'app riceve i token del pool di utenti da Amazon Cognito. Puoi utilizzare i token del pool di utenti per:

- Recupera AWS le credenziali che autorizzano le richieste di risorse applicative in Amazon Servizi AWS DynamoDB e Amazon S3.
- Fornisci una prova di autenticazione temporanea e revocabile.
- Inserisci i dati di identità in un profilo utente nella tua app.
- Autorizza le modifiche al profilo dell'utente che ha effettuato l'accesso nella directory del pool di utenti.
- Autorizza le richieste di informazioni sugli utenti con un token di accesso.
- Autorizza le richieste ai dati che si trovano dietro API esterne protette dall'accesso con token di accesso.
- Autorizza l'accesso agli asset applicativi archiviati sul client o sul server con Amazon Verified Permissions.

Per ulteriori informazioni, consulta Un esempio di sessione di autenticazione e Comprensione dei token web JSON (JWT) del pool di utenti.



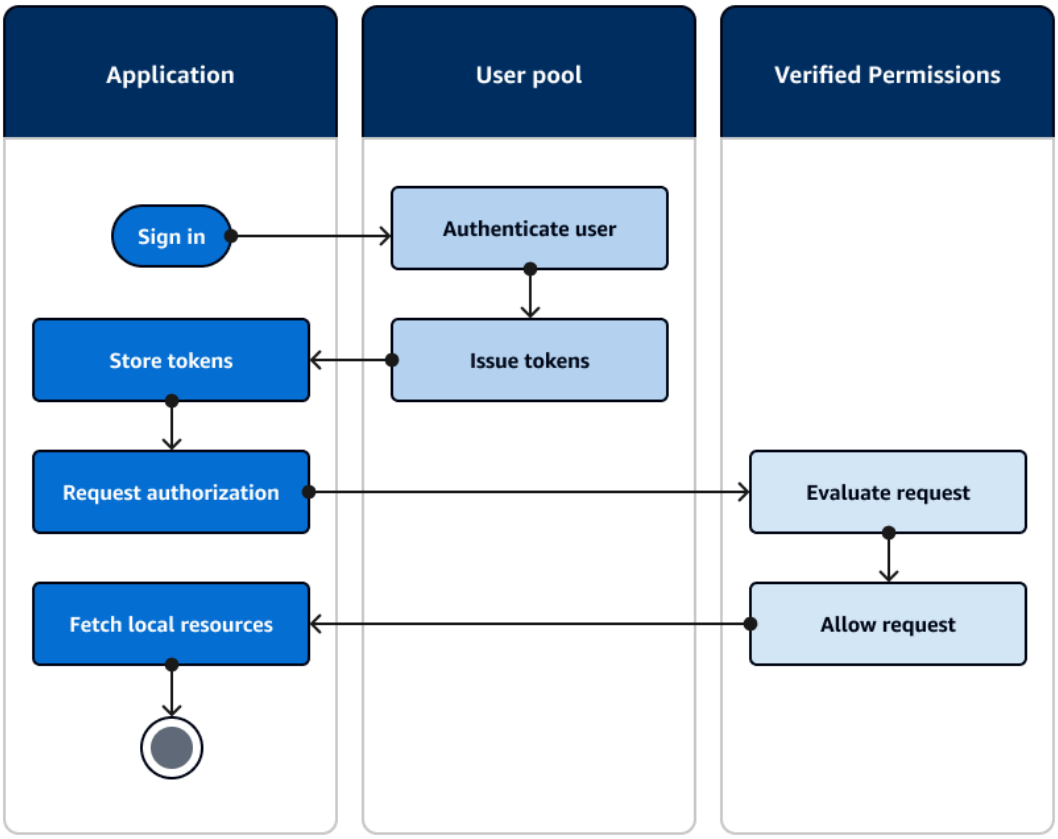
Argomenti

- Autorizzazione dell'accesso alle risorse del client o del server con Amazon Verified Permissions
- Accesso alle risorse con API Gateway dopo l'accesso
- Accesso Servizi AWS utilizzo di un pool di identità dopo l'accesso

Autorizzazione dell'accesso alle risorse del client o del server con Amazon Verified Permissions

La tua app può trasferire i token da un utente registrato ad Amazon Verified Permissions. Verified Permissions è un servizio scalabile e dettagliato di gestione e autorizzazione delle autorizzazioni per le applicazioni che hai creato. Un pool di utenti di Amazon Cognito può fungere da fonte di identità per un archivio di policy di Autorizzazioni verificate. Verified Permissions prende decisioni di autorizzazione per le azioni e le risorse richieste, ad `GetPhoto` esempio`premium_badge.png`, a partire dal principale e dai relativi attributi nei token del pool di utenti.

Il diagramma seguente mostra come l'applicazione può passare il token di un utente a Verified Permissions in una richiesta di autorizzazione.



Inizia a usare Amazon Verified Permissions

Dopo aver integrato il tuo pool di utenti con Autorizzazioni verificate, ottieni una fonte centrale di autorizzazione granulare per tutte le tue app Amazon Cognito. Ciò elimina la necessità di una logica di sicurezza granulare che altrimenti dovresti codificare e replicare tra tutte le tue app. Per ulteriori informazioni sull'autorizzazione con autorizzazioni verificate, consulta. Autorizzazione con Amazon Verified Permissions

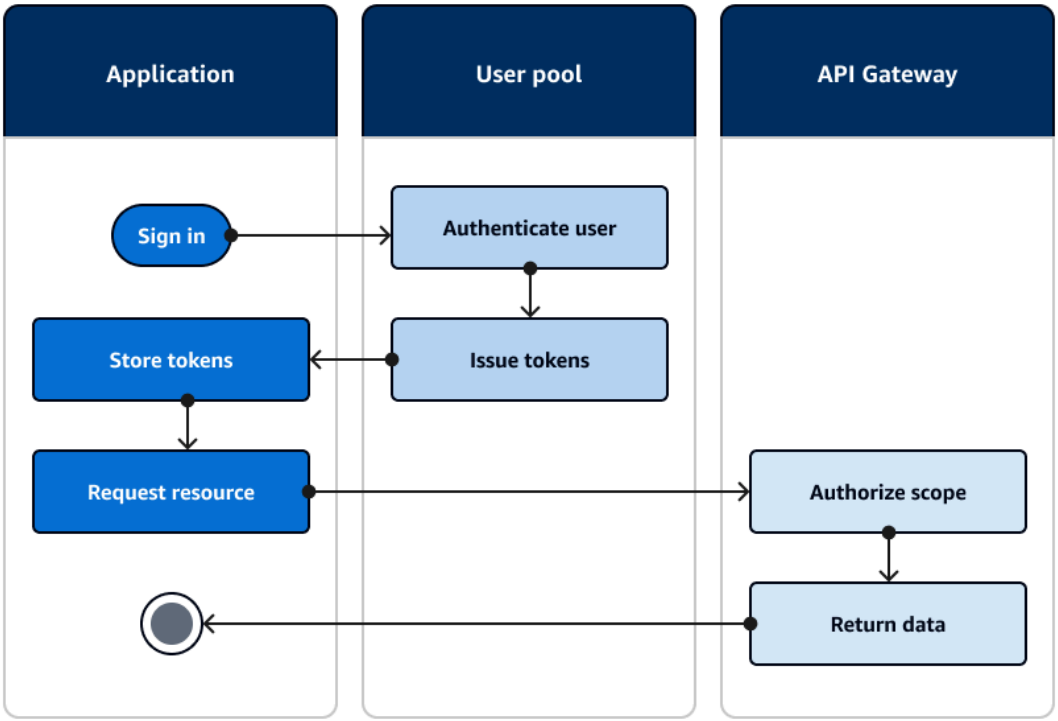
Le richieste di autorizzazione per le autorizzazioni verificate richiedono credenziali AWS . È possibile implementare alcune delle seguenti tecniche per applicare in modo sicuro le credenziali alle richieste di autorizzazione.

- Gestisci un'applicazione web in grado di archiviare segreti nel backend del server.
- Acquisisci credenziali autenticate del pool di identità.
- Invia tramite proxy le richieste degli utenti tramite un'API autorizzata dal token di accesso e aggiungi le credenziali alla richiesta. AWS

Accesso alle risorse con API Gateway dopo l'accesso

Un uso comune dei token dei pool di utenti di Amazon Cognito consiste nell'autorizzare le richieste a un'API REST di API Gateway. Gli ambiti OAuth 2.0 nei token di accesso possono autorizzare un metodo e un percorso, come for. `HTTP GET /app_assets` I token ID possono fungere da autenticazione generica per un'API e passare gli attributi utente al servizio di back-end. API Gateway offre opzioni di autorizzazione personalizzate aggiuntive come gli autorizzatori JWT per le API HTTP e gli autorizzatori Lambda che possono applicare una logica più dettagliata.

Il diagramma seguente illustra un'applicazione che sta ottenendo l'accesso a un'API REST con gli ambiti OAuth 2.0 in un token di accesso.



L'app deve raccogliere i token dalle sessioni autenticate e aggiungerli come token portatori a un'intestazione della richiesta. `Authorization` Configura l'autorizzatore che hai configurato per l'API, il percorso e il metodo per valutare il contenuto dei token. API Gateway restituisce i dati solo se la richiesta soddisfa le condizioni impostate per l'autorizzatore.

Alcuni modi potenziali in cui l'API API Gateway può approvare l'accesso da un'applicazione sono:

- Il token di accesso è valido, non è scaduto e contiene l'ambito OAuth 2.0 corretto. L'autorizzazione dei pool di utenti di Amazon Cognito per un'API REST è un'implementazione comune con una

bassa barriera all'ingresso. Puoi anche valutare il corpo, i parametri della stringa di query e le intestazioni di una richiesta a questo tipo di autorizzazione.

- Il token ID è valido e non è scaduto. Quando passi un token ID a un autorizzatore Amazon Cognito, puoi eseguire un'ulteriore convalida del contenuto del token ID sul tuo server delle applicazioni.
- Un gruppo, una dichiarazione, un attributo o un ruolo in un token di accesso o ID soddisfa i requisiti definiti in una funzione Lambda. Un autorizzatore Lambda analizza il token nell'intestazione della richiesta e lo valuta per una decisione di autorizzazione. Puoi creare una logica personalizzata nella tua funzione o effettuare una richiesta API ad Amazon Verified Permissions.

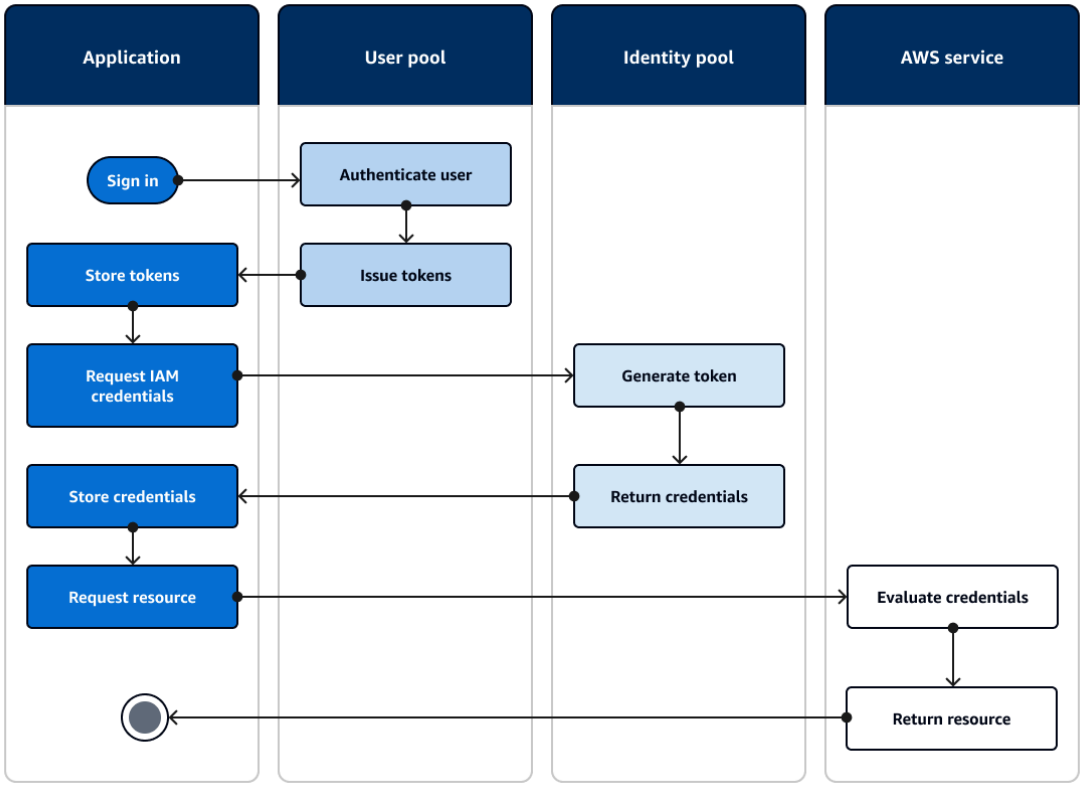
Puoi anche autorizzare le richieste a un'API AWS AppSync GraphQL con token provenienti da un pool di utenti.

Accesso Servizi AWS utilizzo di un pool di identità dopo l'accesso

Dopo aver effettuato l'accesso con un pool di utenti, gli utenti possono accedere Servizi AWS con credenziali API temporanee emesse da un pool di identità.

La tua app web o mobile riceve token da un pool di utenti. Quando configuri il tuo pool di utenti come provider di identità per il tuo pool di identità, il pool di identità scambia i token con credenziali temporanee AWS . Queste credenziali possono essere applicate ai ruoli IAM e alle relative politiche che consentono agli utenti di accedere a un set limitato di risorse. AWS Per ulteriori informazioni, consulta Flusso di autenticazione dei pool di identità.

Il diagramma seguente mostra come un'applicazione accede a un pool di utenti, recupera le credenziali del pool di identità e richiede una risorsa da un. Servizio AWS



Puoi utilizzare le credenziali del pool di identità per:

- Effettua richieste di autorizzazione dettagliate ad Amazon Verified Permissions con le credenziali del tuo utente.
- Connettiti a un'API REST di Amazon API Gateway o a un'API AWS AppSync GraphQL che autorizza le connessioni con IAM.
- Connettiti a un back-end di database come Amazon DynamoDB o Amazon RDS che autorizza le connessioni con IAM.
- Recupera gli asset applicativi da un bucket Amazon S3.
- Avvia una sessione con un desktop WorkSpaces virtuale Amazon.

I pool di identità non funzionano esclusivamente all'interno di una sessione autenticata con un pool di utenti. Accettano inoltre l'autenticazione direttamente da provider di identità di terze parti e possono generare credenziali per utenti ospiti non autenticati.

Per ulteriori informazioni sull'utilizzo dei pool di identità insieme ai gruppi di pool di utenti per controllare l'accesso alle AWS risorse, consulta e. Aggiunta di gruppi a un bacino d'utenza Utilizzo

del controllo degli accessi basato su ruoli Inoltre, per ulteriori informazioni sui pool di identità e AWS Identity and Access Management, vedereFlusso di autenticazione dei pool di identità.

Configurazione di un pool di utenti con Console di gestione AWS

Crea un bacino d'utenza di Amazon Cognito e prendi nota dell'ID bacino d'utenza e dell'ID client app per ciascuna delle app client. Per ulteriori informazioni sulla creazione di bacini d'utenza, consulta l'articolo Crea una nuova applicazione nella console Amazon Cognito.

Configurazione di un pool di identità con Console di gestione AWS

La procedura seguente descrive come utilizzare per Console di gestione AWS integrare un pool di identità con uno o più pool di utenti e app client.

Per aggiungere un gestore dell'identità digitale al pool di utenti Amazon Cognito

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Pool di utenti Amazon Cognito.
5. Inserisci un ID del pool di utenti e un ID client dell'app.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

    a. Puoi assegnare agli utenti di quell'IdP il ruolo predefinito che hai impostato quando hai configurato il ruolo Autenticato oppure puoi scegliere il ruolo con regole. Con un IdP del pool di utenti Amazon Cognito, puoi anche scegliere un ruolo con attestazione preferred_role nei token. Per ulteriori informazioni sulla richiesta `cognito:preferred_role`, consultare Assegnazione dei valori di priorità ai gruppi.

        i. Se hai scelto Scegli ruolo con regole, inserisci il claim di origine tratto dall'autenticazione dell'utente, l'operatore che desideri utilizzare per confrontare l'affermazione con la regola, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

        ii. Se hai scelto Choose role with preferred_role claim in token, Amazon Cognito emette le credenziali per il ruolo nel claim dell'utente. `cognito:preferred_role` Se non è

presente alcuna dichiarazione di ruolo preferito, Amazon Cognito emette le credenziali in base alla risoluzione del ruolo.

b. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   - Per non applicare alcun tag principale, scegli Inattivo.
   - Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.
   - Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Integrazione di un bacino d'utenza con un pool di identità

Dopo l'autenticazione dell'utente dell'app, aggiungi il relativo token di identità alla mappa degli accessi nel fornitore di credenziali. Il nome del provider dipende dall'ID del pool di utenti di Amazon Cognito. Ha la struttura seguente:

```
cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>
```

Puoi ricavare il valore per *<region>* dall'ID del pool di utenti. Ad esempio, se l'ID del pool di utenti è`us-east-1_EXAMPLE1`, allora *<region>* è`us-east-1`. Se l'ID del pool di utenti è`us-west-2_EXAMPLE2`, allora *<region>* è`us-west-2`.

JavaScript

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";

var cognitoUser = userPool.getCurrentUser();

if (cognitoUser != null) {
 cognitoUser.getSession(function(err, result) {
  if (result) {
   console.log('You are now logged in.');
```

```
   // Create a credentials provider with the user's ID token in the logins map.
   const region = '<region>';
   const credentials = fromCognitoIdentityPool({
    identityPoolId: 'YOUR_IDENTITY_POOL_ID',
    logins: {
     'cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>':
 result.getIdToken().getJwtToken()
    },
    clientConfig: { region }
   });

   // Attach the provider to each service client that needs credentials.
   const client = new SomeServiceClient({ region, credentials });
  }
 });
}
```

Android

```
cognitoUser.getSessionInBackground(new AuthenticationHandler() {
 @Override
 public void onSuccess(CognitoUserSession session) {
  String idToken = session.getIdToken().getJWTToken();

  Map<String, String> logins = new HashMap<String, String>();
  logins.put("cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>",
 session.getIdToken().getJWTToken());
  credentialsProvider.setLogins(logins);
 }

});
```

iOS - objective-C

```
AWSServiceConfiguration *serviceConfiguration = [[AWSServiceConfiguration alloc]
 initWithRegion:AWSRegionUSEast1 credentialsProvider:nil];
AWSCognitoIdentityUserPoolConfiguration *userPoolConfiguration =
 [[AWSCognitoIdentityUserPoolConfiguration alloc] initWithClientId:@"YOUR_CLIENT_ID"
  clientSecret:@"YOUR_CLIENT_SECRET" poolId:@"YOUR_USER_POOL_ID"];
[AWSCognitoIdentityUserPool
 registerCognitoIdentityUserPoolWithConfiguration:serviceConfiguration
 userPoolConfiguration:userPoolConfiguration forKey:@"UserPool"];
```

```
AWSCognitoIdentityUserPool *pool = [AWSCognitoIdentityUserPool
 CognitoIdentityUserPoolForKey:@"UserPool"];
AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider
 alloc] initWithRegionType:AWSRegionUSEast1 identityPoolId:@"YOUR_IDENTITY_POOL_ID"
 identityProviderManager:pool];
```

iOS - swift

```
let serviceConfiguration = AWSServiceConfiguration(region: .USEast1,
 credentialsProvider: nil)
let userPoolConfiguration = AWSCognitoIdentityUserPoolConfiguration(clientId:
 "YOUR_CLIENT_ID", clientSecret: "YOUR_CLIENT_SECRET", poolId: "YOUR_USER_POOL_ID")
AWSCognitoIdentityUserPool.registerCognitoIdentityUserPoolWithConfiguration(serviceConfigura
 userPoolConfiguration: userPoolConfiguration, forKey: "UserPool")
let pool = AWSCognitoIdentityUserPool(forKey: "UserPool")
let credentialsProvider = AWSCognitoCredentialsProvider(regionType: .USEast1,
 identityPoolId: "YOUR_IDENTITY_POOL_ID", identityProviderManager:pool)
```

Comprensione dei token web JSON (JWT) del pool di utenti

I token sono artefatti di autenticazione che le applicazioni possono utilizzare come prova dell'autenticazione OIDC e per richiedere l'accesso alle risorse. I token includono le attestazioni che sono informazioni sull'utente. Il token ID contiene le attestazioni relative all'identità, come il nome, il cognome e l'indirizzo e-mail dell'utente. Il token di accesso contiene le attestazioni, ad esempio `scope`, che l'utente autenticato può utilizzare per accedere alle API di terze parti, alle operazioni API self-service degli utenti di Amazon Cognito e all'Endpoint UserInfo. I token ID e di accesso includono entrambi un'attestazione `cognito:groups` che contiene l'appartenenza al gruppo dell'utente nel pool di utenti. Per ulteriori informazioni sui pool di utenti, consulta Aggiunta di gruppi a un bacino d'utenza.

Amazon Cognito dispone anche di token di aggiornamento che puoi usare per ottenere nuovi token o revocare i token esistenti. Aggiorna un token per recuperare nuovi ID e token di accesso. Revoca un token per revocare l'accesso utente consentito dai token di aggiornamento.

Amazon Cognito emette token come stringhe con codifica base64url. Puoi decodificare qualsiasi ID Amazon Cognito o token di accesso `base64url` da un codice JSON in testo semplice. I token di aggiornamento di Amazon Cognito sono crittografati, opachi agli utenti e agli amministratori dei pool di utenti e possono essere letti solo dal pool di utenti.

Autenticazione con i token

Quando un utente accede alla tua applicazione, Amazon Cognito verifica le informazioni di accesso. Se l'accesso avviene correttamente, Amazon Cognito crea una sessione e restituisce un token ID, un token di accesso e un token di aggiornamento per l'utente autenticato. Puoi utilizzare i token per concedere agli utenti l'accesso a risorse a valle e API come Gateway Amazon API. Oppure, puoi scambiarli con credenziali AWS temporanee per accedere ad altri Servizi AWS.



Archiviazione dei token

La tua app deve essere in grado di memorizzare token di varie dimensioni. La dimensione del token può cambiare per motivi tra cui, a titolo esemplificativo, altre attestazioni, modifiche negli algoritmi

di codifica e modifiche negli algoritmi di crittografia. Quando abiliti la revoca dei token nel tuo bacino d'utenza, Amazon Cognito aggiunge ulteriori attestazioni ai token Web JSON, aumentandone le dimensioni. Le nuove attestazioni `origin_jti` e `jti` vengono aggiunte ai token di accesso e ID. Per ulteriori informazioni sulla revoca dei token, consulta Revoca dei token.

> **Important**
>
> Come best practice, proteggere tutti i token durante il transito e lo storage nel contesto dell'applicazione. I token possono contenere informazioni di identificazione personali sugli utenti e informazioni sul modello di sicurezza utilizzato per il pool di utenti.

Personalizzazione dei token

Puoi personalizzare i token di accesso e ID trasferiti da Amazon Cognito alla tua app. In un Trigger Lambda di pre-generazione del token, puoi aggiungere, modificare e sopprimere le richieste relative ai token. Il trigger di pre-generazione di token è una funzione Lambda a cui Amazon Cognito invia un set predefinito di richieste. Le richieste includono ambiti OAuth 2.0, appartenenza a gruppi di pool di utenti, attributi utente e altro. La funzione può quindi cogliere l'occasione per apportare modifiche al runtime e restituire richieste di token aggiornate ad Amazon Cognito.

Costi aggiuntivi vengono applicati alla personalizzazione dei token di accesso con gli eventi della versione 2. Per ulteriori informazioni, consultare Prezzi di Amazon Cognito.

Argomenti

- Comprendere il token di identità (ID)
- Comprendere il token di accesso
- Analizza i token del pool di utenti in modo sicuro
- Aggiorna i token
- Fine delle sessioni utente con revoca del token
- Verifica dei token web JSON
- Gestione della scadenza e della memorizzazione nella cache dei token del pool di utenti
- Ambiti, M2M e server di risorse

Comprendere il token di identità (ID)

Il token ID è un JSON Web Token (JWT) che contiene attestazioni relative all'identità dell'utente autenticato, ad esempio `name`, `email` e `phone_number`. Puoi utilizzare queste informazioni sull'identità all'interno dell'applicazione. Il token ID può essere utilizzato anche per autenticare utenti in relazione al server di risorse o alle applicazioni server. È inoltre possibile utilizzare un token ID al di fuori dell'applicazione con le operazioni dell'API Web. In questi casi, prima di poter considerare attendibile qualsiasi attestazione all'interno del token ID, devi verificare la firma del token ID. Consulta Verifica dei token web JSON.

Puoi configurare il periodo di scadenza del token ID su qualsiasi valore compreso tra 5 minuti e 1 giorno. Puoi configurare questo valore per il client dell'app.

Il token ID include le `amr` attestazioni `acr` and, che descrivono il livello di autenticazione soddisfatto dall'utente e i metodi che ha completato. Per ulteriori informazioni, consulta Dichiarazioni relative ai token ACR e AMR.

> **Important**
>
> Quando l'utente accede con accesso gestito, Amazon Cognito imposta cookie di sessione validi per 1 ora. Se utilizzi l'accesso gestito per l'autenticazione nella tua applicazione e specifichi una durata minima inferiore a 1 ora per i token di accesso e ID, i tuoi utenti avranno comunque una sessione valida fino alla scadenza del cookie. Se l'utente dispone di token che scadono durante la sessione di un'ora, può aggiornare i token senza dover eseguire nuovamente l'autenticazione.

Intestazione del token ID

L'intestazione contiene due informazioni: l'ID della chiave (`kid`) e l'algoritmo (`alg`).

```
{
"kid" : "1234example=",
"alg" : "RS256"
}
```

`kid`

L'ID della chiave . Il valore indica quale chiave è utilizzata per proteggere la firma JWS (JSON Web Signature) del token. È possibile visualizzare gli ID delle chiavi di firma del pool di utenti nell'endpoint `jwks_uri`.

Per ulteriori informazioni sul `kid` parametro, consulta la sezione parametro di intestazione dell'ID della chiave (kid).

`alg`

L'algoritmo di crittografia utilizzato da Amazon Cognito per proteggere il token di accesso. I pool di utenti utilizzano un algoritmo crittografico RS256, che è una firma RSA con. SHA-256

Per ulteriori informazioni sul `alg` parametro, consultare la sezione relativa al parametro di intestazione dell'algoritmo (alg).

Payload predefinito del token ID

Questo è un esempio di payload derivante da un token ID. Contiene attestazioni relative all'utente autenticato. Per ulteriori informazioni sulle attestazioni standard OpenID Connect (OIDC), consulta l'elenco delle attestazioni standard OIDC. Puoi aggiungere rivendicazioni di tua progettazione con un. Trigger Lambda di pre-generazione del token

Questo carico utile è rappresentativo, non esaustivo. L'insieme di attestazioni in un token ID Amazon Cognito cresce nel tempo man mano che nuove funzionalità aggiungono nuove attestazioni, quindi le attestazioni associate ai token possono differire da questo esempio. Considera questo esempio come un'istantanea delle affermazioni comuni anziché come un elenco completo e fisso e leggilo Come analizzare questo token in modo sicuro prima di scrivere codice che analizzi i token di Amazon Cognito.

```
<header>.{
    "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "cognito:groups": [
        "test-group-a",
        "test-group-b",
        "test-group-c"
    ],
    "email_verified": true,
    "cognito:preferred_role": "arn:aws:iam::111122223333:role/my-test-role",
    "iss": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_example",
```

```
    "cognito:username": "my-test-user",
    "middle_name": "Jane",
    "nonce": "abcdefg",
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "cognito:roles": [
        "arn:aws:iam::111122223333:role/my-test-role"
    ],
    "aud": "xxxxxxxxxxxxexample",
    "identities": [
        {
            "userId": "amzn1.account.EXAMPLE",
            "providerName": "LoginWithAmazon",
            "providerType": "LoginWithAmazon",
            "issuer": null,
            "primary": "true",
            "dateCreated": "1642699117273"
        }
    ],
    "event_id": "64f513be-32db-42b0-b78e-b02127b4f463",
    "token_use": "id",
    "auth_time": 1676312777,
    "exp": 1676316377,
    "iat": 1676312777,
    "jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "email": "my-test-user@example.com"
}
.<token signature>
```

sub

Un identificatore univoco (UUID), o oggetto, per l'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.

> **Important**
>
> Amazon Cognito genera `sub` in un Cognito-specific formato Amazon non conforme a un formato UUID specifico, incluso l'UUID RFC. Non dovresti convalidare rigorosamente il formato di. `sub` Più in generale, come descritto nella guida sull'analisi sicura delle attestazioni relative ai token, leggi ogni attestazione come quella documentata in questa pagina e non imponete un formato più rigoroso di quello indicato nella documentazione.

`cognito:groups`

Una matrice di stringhe. Ogni stringa è il nome di un gruppo di utenti di cui l'utente è membro. I gruppi possono essere un identificatore da presentare all'app oppure possono generare una richiesta per un ruolo IAM preferito da un pool di identità. Questa affermazione è sempre una matrice, anche quando l'utente appartiene a un singolo gruppo.

`cognito:preferred_role`

L'ARN del ruolo IAM associato al gruppo dell'utente con la priorità più alta nel pool di utenti. Per ulteriori informazioni su come il pool di utenti seleziona questa attestazione ruolo, consulta Assegnazione dei valori di priorità ai gruppi.

`iss`

Una singola stringa. L'emittente del token. Questa affermazione identifica il pool di utenti che ha generato il token. L'applicazione deve verificare che questo valore corrisponda all'URL emittente previsto del pool di utenti. L'attestazione ha il seguente formato.

`https://cognito-idp.<Region>.amazonaws.com/<your user pool ID>`

Il tuo pool di utenti può utilizzare un emittente originale o aggiornato. Gli emittenti aggiornati ospitano gli stessi contenuti JWKS in più regioni, con conseguente miglioramento della resilienza e dell'efficienza. Per ulteriori informazioni, consulta Pool di utenti di Amazon Cognito come emittente OIDC.

`cognito:username`

Il nome utente nel pool di utenti.

`nonce`

L'attestazione `nonce` proviene da un parametro con lo stesso nome che puoi aggiungere alle richieste al tuo endpoint OAuth 2.0 `authorize`. Quando si aggiunge il parametro, l'attestazione `nonce` è inclusa nel token ID emesso da Amazon Cognito e puoi usarla per proteggerti dagli attacchi di tipo replay. Se non fornisci un valore `nonce` nella tua richiesta, Amazon Cognito genera e convalida automaticamente un nonce quando esegui l'autenticazione tramite un provider di identità di terze parti, quindi lo aggiunge come attestazione `nonce` al token ID. L'implementazione dell'attestazione `nonce` in Amazon Cognito si basa su standard OIDC.

`origin_jti`

Un identificatore di revoca del token associato al token di aggiornamento dell'utente. Amazon Cognito fa riferimento all'`origin_jti`affermazione quando verifica se hai revocato il token

dell'utente con l'operazione Endpoint Revoke o l'RevokeTokenAPI. Quando revochi un token, Amazon Cognito invalida tutti i token di accesso e ID con lo stesso valore `origin_jti`.

`cognito:roles`

Una matrice di stringhe. Ogni stringa è l'ARN di un ruolo IAM associato a uno dei gruppi dell'utente. A ogni gruppo di pool di utenti può essere associato un ruolo IAM. Questo array rappresenta tutti i ruoli IAM per i gruppi degli utenti, indipendentemente dalla precedenza, ed è sempre un array anche quando contiene un singolo ruolo. Per ulteriori informazioni, consulta Aggiunta di gruppi a un bacino d'utenza.

`aud`

Una singola stringa. 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.

`identities`

Un array di oggetti . Ogni oggetto descrive un profilo di provider di identità di terze parti che hai collegato all'utente, tramite accesso federato o collegando un utente federato a un profilo locale. Ogni oggetto contiene i campi `userId` (l'ID univoco dell'utente presso il provider),, `providerNameproviderType`, `issuer` e. `primary` `dateCreated` Tutti questi campi sono stringhe, con due dettagli di serializzazione da analizzare con attenzione: `primary` è la stringa `"true"` o non è un booleano JSON`"false"`, ed `dateCreated` è una stringa che contiene il tempo Unix in millisecondi (ad esempio), non un numero. `"1642699117273"` Il campo `null` può essere. `issuer` Questa affermazione è sempre una matrice, anche quando l'utente ha un'identità collegata.

`token_use`

Lo scopo previsto del token. In un token ID, il valore è `id`.

`auth_time`

Un numero . L'ora di autenticazione, nel formato orario Unix, quando l'utente ha completato l'autenticazione.

`exp`

Un numero . L'ora di scadenza, in formato orario Unix, alla scadenza del token dell'utente.

`iat`

Un numero . L'ora emessa, in formato orario Unix, quando Amazon Cognito ha emesso il token dell'utente.

`jti`

L'identificatore univoco del JWT.

`email`

Una singola stringa. L'indirizzo e-mail dell'utente, quando l'`email`attributo è presente e leggibile dal client dell'app.

Il token ID può contenere le attestazioni standard OIDC definite nelle attestazioni standard OIDC. Il token ID può contenere inoltre gli attributi personalizzati definiti nel bacino d'utenza. Amazon Cognito scrive i valori degli attributi personalizzati nel token ID come stringhe indipendentemente dal tipo di attributo.

> **Note**
>
> Gli attributi personalizzati del pool di utenti hanno sempre il prefisso. `custom:`

Come analizzare questo token in modo sicuro

L'affermazione impostata in un token ID Amazon Cognito non è stata corretta. Le affermazioni supportate da un token variano a seconda del flusso di autenticazione, dell'appartenenza al gruppo dell'utente e del provider di identità. Dipendono anche dalle funzionalità abilitate nel pool di utenti. La specifica OpenID Connect richiede solo`iss`,`sub`, e `aud` `expiat`, e questi sono presenti in ogni token ID Amazon Cognito. Inoltre, Amazon `token_use` Cognito aggiunge sempre il valore`id`, che dovresti verificare per confermare di aver ricevuto un token ID anziché un token di accesso. Considera ogni altra affermazione come facoltativa e subordinata al modo in cui l'utente si è autenticato.

Il token ID è uno dei diversi tipi di token di Amazon Cognito e le stesse regole di analisi si applicano a tutti. Per ulteriori informazioni sull'analisi sicura dei token, consulta. Analizza i token del pool di utenti in modo sicuro

Esempi di token ID per scenario

Payload predefinito del token IDCombina molte affermazioni opzionali in un unico esempio. In pratica, le affermazioni supportate da un token dipendono dal modo in cui l'utente ha effettuato l'accesso. Gli esempi seguenti mostrano le combinazioni di claim emesse da Amazon Cognito per quattro scenari comuni. Ogni esempio è un payload decodificato, non un JWT firmato e serializzato: un token reale è

`header.payload.signature` codificato in base64url e con la chiave del tuo pool di utenti. RS256-signed I valori`sub`, ID del pool di utenti e timestamp sono segnaposto illustrativi.

Token ID del codice di autorizzazione di base

Il token di identità utilizzato dalla maggior parte delle applicazioni dopo lo scambio di un codice di autorizzazione sull'endpoint del token. Il `token_use` valore di lo `id` distingue da un token di accesso. L'`nonce`affermazione è presente perché il client ha inviato un `nonce` parametro nella richiesta di autorizzazione per aiutare a prevenire attacchi replay.

```
{
    "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
    "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
    "aud": "6o7h8i9jexampleclientid",
    "cognito:username": "my-test-user",
    "email": "my-test-user@example.com",
    "email_verified": true,
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "token_use": "id",
    "auth_time": 1759345200,
    "iat": 1759345200,
    "exp": 1759348800,
    "jti": "11112222-3333-4444-5555-666677778888",
    "nonce": "n-0S6_WzA2Mj"
}
```

Appartenenza a gruppi e ruoli

Amazon Cognito aggiunge le richieste di autorizzazione per gruppi e ruoli quando l'utente appartiene a uno o più gruppi di pool di utenti. Le `cognito:roles` attestazioni `cognito:groups` e sono matrici; leggile come array anche quando contengono un singolo elemento. `cognito:preferred_role`è una stringa singola.

```
{
    "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
    "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
    "aud": "6o7h8i9jexampleclientid",
    "cognito:username": "my-test-user",
    "cognito:groups": ["admins", "beta-testers"],
    "cognito:roles": ["arn:aws:iam::111122223333:role/my-test-role"],
    "cognito:preferred_role": "arn:aws:iam::111122223333:role/my-test-role",
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
```

```
    "token_use": "id",
    "auth_time": 1759345200,
    "iat": 1759345200,
    "exp": 1759348800,
    "jti": "11112222-3333-4444-5555-666677778888"
}
```

Accesso federato

Un utente che ha effettuato l'accesso tramite un provider di identità esterno, ad esempio un social network o un SAML/OIDC provider, presenta il `identities` reclamo, che descrive il link del provider esterno.

```
{
    "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
    "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
    "aud": "6o7h8i9jexampleclientid",
    "cognito:username": "LoginWithAmazon_amzn1.account.EXAMPLE",
    "identities": [
        {
            "userId": "amzn1.account.EXAMPLE",
            "providerName": "LoginWithAmazon",
            "providerType": "LoginWithAmazon",
            "issuer": null,
            "primary": "true",
            "dateCreated": "1642699117273"
        }
    ],
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "token_use": "id",
    "auth_time": 1759345200,
    "iat": 1759345200,
    "exp": 1759348800,
    "jti": "11112222-3333-4444-5555-666677778888"
}
```

> **Note**
>
> All'interno di ogni `identities` oggetto, `primary` `dateCreated` sono serializzati come stringhe (`"true"`e`"1642699117273"`), non come valori booleani e numeri JSON. Un confronto come ad esempio restituisce silenziosamente false`identity.primary ===`

`true`. Leggi `primary` come stringa `"true"` o `"false"` e `dateCreated` come stringa che contiene il tempo Unix in millisecondi.

Forza di autenticazione (ACR e AMR)

Quando il tuo pool di utenti segnala il livello di autenticazione raggiunto dall'utente, Amazon Cognito aggiunge `acr` e `amr` dichiara. Il token seguente proviene da un utente che si è autenticato con una password e un TOTP da un'app di autenticazione, che è il livello più alto. Ogni token contiene esattamente uno dei livelli possibili. `acr` Per l'insieme fisso di livelli, le combinazioni di fattori che soddisfano ciascuno di essi e l'elenco completo dei `amr` valori, vedereLivelli di autenticazione con attestazioni ACR e AMR.

```
{
    "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
    "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
    "aud": "6o7h8i9jexampleclientid",
    "cognito:username": "my-test-user",
    "acr": "urn:cognito:loa:4",
    "amr": ["pwd", "otp", "mfa"],
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "token_use": "id",
    "auth_time": 1759345200,
    "iat": 1759345200,
    "exp": 1759348800,
    "jti": "11112222-3333-4444-5555-666677778888"
}
```

La richiesta di un livello di autenticazione richiede il piano di funzionalità Essentials o Plus. Le `amr` affermazioni `acr` e vengono calcolate da Amazon Cognito e non possono essere modificate da un trigger Lambda precedente alla generazione di token. La traduzione dei livelli di Amazon Cognito in un altro standard, come NIST o eIDAS, è responsabilità dell'applicazione.

Firma con token ID

La firma del token ID viene calcolata in base all'intestazione e al payload del token JWT. Verifica la firma del token prima di accettare le attestazioni in qualsiasi token ID ricevuto dall'app. Per ulteriori informazioni, consulta Verifica di un token web JSON. Verifica dei token web JSON.

Comprendere il token di accesso

Il token di accesso al bacino d'utenza contiene attestazioni relative all'utente autenticato, un elenco dei gruppi dell'utente e un elenco di ambiti. Lo scopo del token di accesso è autorizzare le operazioni API. Il pool di utenti accetta i token di accesso per autorizzare le operazioni self-service degli utenti. Ad esempio, puoi utilizzare il token di accesso per consentire agli utenti di accedere per aggiornare, modificare o eliminare gli attributi utente.

Con gli ambiti OAuth 2.0 in un token di accesso, determinato dagli ambiti personalizzati che aggiungi al pool di utenti, puoi autorizzare l'utente a recuperare informazioni da un'API. Gateway Amazon API, ad esempio, supporta l'autorizzazione con token di accesso di Amazon Cognito. Puoi compilare un autorizzatore dell'API REST con le informazioni del tuo pool di utenti. Per i dettagli, consulta Controllare l'accesso alle API REST utilizzando i pool di utenti di Amazon Cognito come autorizzatore. Puoi anche usare Amazon Cognito come autorizzatore JWT per un'API HTTP. Per i dettagli, consulta Controllare l'accesso alle API HTTP con gli autorizzatori JWT in API Gateway. Per generare un token di accesso con ambiti personalizzati, è necessario effettuare la richiesta tramite gli endpoint pubblici del pool di utenti.

Con il piano di funzionalità Essentials o Plus, puoi anche implementare un trigger Lambda prima della generazione di token che aggiunge ambiti ai token di accesso in fase di esecuzione. Per ulteriori informazioni, consulta Trigger Lambda di pre-generazione del token.

Il token di accesso di un utente con lo `openid` scopo è l'autorizzazione a richiedere ulteriori informazioni sugli attributi dell'utente al. Endpoint UserInfo La quantità di informazioni dall'`userInfo`endpoint deriva dagli ambiti aggiuntivi del token di accesso: ad esempio, per tutti i dati dell'utente, `profile` per il relativo indirizzo `email` e-mail.

Il token di accesso di un utente con lo `aws.cognito.signin.user.admin` scopo è l'autorizzazione a leggere e scrivere gli attributi utente, elencare i fattori di autenticazione, configurare le preferenze di autenticazione a più fattori (MFA) e gestire i dispositivi ricordati. Il livello di accesso agli attributi che il token di accesso concede a questo ambito corrisponde alle read/write autorizzazioni degli attributi che assegnate al client dell'app.

Il token di accesso è un token web JSON (JWT). L'intestazione per il token di accesso ha la stessa struttura del token ID. Amazon Cognito firma i token di accesso con una chiave diversa da quella usata per la firma dei token ID. Il valore di un'attestazione ID chiave di accesso (`kid`) non corrisponde al valore dell'attestazione `kid` di un token ID nella stessa sessione utente. Verifica nel codice dell'app i token ID e di accesso in modo indipendente. Non considerare attendibili le

attestazioni contenute in un token di accesso finché non verifichi la firma. Per ulteriori informazioni, consulta Verifica dei token web JSON. È possibile configurare il periodo di scadenza del token di accesso su qualsiasi valore compreso tra 5 minuti e 1 giorno. Puoi configurare questo valore per il client dell'app.

> **Important**
>
> Per i token di accesso e ID, non specificare un periodo minimo inferiore a un'ora se utilizzi l'accesso gestito. L'accesso gestito imposta i cookie del browser validi per un'ora. Se si configura una durata del token di accesso inferiore a un'ora, ciò non ha alcun effetto sulla validità del cookie di accesso gestito e sulla capacità degli utenti di riautenticarsi senza credenziali aggiuntive per un'ora dopo l'accesso iniziale.

Il token di accesso trasmette le attestazioni in modo aggiuntivo, allo stesso modo del token ID, e si applicano le stesse regole di analisi. Per ulteriori informazioni sull'analisi sicura dei token, consulta. Analizza i token del pool di utenti in modo sicuro

Intestazione del token di accesso

L'intestazione contiene due informazioni: l'ID della chiave (`kid`) e l'algoritmo (`alg`).

```
{
"kid" : "1234example="
"alg" : "RS256",
}
```

`kid`

L'ID della chiave . Il valore indica quale chiave è utilizzata per proteggere la firma JWS (JSON Web Signature) del token. È possibile visualizzare gli ID delle chiavi di firma del pool di utenti nell'endpoint `jwks_uri`.

Per ulteriori informazioni sul `kid` parametro, consulta la sezione parametro di intestazione dell'ID della chiave (kid).

`alg`

L'algoritmo di crittografia utilizzato da Amazon Cognito per proteggere il token di accesso. I pool di utenti utilizzano un algoritmo crittografico RS256, che è una firma RSA con. SHA-256

Per ulteriori informazioni sul `alg` parametro, consultare la sezione relativa al parametro di intestazione dell'algoritmo (alg).

Payload predefinito del token di accesso

Questo è un esempio di payload di un token di accesso. Per ulteriori informazioni, consultare la sezione relativa alle attestazioni JWT. Puoi aggiungere rivendicazioni personalizzate con unTrigger Lambda di pre-generazione del token.

```
<header>.
{
   "sub":"aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "device_key": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "cognito:groups":[
      "testgroup"
   ],
   "iss":"https://cognito-idp.us-west-2.amazonaws.com/us-west-2_example",
   "version":2,
   "client_id":"xxxxxxxxxxxxexample",
   "aud": "https://api.example.com",
   "origin_jti":"aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "event_id":"aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "token_use":"access",
   "scope":"phone openid profile resourceserver.1/appclient2 email",
   "auth_time":1676313851,
   "exp":1676317451,
   "iat":1676313851,
   "jti":"aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "username":"my-test-user"
}
.<token signature>
```

sub

Un identificatore univoco (UUID), o oggetto, per l'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.

> **Important**
>
> Amazon Cognito genera `sub` in un Cognito-specific formato Amazon non conforme a un formato UUID specifico, incluso l'UUID RFC. Non dovresti convalidare rigorosamente il formato di. `sub`

`cognito:groups`

Una serie di nomi di gruppi del pool di utenti che includono l'utente come membro.

`iss`

L'emittente del token. Questa affermazione identifica il pool di utenti che ha generato il token. L'applicazione deve verificare che questo valore corrisponda all'URL emittente previsto del pool di utenti. L'attestazione ha il seguente formato.

`https://cognito-idp.`*`us-east-1`*`.amazonaws.com/`*`us-east-1_EXAMPLE`*

Il tuo pool di utenti può utilizzare un emittente originale o aggiornato. Gli emittenti aggiornati ospitano gli stessi contenuti JWKS in più regioni, con conseguente miglioramento della resilienza e dell'efficienza. Per ulteriori informazioni, consulta Pool di utenti di Amazon Cognito come emittente OIDC.

`client_id`

Il client dell'app del pool di utenti che ha autenticato l'utente. Amazon Cognito restituisce lo stesso valore nell'attestazione `aud` del token ID.

aud

L'URL dell'API che il token di accesso è destinato ad autorizzare. Presente solo se l'applicazione ha richiesto un'associazione di risorse dal server di autorizzazione.

`origin_jti`

Un identificatore di revoca del token associato al token di aggiornamento dell'utente. Amazon Cognito fa riferimento all'`origin_jti`affermazione quando verifica se hai revocato il token dell'utente con l'operazione Endpoint Revoke o l'RevokeTokenAPI. Quando revochi un token, Amazon Cognito non convalida più i token di accesso e ID con lo stesso valore. `origin_jti`

`token_use`

Lo scopo previsto del token. In un token di accesso, il valore è `access`.

`scope`

Un elenco di ambiti OAuth 2.0 rilasciati all'utente che ha effettuato l'accesso. Gli ambiti definiscono l'accesso che il token fornisce alle API esterne, alle operazioni self-service degli utenti e ai dati degli utenti sull'endpoint. `userInfo` Un token dell'Endpoint Token può contenere tutti gli ambiti supportati dal client dell'app. Un token dell'accesso API di Amazon Cognito contiene solo l'ambito `aws.cognito.signin.user.admin`.

`auth_time`

L'ora di fine dell'autenticazione dell'utente, in formato Unix.

`exp`

L'ora di scadenza del token dell'utente, in formato Unix.

`iat`

L'ora di emissione del token dell'utente da parte di Amazon Cognito, in formato Unix.

`jti`

L'identificatore univoco del JWT.

`username`

Il nome utente dell'utente nel pool di utenti.

Altre risorse

- Come personalizzare i token di accesso nei pool di utenti di Amazon Cognito

Machine-to-machine payload dei token di accesso

Poiché un token di accesso da macchina a macchina (M2M) autorizza un'applicazione anziché un utente, il suo payload è diverso dal token di accesso utente descritto in precedenza. Nell'autorizzazione M2M, l'`sub`attestazione contiene l'ID del client dell'app che ha richiesto il token, non un identificatore utente, e le attestazioni and hanno lo stesso valore. `sub` `client_id` Il token contiene gli ambiti personalizzati che autorizzano le richieste al server di risorse, insieme a attestazioni standard come`iss`, `token_use` (con un valore di`access`),, `exp` e. `iat` `jti`

Poiché l'autorizzazione M2M non coinvolge alcun utente e non emette alcun token di aggiornamento, un token di accesso M2M omette diverse affermazioni contenute nel token di accesso utente. Il tuo server di risorse non può aspettarsi queste affermazioni:

- Le `cognito:groups` affermazioni `username` and, che identificano un utente autenticato e la sua appartenenza al gruppo.
- L'`device_key`affermazione, che identifica un dispositivo utente ricordato.
- L'`origin_jti`affermazione, che collega un token a un token di aggiornamento revocabile che l'autorizzazione M2M non rilascia.

Firma del token di accesso

La firma del token di accesso, firmata con la chiave pubblicizzata sull'`.well-known/jwks.json`endpoint, convalida l'integrità dell'intestazione e del payload del token. Quando utilizzi i token di accesso per autorizzare l'accesso alle API esterne, configura sempre l'autorizzatore dell'API per verificare questa firma rispetto alla chiave che l'ha firmata. Per ulteriori informazioni, consulta Verifica dei token web JSON.

Analizza i token del pool di utenti in modo sicuro

Il set di claim di un token Amazon Cognito non è fisso. Le affermazioni supportate dal token variano a seconda del flusso di autenticazione, dell'appartenenza al gruppo dell'utente, del provider di identità e delle funzionalità abilitate nel pool di utenti. Amazon Cognito aggiunge nuove attestazioni in modo progressivo nel tempo: introduce nuove chiavi, ma non rinomina, modifica il tipo o rimuove le attestazioni esistenti documentate in questa guida. Queste regole si applicano sia ai token ID che ai token di accesso.

Non scrivere il tuo parser di token. Usa una libreria di token web JSON (JWT) e un parser JSON che continuano a funzionare man mano che il token cresce. Le due regole che impediscono quasi tutte le violazioni nel mondo reale sono le prime due seguenti: tollera le affermazioni che non riconosci e fai le giuste ipotesi sul tipo e sulla cardinalità di ogni affermazione che leggi.

Ignora le affermazioni che non riconosci

> Leggi le affermazioni di cui ha bisogno la tua candidatura e lascia perdere il resto. Non rifiutare, fallire o generare un errore quando un token riporta un'affermazione che non è elencata in questa guida. Le nuove attestazioni vengono aggiunte in modo aggiuntivo e un token che contiene un'attestazione non elencata è ancora valido.

Leggi ogni reclamo nel suo tipo documentato

Un parser JSON identifica il tipo primitivo di ogni valore dal JSON stesso, quindi il rischio sta nelle ipotesi che il codice fa su quel valore. Considera ogni affermazione come il tipo e la cardinalità che questa guida documenta. Un'affermazione documentata come matrice è sempre una matrice, anche quando attualmente contiene un singolo valore. Non comprimerlo in una stringa. Ad esempio, `cognito:groups`cognito:roles, e `identities` sono sempre matrici. Lo stesso principio si applica alle rivendicazioni definite da altre specifiche. Ad esempio`amr`, quando è presente, è presente un array di stringhe come definito nella RFC 8176 sul sito Web IETF, quindi leggilo in quel tipo.

Non convalidate eccessivamente i formati di attestazione

Convalida ciò che richiede la specifica di un reclamo e non di più. Ad esempio, `sub` è una stringa unica ma non è garantito che sia in un particolare formato UUID. Alcuni valori all'interno di dichiarazioni personalizzate, ad esempio `identities` quelli serializzati come stringhe anziché come valori booleani o numeri JSON, e non è garantito che i valori opachi abbiano una lunghezza particolare. L'imposizione di un formato più rigoroso rispetto agli stati della documentazione causa il fallimento dei token validi.

Non date per scontato il numero di reclami o il loro ordine

Le rivendicazioni in un token possono apparire in qualsiasi ordine e il conteggio può variare da un token all'altro e nel tempo. Indirizza ogni affermazione per nome anziché per posizione e analizza il token come un oggetto JSON generale anziché come una struttura fissa.

Budget per la crescita delle dimensioni dei token

Poiché il set di claim cresce, cresce anche la dimensione complessiva del token. Non presupponete una dimensione fissa o massima del token nei buffer, nella larghezza delle colonne, nelle intestazioni o nei cookie.

Il documento di `openid-configuration` scoperta segue lo stesso approccio additivo e compatibile con le versioni precedenti. Per ulteriori informazioni sull'analisi sicura del documento, consulta. Preparati per le modifiche al documento di scoperta

Aggiorna i token

Puoi utilizzare il token di aggiornamento per recuperare nuovi ID e token di accesso. Di default, il token di aggiornamento scade 30 giorni dopo l'accesso dell'utente dell'app al bacino d'utenza.

Quando crei un'applicazione per il bacino d'utenza, puoi impostare la scadenza del token di aggiornamento dell'applicazione su qualsiasi valore compreso tra 60 minuti e 10 anni.

Ottenere nuovi token di accesso e identità con un token di aggiornamento

Amazon Cognito emette token di aggiornamento in risposta a un'autenticazione riuscita con il flusso di codice di autorizzazione all'accesso gestito e con operazioni API o metodi SDK. Il token di aggiornamento restituisce nuovi ID e token di accesso e, facoltativamente, un nuovo token di aggiornamento. È possibile utilizzare i token di aggiornamento nei seguenti modi.

GetTokensFromRefreshToken

L'operazione GetTokensFromRefreshTokenAPI emette nuovi ID e token di accesso da un token di aggiornamento valido. Ottieni anche un nuovo token di aggiornamento se hai abilitato la rotazione dei token di aggiornamento.

InitiateAuth e AdminInitiateAuth

Le operazioni AdminInitiateAutho InitiateAuthAPI includono il flusso di `REFRESH_TOKEN_AUTH` autenticazione. In questo flusso, si passa un token di aggiornamento e si ottengono nuovi ID e token di accesso. Non puoi autenticarti con client interni `REFRESH_TOKEN_AUTH` all'app con rotazione dei token di aggiornamento abilitata.

Endpoint del token OAuth

L'endpoint del token nei pool di utenti con un dominio ha un tipo di `refresh_token` concessione che emette un nuovo ID, un nuovo accesso e, facoltativamente (con la rotazione del token di aggiornamento), i token di aggiornamento da un token di aggiornamento valido.

Aggiorna la rotazione dei token

Puoi facoltativamente configurare la rotazione dei token di aggiornamento nel client dell'app. Con la rotazione dei token di aggiornamento, il client può invalidare il token di aggiornamento originale ed emettere un nuovo token di aggiornamento ad ogni aggiornamento del token. Quando questa impostazione è abilitata, ogni richiesta riuscita in tutte le forme di aggiornamento del token restituisce un nuovo ID, un nuovo token di accesso e di aggiornamento. Quando questa impostazione è disabilitata, le richieste di aggiornamento dei token restituiscono solo nuovi token di accesso e ID e il token di aggiornamento originale rimane valido. Il nuovo token di aggiornamento è valido per la durata residua del token di aggiornamento originale. Puoi configurare i client dell'app in modo che

ruotino i token di aggiornamento o trasferiscano il token di aggiornamento originale. Per consentire nuovi tentativi per un breve periodo, puoi anche configurare un periodo di prova per il token di aggiornamento originale fino a 60 secondi.

> **Note**
>
> Rotation riemette un token di aggiornamento all'interno della stessa finestra di validità fissa. Non estende la finestra. Ogni nuovo token di aggiornamento è valido solo per il tempo rimanente nel `RefreshTokenValidity` periodo originale, indipendentemente dal numero di volte in cui il token ruota. Ad esempio, supponiamo di impostare su `RefreshTokenValidity` 30 giorni e che una sessione ruoti il proprio token di aggiornamento ogni giorno. La sessione scade ancora 30 giorni dopo il primo accesso, non 30 giorni dopo la rotazione più recente. Per mantenere l'accesso di un utente oltre tale finestra, deve autenticarsi nuovamente.

Cose da sapere sulla rotazione dei token di aggiornamento

- Dopo aver abilitato la rotazione dei token di aggiornamento, vengono aggiunte nuove attestazioni nei token web JSON dal tuo pool di utenti. Le attestazioni `origin_jti` e `jti` vengono aggiunte ai token di accesso e ID. Queste affermazioni aumentano le dimensioni dei JWT.
- La rotazione dei token di aggiornamento non è compatibile con il flusso di autenticazione. `REFRESH_TOKEN_AUTH` Per implementare la rotazione dei token di aggiornamento, devi disabilitare questo flusso di autenticazione nel client dell'app e progettare l'applicazione per inviare richieste di aggiornamento dei token con l'operazione GetTokensFromRefreshTokenAPI o il metodo SDK equivalente.
- Con la rotazione dei token di aggiornamento inattiva, puoi completare le richieste di aggiornamento dei token con o. `GetTokensFromRefreshToken` `REFRESH_TOKEN_AUTH`
- Quando la memorizzazione del dispositivo è attiva nel tuo pool di utenti, devi fornire la chiave del dispositivo nelle richieste. `GetTokensFromRefreshToken` Se l'utente non dispone di una chiave del dispositivo confermata che l'applicazione invia nella richiesta di autenticazione iniziale, Amazon Cognito ne emette una nuova. Per aggiornare i token in questa configurazione, devi fornire una chiave del dispositivo, indipendentemente dal fatto che tu ne abbia specificata una `AuthParameters` o ne abbia ricevuta una nuova nella risposta di autenticazione.
- Puoi passare `ClientMetadata` al trigger Lambda prima della generazione del token nella tua `GetTokensFromRefreshToken` richiesta. Questi dati, che vengono passati all'evento di input per

il trigger, forniscono un contesto aggiuntivo che puoi utilizzare nella logica personalizzata della tua funzione Lambda.

Come best practice di sicurezza, abilita la rotazione dei token di aggiornamento sui client dell'app.

Enable refresh token rotation (console)

La procedura seguente attiva o disattiva la rotazione dei token di aggiornamento per il client dell'app. Questa procedura richiede un client di app esistente. Per ulteriori informazioni sulla creazione di un client per app, consultaApplication-specific impostazioni con i client dell'app.

Per abilitare la rotazione dei token di aggiornamento

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Vai al menu App client e seleziona un client di app esistente.
5. Seleziona Modifica nella sezione Informazioni sul client dell'app della pagina.
6. In Configurazioni di sicurezza avanzate, individua l'opzione Abilita la rotazione del token di aggiornamento.
7. Per abilitare la rotazione, seleziona la casella di controllo. Per disabilitare la rotazione, deseleziona la casella di controllo.
8. In Periodo di grazia di rotazione del token di aggiornamento, inserisci il numero di secondi, fino a 60, che desideri impostare come ritardo prima della revoca del token di aggiornamento ruotato verso l'esterno.

Enable refresh token rotation (API)

Configura la rotazione del token di aggiornamento in una richiesta o API. CreateUserPoolClientUpdateUserPoolClient Il seguente corpo parziale della richiesta attiva la rotazione del token di aggiornamento e imposta il periodo di grazia su dieci secondi.

```
"RefreshTokenRotation" : {
   "Feature" : "ENABLED",
   "RetryGracePeriodSeconds" : 10
}
```

Aggiornamento dell'API e dei token SDK

Esistono due modi per utilizzare il token di aggiornamento per ottenere un nuovo ID e accedere ai token con l'API dei pool di utenti, a seconda che la rotazione dei token di aggiornamento sia attiva. Nei client dell'app con rotazione del token di aggiornamento attiva, utilizza l'operazione API. GetTokensFromRefreshToken Nei client di app senza rotazione del token di aggiornamento, utilizza il `REFRESH_TOKEN_AUTH` flusso delle operazioni AdminInitiateAutho dell'InitiateAuthAPI.

> **Note**
>
> Gli utenti possono autenticarsi con pool di utenti nell'accesso gestito o nelle applicazioni personalizzate create con AWS SDK e operazioni API di Amazon Cognito. Il `REFRESH_TOKEN_AUTH` flusso e `GetTokensFromRefreshToken` possono entrambi completare l'aggiornamento del token per gli utenti con accesso gestito. L'aggiornamento dei token nelle applicazioni personalizzate non influisce sulle sessioni di accesso gestite. Queste sessioni sono impostate in un cookie del browser e sono valide per un'ora. La `GetTokensFromRefreshToken` risposta emette nuovi ID, accessi e, facoltativamente, token di aggiornamento, ma non rinnova il cookie della sessione di accesso gestita. `REFRESH_TOKEN_AUTH`non è disponibile nei client di app con la rotazione dei token di aggiornamento abilitata.

GetTokensFromRefreshToken

GetTokensFromRefreshTokenrestituisce nuovi ID, token di accesso e aggiornamento da una richiesta autorizzata con un token di aggiornamento. Di seguito è riportato un esempio di corpo della richiesta per. `GetTokensFromRefreshToken` Puoi inviare i metadati del client ai trigger Lambda nelle richieste di questa operazione.

```
{
    "RefreshToken": "eyJjd123abcEXAMPLE",
    "ClientId": "1example23456789",
    "ClientSecret": "myappclientsecret123abc",
    "ClientMetadata": {
      "MyMetadataKey" : "MyMetadataValue"
   },
}
```

AdminInitiateAuth/InitiateAuth

Per utilizzare il token di aggiornamento quando la rotazione del token di aggiornamento è inattiva, utilizza le operazioni o API. AdminInitiateAuthInitiateAuth Invia `REFRESH_TOKEN_AUTH` per il parametro `AuthFlow`. Nella proprietà `AuthParameters` di `AuthFlow`, passa il token di aggiornamento dell'utente come valore di `"REFRESH_TOKEN"`. Amazon Cognito restituisce ID e token di accesso nuovi dopo che la richiesta API ha superato tutte le sfide.

Di seguito è riportato un esempio di corpo di richiesta per un aggiornamento del token con l'API or. `InitiateAuth` `AdminInitiateAuth`

```
{
    "AuthFlow": "REFRESH_TOKEN_AUTH",
    "ClientId": "1example23456789",
    "UserPoolId": "us-west-2_EXAMPLE",
    "AuthParameters": {
        "REFRESH_TOKEN": "eyJjd123abcEXAMPLE",
        "SECRET_HASH": "kT5acwCVrbD6JexhW3EQwnRSe6fLuPTRkEQ50athqv8="
    }
}
```

Aggiornamento del token OAuth

Puoi anche inviare token di aggiornamento a Endpoint Token in un pool di utenti in cui hai configurato un dominio. Nel corpo della richiesta, includi un valore `grant_type` di `refresh_token` e un valore `refresh_token` del token di aggiornamento dell'utente.

Le richieste all'endpoint del token sono disponibili nei client di app con rotazione dei token di aggiornamento attiva e in quelli in cui è inattiva. Quando la rotazione del token di aggiornamento è attiva, l'endpoint del token restituisce un nuovo token di aggiornamento.

Di seguito è riportato un esempio di richiesta con un token di aggiornamento.

```
POST /oauth2/token HTTP/1.1
Host: auth.example.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw
Content-Length: **

client_id=1example23456789&grant_type=refresh_token&refresh_token=eyJjd123abcEXAMPLE
```

Revoca dei token di aggiornamento

È possibile revocare i token di aggiornamento che appartengono a un utente. Per ulteriori informazioni sulla revoca dei token, consulta Fine delle sessioni utente con revoca del token.

> **Note**
>
> La revoca del token di aggiornamento revoca tutti gli ID e i token di accesso emessi da Amazon Cognito in base alle richieste di aggiornamento con quel token.

Per disconnettere gli utenti da tutte le sessioni di accesso correnti, revoca tutti i token o le richieste API. GlobalSignOutAdminUserGlobalSignOut Dopo che l'utente viene disconnesso, si verifica quanto segue.

- Il token di aggiornamento dell'utente non può ottenere nuovi token per l'utente.
- Il token di accesso dell'utente non può effettuare richieste API autorizzate dal token.
- L'utente deve effettuare nuovamente l'autenticazione per ottenere nuovi token. Poiché i cookie della sessione di accesso gestita non scadono automaticamente, l'utente può riautenticarsi con un cookie di sessione, senza richiedere ulteriori credenziali. Dopo aver disconnesso gli utenti con accesso gestito, reindirizzali alEndpoint Logout, dove Amazon Cognito cancella il cookie di sessione.

Con i token di aggiornamento, puoi mantenere le sessioni degli utenti nella tua app a lungo. Nel tempo, i tuoi utenti potrebbero voler annullare l'autorizzazione di alcune applicazioni a cui sono rimasti connessi con i loro token di aggiornamento. Per disconnettere un utente da una singola sessione, revoca il token di aggiornamento. Quando l'utente desidera disconnettersi da tutte le sessioni autenticate, genera una richiesta API. GlobalSignOut La tua app può offrire all'utente una scelta come Esci da tutti i dispositivi. `GlobalSignOut`accetta il token di accesso valido, inalterato, non scaduto e non revocato di un utente. Poiché questa API è autorizzata dal token, un utente non può utilizzarla per avviare la disconnessione di un altro utente.

Tuttavia, puoi generare una richiesta AdminUserGlobalSignOutAPI che autorizzi con le tue credenziali per disconnettere qualsiasi utente da tutti i suoi dispositivi. AWS L'applicazione di amministrazione deve richiamare questa operazione API con le credenziali AWS dello sviluppatore e passare l'ID del pool di utenti e il nome utente dell'utente come parametri. L'API `AdminUserGlobalSignOut` è in grado di disconnettere qualsiasi utente nel bacino d'utenza.

Per ulteriori informazioni sulle richieste che è possibile autorizzare con AWS credenziali o con il token di accesso di un utente, vedere. Elenco delle operazioni API raggruppate per modello di autorizzazione

Fine delle sessioni utente con revoca del token

È possibile revocare i token di aggiornamento e le sessioni degli utenti finali con i seguenti metodi. Quando si revoca un token di aggiornamento, tutti i token di accesso precedentemente emessi da tale token di aggiornamento diventano non validi. Gli altri token di aggiornamento rilasciati all'utente non sono interessati da questa operazione.

RevokeToken operazione

RevokeTokenrevoca tutti i token di accesso per un determinato token di aggiornamento, incluso il token di accesso iniziale dall'accesso interattivo. Questa operazione non ha effetto su nessuno degli altri token di aggiornamento dell'utente o sui figli ID e token di accesso di quegli altri token di aggiornamento.

Endpoint di revoca

L'endpoint di revoca revoca un determinato token di aggiornamento e tutti gli ID e i token di accesso generati dal token di aggiornamento. Questo endpoint revoca anche il token di accesso iniziale dall'accesso interattivo. Le richieste a questo endpoint non influiscono su nessuno degli altri token di aggiornamento dell'utente o sui figli ID e token di accesso di tali altri token di aggiornamento.

GlobalSignOut operazione

GlobalSignOutè un'operazione self-service che un utente autorizza con il proprio token di accesso. Questa operazione revoca tutti i token di aggiornamento, ID e accesso dell'utente richiedente.

AdminUserGlobalSignOut operazione

AdminUserGlobalSignOutè un'operazione lato server che un amministratore autorizza con credenziali IAM. Questa operazione revoca tutti i token di aggiornamento, ID e accesso dell'utente di destinazione.

Cose da sapere sulla revoca dei token

- La richiesta per revocare un token di aggiornamento deve includere l'ID client utilizzato per ottenerlo.
- I JWT del pool di utenti sono autonomi con una firma e un orario di scadenza assegnati al momento della creazione del token. I token revocati non possono essere utilizzati con chiamate API Amazon Cognito che richiedono un token. Tuttavia, i token revocati saranno comunque validi se vengono verificati utilizzando una qualsiasi libreria JWT che verifica la firma e la scadenza del token.
- Quando crei un nuovo client del bacino d'utenza, la revoca dei token è attivata di default.
- È possibile revocare i token di aggiornamento solo nei client di app con la revoca dei token abilitata.
- Dopo aver abilitato la revoca dei token, nuove attestazioni vengono aggiunte nei token Web JSON di Amazon Cognito. Le attestazioni `origin_jti` e `jti` vengono aggiunte ai token di accesso e ID. Queste attestazioni aumentano le dimensioni dei token ID e di accesso del client dell'app.
- Quando disabiliti la revoca dei token in un client di app in cui era abilitata in precedenza, i token revocati non diventano più attivi.
- Quando disabiliti un account utente (che revoca i token di aggiornamento e accesso), i token revocati non diventano attivi se abiliti nuovamente l'account utente.
- Quando crei un nuovo client con pool di utenti utilizzando l', l'o l' AWS API Console di gestione AWS AWS CLI, la revoca dei token è abilitata per impostazione predefinita.

Abilitazione della revoca dei token

Prima di poter revocare un token per un client del bacino d'utenza esistente, è necessario abilitare la revoca del token. È possibile abilitare la revoca dei token per i client del pool di utenti esistenti utilizzando AWS CLI o l' AWS API. Per fare ciò, chiama il comando `aws cognito-idp describe-user-pool-client` della CLI o l'operazione API `DescribeUserPoolClient` per recuperare le impostazioni correnti dal client app. Quindi, chiama il comando `aws cognito-idp update-user-pool-client` della CLI o l'operazione API `UpdateUserPoolClient`. Includi le impostazioni correnti dal client app e imposta il parametro `EnableTokenRevocation` su `true`.

Per creare o modificare un client di app con revoca del token abilitata con l'API Amazon Cognito o con AWS un SDK, includi il seguente parametro nella CreateUserPoolClienttua UpdateUserPoolClientrichiesta o API.

```
"EnableTokenRevocation": true
```

Per configurare la revoca dei token nella console Amazon Cognito, seleziona un client dell'app dal menu App client nel tuo pool di utenti. Seleziona il pulsante Modifica nelle informazioni sul client dell'app e abilita o disabilita la revoca del token in Configurazione avanzata.

Revoca di un token

È possibile revocare un token di aggiornamento utilizzando una richiesta RevokeTokenAPI, ad esempio con il comando `aws cognito-idp revoke-token` CLI. Puoi anche revocare i token usando il Endpoint Revoke. Questo endpoint diventa disponibile dopo l'aggiunta di un dominio al bacino d'utenza. Puoi utilizzare l'endpoint di revoca su un dominio ospitato Amazon Cognito o sul tuo dominio personalizzato.

Di seguito è riportato il corpo di una richiesta API `RevokeToken` di esempio.

```
{
   "ClientId": "1example23456789",
   "ClientSecret": "abcdef123456789ghijklexample",
   "Token": "eyJjdHkiOiJKV1QiEXAMPLE"
}
```

Di seguito è riportato un esempio di richiesta cURL all'endpoint `/oauth2/revoke` di un pool di utenti con un dominio personalizzato.

```
curl --location 'auth.mydomain.com/oauth2/revoke' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: Basic Base64Encode(client_id:client_secret)' \
--data-urlencode 'token=abcdef123456789ghijklexample' \
--data-urlencode 'client_id=1example23456789'
```

L'operazione `RevokeToken` e l'endpoint `/oauth2/revoke` non richiedono alcuna autorizzazione aggiuntiva a meno che il client dell'app non disponga di un segreto del client.

Verifica dei token web JSON

I token web JSON (JWT) possono essere decodificati, letti e modificati facilmente. Un token di accesso modificato comporta il rischio di un aumento dei privilegi. Un token ID modificato crea un

rischio di impersonificazione. L'applicazione considera attendibile il pool di utenti come emittente di token, ma cosa succede se un utente intercetta il token in transito? Devi assicurarti che la tua applicazione riceva lo stesso token emesso da Amazon Cognito.

Amazon Cognito emette token che utilizzano alcune delle funzionalità di integrità e riservatezza della specifica OpenID Connect (OIDC). I token del pool di utenti indicano la validità con oggetti come l'ora di scadenza, l'emittente e la firma digitale. La firma, il terzo e ultimo segmento del JWT `.` delimitato, è il componente chiave della convalida dei token. Un utente malintenzionato può modificare un token, ma se l'applicazione recupera la chiave pubblica e confronta la firma, questa non corrisponderà. Qualsiasi applicazione che elabora JWT dall'autenticazione OIDC deve eseguire questa operazione di verifica a ogni accesso.

In questa pagina, forniamo alcuni consigli generali e specifici per la verifica dei JWT. Lo sviluppo di applicazioni abbraccia una varietà di linguaggi e piattaforme di programmazione. Poiché Amazon Cognito implementa OIDC in modo sufficientemente simile alle specifiche pubbliche, qualsiasi libreria JWT affidabile nel tuo ambiente di sviluppo preferito può gestire i tuoi requisiti di verifica.

Queste fasi descrivono la verifica del token JWT (JSON Web Token) di un pool di utenti.

Argomenti

- Prerequisiti
- Convalida dei token con aws-jwt-verify
- Comprensione e ispezione dei token

Prerequisiti

La tua libreria, SDK o il framework del software potrebbero già gestire le attività in questa sezione. AWS Gli SDK forniscono strumenti per la gestione e la gestione dei token del pool di utenti di Amazon Cognito nella tua app. AWS Amplify include funzioni per recuperare e aggiornare i token Amazon Cognito.

Per ulteriori informazioni, consulta le pagine seguenti.

- Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili
- Esempi di codice per Amazon Cognito Identity Provider che utilizza SDK AWS
- Advanced workflows (Flussi di lavoro avanzati) in Amplify Dev Center (Centro per sviluppatori Amplify)

Sono disponibili molte librerie per la decodifica e la verifica di un token web JSON (JWT). Queste librerie sono utili se vuoi gestire i token manualmente per l'elaborazione dell'API lato server oppure se stai usando altri linguaggi di programmazione. Consulta l'elenco di librerie OpenID Foundation per l'operatività con i token JWT.

Convalida dei token con aws-jwt-verify

In un' Node.js app, AWS consiglia alla aws-jwt-verifylibreria di convalidare i parametri nel token che l'utente passa all'app. Con `aws-jwt-verify` puoi popolare `CognitoJwtVerifier` con i valori delle attestazioni che desideri verificare per uno o più pool di utenti. Di seguito sono riportati alcuni valori che possono essere verificati.

- Verifica che i token di accesso o ID non siano malformati o scaduti e abbiano una firma valida.
- Verifica che i token di accesso provengano dai pool di utenti e dai client dell'app corretti.
- Verifica che le attestazioni del token di accesso contengano gli ambiti OAuth 2.0 corretti.
- Verifica che le chiavi che hanno firmato i token ID e di accesso corrispondano a una chiave di firma `kid` dell'URI JWKS dei pool di utenti.

  L'URI JWKS contiene le informazioni pubbliche sulla chiave privata usata per la firma del token dell'utente. Puoi trovare l'URI JWKS del tuo pool di utenti all'indirizzo `https://cognito-idp.<Region>.amazonaws.com/<userPoolId>/.well-known/jwks.json`.

Per ulteriori informazioni e un codice di esempio che puoi utilizzare in un' Node.js app o in un programma di AWS Lambda autorizzazione, consulta aws-jwt-verifyon. GitHub

> **Note**
>
> Quando l'attestazione `iss` (emittente) contenuta in un token non corrisponde al pool di utenti `CognitoJwtVerifier` con cui hai configurato il token, la verifica ha esito negativo con un. `ParameterValidationError` Ciò accade in genere quando un token viene presentato a un verificatore per un pool di utenti o una regione diversi. AWS Verifica che il destinatario `userPoolId` che passi `CognitoJwtVerifier.create()` corrisponda all'emittente dei token ricevuti dalla tua applicazione.

Comprensione e ispezione dei token

Prima di integrare l'ispezione dei token nell'app occorre considerare come Amazon Cognito assembla i JWT. Recupera i token di esempio dal pool di utenti, decodificali ed esaminali accuratamente per comprenderne le caratteristiche e determinare cosa vuoi verificare e quando. Ad esempio, potresti voler esaminare l'appartenenza al gruppo in uno scenario e gli ambiti in un altro.

Le sezioni seguenti descrivono un processo per ispezionare manualmente i JWT di Amazon Cognito durante la preparazione dell'app.

Conferma della struttura del JWT

Un token web JSON (JWT) include tre sezioni separate da un delimitatore `.` (punto).

Header

L'ID della chiave (`kid`) e l'algoritmo RSA (`alg`) utilizzati da Amazon Cognito per firmare il token. Amazon Cognito firma i token con un `alg` di `RS256`. `kid`Si tratta di un riferimento troncato a una chiave di firma privata RSA a 2048 bit detenuta dal pool di utenti.

Carico utile

Le attestazioni del token. In un token ID, le attestazioni includono gli attributi dell'utente e le informazioni sul pool di utenti (`iss`) e sul client dell'app (`aud`). In un token di accesso, il payload include gli ambiti, l'appartenenza al gruppo, il pool di utenti come `iss` e il client dell'app come `client_id`.

Firma

La firma non è decodificabile in base64url come l'header e il payload. È un identificatore RSA256 derivato da una chiave di firma e dai parametri che puoi osservare all'URI JWKS.

L'intestazione e il payload sono JSON con codifica base64url. Puoi identificarli tramite i caratteri di apertura `eyJ` che si decodificano nel carattere iniziale `{`. Se l'utente presenta alla tua app un JWT con codifica base64url e non è nel formato`[JSON Header].[JSON Payload].[Signature]`, non è un token Amazon Cognito valido e puoi eliminarlo.

L'applicazione di esempio seguente verifica i token del pool di utenti con. `aws-jwt-verify`

```
// cognito-verify.js
```

```
// Usage example: node cognito-verify.js eyJra789ghiEXAMPLE

const { CognitoJwtVerifier } = require('aws-jwt-verify');

// Replace with your Amazon Cognito user pool ID
const userPoolId = 'us-west-2_EXAMPLE';

async function verifyJWT(token) {
  try {
    const verifier = CognitoJwtVerifier.create({
      userPoolId,
      tokenUse: 'access', // or 'id' for ID tokens
      clientId: '1example23456789', // you must verify the token audience
    });

    const payload = await verifier.verify(token);
    console.log('Decoded JWT:', payload);
  } catch (err) {
    console.error('Error verifying JWT:', err);
  }
}

// Example usage
if (process.argv.length < 3) {
  console.error('Please provide a JWT token as an argument.');
  process.exit(1);
}

const MyToken = process.argv[2];
verifyJWT(MyToken);
```

Convalida del JWT

La firma JWT è una combinazione con hash di intestazione e payload. Amazon Cognito genera due coppie di chiavi di crittografia RSA per ogni bacino d'utenza. Una chiave privata firma i token di accesso e l'altra i token ID.

Per verificare la firma di un token JWT

1. Decodifica il token ID.

   Anche OpenID Foundation mantiene un elenco di librerie per l'utilizzo di token JWT.

È inoltre possibile utilizzarlo AWS Lambda per decodificare i JWT del pool di utenti. Per ulteriori informazioni, consulta Decodificare e verificare utilizzando i token Amazon Cognito JWT. AWS Lambda

2. Confronta l'ID della chiave (`kid`) locale con il `kid` pubblico.

   a. Scarica e archivia la chiave JWK (JSON Web Key) pubblica corrispondente per il tuo bacino d'utenza. È disponibile come parte di un set JWKS (JSON Web Key Set). Puoi individuarlo creando il seguente URI `jwks_uri` per l'ambiente:

   ```
   https://cognito-idp.<Region>.amazonaws.com/<userPoolId>/.well-known/jwks.json
   ```

   Per ulteriori informazioni su JWK e JWKS, consulta l'articolo su JSON Web Key (JWK).

   > **Note**
   >
   > Amazon Cognito può alternare le chiavi di firma nel pool di utenti. Come procedura consigliata, memorizza nella cache le chiavi pubbliche dell'app utilizzando il `kid` come chiave di cache e aggiorna periodicamente la cache. Confronta il `kid` nei token che la tua app riceve nella cache.
   > Se ricevi un token con l'emittente corretto ma un diverso `kid`, è possibile che Amazon Cognito abbia alternato la chiave di firma. Aggiorna la cache dall'endpoint `jwks_uri` del pool di utenti.

   Questo è un file `jwks.json` di esempio:

   ```
   {
    "keys": [{
     "kid": "1234example=",
     "alg": "RS256",
     "kty": "RSA",
     "e": "AQAB",
     "n": "1234567890",
     "use": "sig"
    }, {
     "kid": "5678example=",
     "alg": "RS256",
     "kty": "RSA",
   ```

```
  "e": "AQAB",
  "n": "987654321",
  "use": "sig"
 }]
}
```

ID chiave (**`kid`**)

Il `kid` è un suggerimento che indica quale chiave è stata utilizzata per proteggere la JSON Web Signature (JWS) del token.

Algoritmo (**`alg`**)

Il parametro di intestazione `alg` rappresenta l'algoritmo di crittografia utilizzato per proteggere il token ID. I pool di utenti utilizzano un algoritmo crittografico RS256, che è una firma RSA con. SHA-256 Per ulteriori informazioni su RSA, consulta crittografia RSA.

Tipo di chiavi (**`kty`**)

Il parametro `kty` identifica la famiglia di algoritmi di crittografia usata con la chiave, come "RSA" in questo esempio.

Esponente RSA (**`e`**)

Il parametro `e` contiene il valore dell'esponente per la chiave pubblica RSA. È rappresentato come valore. Base64urlUInt-encoded

Modulo RSA (**`n`**)

Il parametro `n` contiene il valore del modulo per la chiave pubblica RSA. È rappresentata come Base64urlUInt-encoded valore.

Utilizza (**`use`**)

Il parametro `use` descrive l'uso previsto della chiave pubblica. Per questo esempio, il valore `use` value `sig` rappresenta la firma.

b. Cerca la JSON Web Key pubblica per un `kid` che corrisponde al `kid` del JWT.

Verifica delle attestazioni

Per verificare le attestazioni JWT

1. Verifica che il token non sia scaduto. Decodifica il token e confronta la richiesta `exp` con l'ora corrente. Poiché i JWT del pool di utenti sono autonomi e hanno una propria ora di scadenza, puoi controllare la scadenza offline, senza una chiamata di rete ad Amazon Cognito.

   > **Note**
   >
   > Le operazioni API come GetUsero l'Endpoint UserInfoendpoint restituiscono un errore per un token di accesso scaduto, ma una richiesta rifiutata è un effetto collaterale dell'autorizzazione, non un metodo di verifica dei token. Non utilizzare queste chiamate di rete per verificare la scadenza: decodifica il token e confronta invece l'attestazione. `exp`

2. L'`aud`attestazione in un token ID e l'`client_id`attestazione in un token di accesso devono corrispondere all'ID client dell'app creato nel pool di utenti di Amazon Cognito.
3. L'attestazione issuer (`iss`) deve corrispondere al tuo pool di utenti. Ad esempio, un pool di utenti creato nella `us-east-1` regione ha il seguente `iss` valore:

   `https://cognito-idp.us-east-1.amazonaws.com/<userpoolID>`.
4. Controlla l'attestazione `token_use`.

   - Se accetti il token di accesso solo nelle operazioni dell'API Web, il valore deve essere `access`.
   - Se utilizzi solo il token ID, il valore deve essere `id`.
   - Se si usano sia ID sia token di accesso, l'attestazione `token_use` deve essere `id` o `access`.
5. Conferma che il token non è stato revocato. Un token con una firma valida e un `exp` claim non scaduto può comunque essere stato revocato prima della scadenza, ad esempio quando un utente si disconnette o quando chiami l'operatore API. RevokeToken I controlli di firma e scadenza offline non sono in grado di rilevare da soli la revoca. Se la tua richiesta deve onorare la revoca prima della scadenza di un token, verifica lo stato del token su Amazon Cognito, ad esempio con una richiesta API autorizzata, o riduci la durata del token di accesso. Per ulteriori informazioni, consulta Fine delle sessioni utente con revoca del token.

Ora è possibile considerare attendibili le attestazioni all'interno del token.

> **Note**
>
> Alcuni Servizi AWS possono verificare Amazon Cognito JWT per te, anziché nel codice dell'applicazione. Se utilizzi Amazon Cognito come autorizzatore JWT per un'API HTTP Amazon API Gateway, fornisci gli stessi valori di emittente e pubblico nella configurazione dell'autorizzatore. Imposta l'emittente sull'endpoint del tuo pool di utentihttps://cognito-idp.*<Region>*.amazonaws.com/*<userPoolId>*, che è anche il valore del campo nel documento del issuer pool di utenti. /.well-known/openid-configuration Imposta il pubblico sull'ID del client dell'app.
> API Gateway confronta il pubblico con la aud rivendicazione del token o con la client_id rivendicazione quando il token non ha alcuna aud rivendicazione. I token ID di Amazon Cognito vengono trasmessi, mentre i token di accesso aud vengono trasmessi client_id e no, a aud meno che la richiesta non includa un'associazione resource-server. Per un token ID o un token di accesso senza associazione di risorse, imposta il pubblico sull'ID client dell'app. Quando l'applicazione richiede un token di accesso associato a un server di risorse, l'audattestazione del token viene impostata sull'identificatore della risorsa (l'URL del server delle risorse) anziché sull'ID del client dell'app; in tal caso, configura l'URL del server di risorse come pubblico in modo che API Gateway corrisponda su di esso. Per i passaggi di configurazione di API Gateway, consulta Controllo dell'accesso alle API REST utilizzando i pool di utenti di Amazon Cognito come autorizzatore e Controllo dell'accesso alle API HTTP con gli autorizzatori JWT in API Gateway.
> Un Application Load Balancer può anche autenticare gli utenti con il tuo pool di utenti. Il load balancer trasmette le dichiarazioni utente firmate alle destinazioni in un'intestazione JWT-formatted x-amzn-oidc-data HTTP verificata dall'applicazione. Per ulteriori informazioni, consulta Autenticazione degli utenti tramite Application Load Balancer.

Gestione della scadenza e della memorizzazione nella cache dei token del pool di utenti

La tua app deve completare con successo una delle seguenti richieste ogni volta che desideri ottenere un nuovo token Web JSON (JWT).

- Richiesta di una concessione di credenziali client o codici di autorizzazione dall'Endpoint Token.
- Richiedi un token di accesso alle credenziali del cliente con l'operazione GetClientTokenAPI, che non richiede alcun dominio del pool di utenti.

- Richiedi una concessione implicita dalle tue pagine di accesso gestite.
- Autentica un utente locale in una richiesta API Amazon Cognito come. InitiateAuth

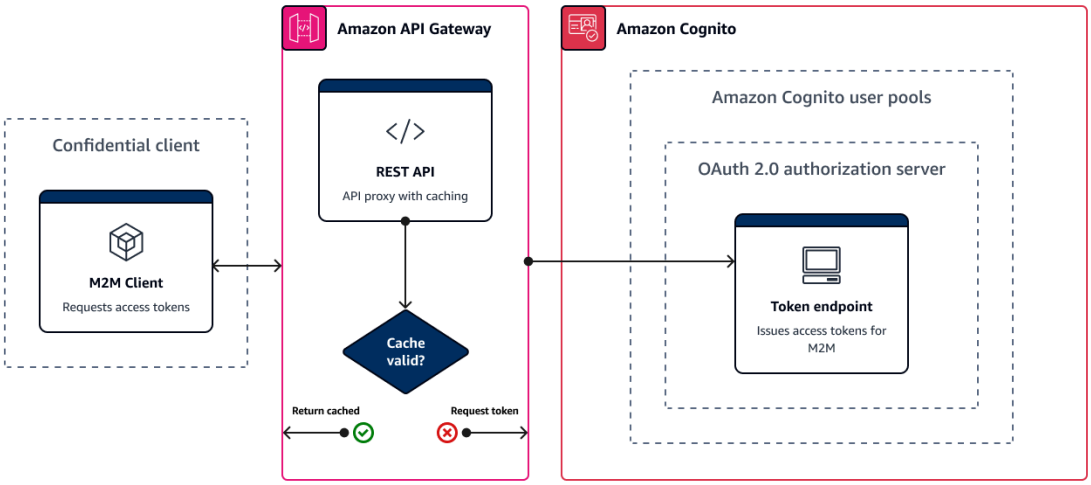
Puoi configurare il pool di utenti in modo da impostare la scadenza dei token entro un valore definito in minuti, ore o giorni. Per garantire le prestazioni e la disponibilità della tua app, utilizza i token Amazon Cognito per circa il 75% della durata del token e solo dopo recupera nuovi token. Una soluzione di caching creata per l'app mantiene disponibili i token e impedisce ad Amazon Cognito di rifiutare le richieste quando la frequenza delle richieste è troppo alta. Un'app lato client deve archiviare i token in una cache di memoria. Un'app lato server può aggiungere un meccanismo di caching crittografato per archiviare i token.

Quando il pool di utenti genera un volume elevato di attività utente-computer o computer-computer, si potrebbero incontrare i limiti imposti da Amazon Cognito al numero di richieste di token che è possibile effettuare. Per ridurre il numero di richieste inviate agli endpoint Amazon Cognito, è possibile archiviare e riutilizzare in modo sicuro i dati di autenticazione o implementare backoff esponenziale e nuovi tentativi.

I dati di autenticazione provengono da due classi di endpoint. Gli endpoint Amazon Cognito OAuth 2.0 includono l'endpoint token, che gestisce le credenziali dei client e le richieste di codice di autorizzazione all'accesso gestito. Gli endpoint del servizio rispondono a richieste API come `InitiateAuth` e `RespondToAuthChallenge`. Ogni tipo di richiesta è caratterizzato da un limite specifico. Per ulteriori informazioni sui limiti, consulta Quote in Amazon Cognito.

Caching di token di accesso computer-computer con Gateway Amazon API

Con il caching dei token di Gateway API, l'app può eseguire il ridimensionamento in risposta a eventi che superano il limite predefinito per la frequenza delle richieste degli endpoint OAuth di Amazon Cognito.



Puoi memorizzare nella cache i token di accesso in modo che l'app richieda un nuovo token di accesso solo se un token memorizzato nella cache è scaduto. In caso contrario, l'endpoint di caching restituisce un token dalla cache. Ciò evita una chiamata aggiuntiva a un endpoint dell'API Amazon Cognito. Quando si utilizza Gateway Amazon API come proxy per l'Endpoint Token, l'API risponde alla maggior parte delle richieste che altrimenti contribuirebbero alla quota di richieste, evitando richieste non riuscite risultanti dalla limitazione della frequenza.

La seguente Gateway-based soluzione API offre un'implementazione a bassa latenza e basso code/no codice della memorizzazione nella cache dei token. Le API di Gateway API sono crittografate in transito e, facoltativamente, quando sono inattive. Una cache di Gateway API è ideale per la concessione di credenziali client OAuth 2.0, un tipo di concessione spesso ad alto volume che genera token di accesso per autorizzare sessioni computer-computer e microservizi. In un evento come un aumento del traffico che causa la scalabilità orizzontale dei microservizi, è possibile che molti sistemi utilizzino le stesse credenziali client a un volume che supera il limite di frequenza delle AWS richieste del pool di utenti o del client dell'app. Per preservare la disponibilità delle app e la bassa latenza, una soluzione di caching è la best practice consigliata in tali scenari.

In questa soluzione si definisce una cache nell'API per archiviare un token di accesso distinto per ogni combinazione di ambiti OAuth e client dell'app che si desidera richiedere nell'app. Quando l'app effettua una richiesta corrispondente alla chiave della cache, l'API risponde con un token di accesso emesso da Amazon Cognito alla prima richiesta corrispondente alla chiave della cache. Alla scadenza della chiave, l'API inoltra la richiesta all'endpoint del token e memorizza nella cache un nuovo token di accesso.

> **Note**
>
> La durata della chiave della cache deve essere inferiore alla durata del token di accesso del client dell'app.

La chiave cache è una combinazione degli ambiti OAuth richiesti nel `scope` parametro nel corpo della richiesta e nell'intestazione della richiesta. `Authorization` L'intestazione `Authorization` contiene l'ID del client dell'app e il segreto client. Non è necessario implementare una logica aggiuntiva nell'app per implementare questa soluzione. È solo necessario aggiornare la configurazione per modificare il percorso dell'endpoint del token del pool di utenti.

Puoi anche implementare la memorizzazione nella cache dei token con ElastiCache (Redis OSS). Per un controllo granulare con le policy AWS Identity and Access Management (IAM), valuta l'ipotesi di utilizzare una cache Amazon DynamoDB.

> **Note**
>
> Il caching in Gateway API è soggetto a costi aggiuntivi. Per maggiori dettagli, consulta la pagina relativa ai prezzi.

Per configurare un proxy di caching con Gateway API

1. Apri la Console Gateway API e crea una REST API.
2. In Resources (Risorse), crea un metodo POST.
   a. In Integration type (Tipo di integrazione), scegli HTTP.
   b. Seleziona Use HTTP proxy integration (Utilizza integrazione proxy HTTP).
   c. In Endpoint URL (URL endpoint), immetti un URL nel formato `https://<your user pool domain>/oauth2/token`.
3. In Resources (Risorse), configura la chiave della cache.
   a. In Method request (Richiesta metodo), modifica la richiesta del metodo POST.

> **Note**
>
> Questo metodo di convalida della richiesta deve essere utilizzato con l'`client_secret_basic`autorizzazione nelle richieste di token, in cui il segreto del client è codificato nell'intestazione della richiesta. `Authorization` Per la convalida del corpo della richiesta JSON nell'`client_secret_post`autorizzazione, crea invece un modello di dati che richieda la presenza di client_secret. In questo modello, il validatore Request dovrebbe convalidare il corpo, i parametri della stringa di query e le intestazioni.

b. Configura il metodo Request validator per convalidare i parametri e le intestazioni della stringa di query. Per ulteriori informazioni sulla convalida della richiesta, consulta Request validation nella Amazon API Gateway Developer Guide.

c. Imposta il parametro `scope` e l'intestazione `Authorization` come chiave di caching.

   i. Aggiungi una stringa di query ai parametri della stringa di query URL. Inserisci il nome della stringa di query `scope` e seleziona Obbligatorio e Memorizzazione nella cache.

   ii. Aggiungi un'intestazione alle intestazioni delle richieste HTTP. Inserisci il nome dell'intestazione della richiesta **`Authorization`** e seleziona Obbligatorio e Memorizzazione nella cache.

4. In Stages (Fasi), configura il caching.

   a. Scegliete la fase che desiderate modificare e scegliete Modifica da Dettagli della fase.

   b. In Impostazioni aggiuntive, Impostazioni cache, attiva l'opzione cache Provision API.

   c. Scegli un valore in Cache capacity (Capacità cache). Una maggiore capacità di cache migliora le prestazioni ma comporta costi aggiuntivi.

   d. Deseleziona la casella di controllo Require authorization (Richiedi autorizzazione). Seleziona Continua.

   e. API Gateway applica le policy di cache ai metodi GET solo a livello di fase. È necessario applicare un'eccezione alla politica della cache al metodo POST.

      Espandi la fase che hai configurato e seleziona il `POST` metodo. Per creare le impostazioni della cache per il metodo, scegliete Create override.

   f. Attiva l'opzione Abilita la cache del metodo.

   g. Immettete un time-to-live (TTL) della cache di 3600 secondi. Selezionare Salva.

5. In Stages (Fasi), annota il valore visualizzato in Invoke URL (Richiama URL).
6. Aggiorna la tua app per le richieste di token POST in base al valore di Invoke URL (Richiama URL) della tua API anziché in base all'endpoint `/oauth2/token` del pool di utenti.

Ambiti, M2M e server di risorse

Dopo aver configurato un dominio per il pool di utenti, Amazon Cognito effettua automaticamente il provisioning di un server di autorizzazione OAuth 2.0 e di un'interfaccia utente Web ospitata con pagine di registrazione e accesso mostrate dall'app agli utenti. Per ulteriori informazioni, consulta Accesso gestito dal pool di utenti. Puoi scegliere gli ambiti che desideri che il server di autorizzazione aggiunga ai token di accesso. Gli ambiti autorizzano l'accesso ai server di risorse e ai dati degli utenti.

Un server di risorse è un server API OAuth 2.0. Per proteggere le risorse protette dall'accesso, verifica che i token di accesso del pool di utenti contengano gli ambiti che autorizzano il metodo e il percorso richiesti nell'API che protegge. Verifica l'emittente in base alla firma del token, la validità in base alla scadenza del token e il livello di accesso in base agli ambiti delle richieste di token. Gli ambiti del pool di utenti sono indicati nella dichiarazione del token di accesso. `scope` Per ulteriori informazioni sulle dichiarazioni nei token di accesso di Amazon Cognito, consulta Comprendere il token di accesso.

Con Amazon Cognito, gli ambiti dei token di accesso possono autorizzare l'accesso alle API esterne o agli attributi utente. Puoi emettere token di accesso a utenti locali, utenti federati o identità di macchine.

Argomenti

- Autorizzazione dell'API
- Machine-to-machine Autorizzazione (M2M)
- Informazioni sugli ambiti
- Informazioni sui server di risorse
- Associazione di risorse

Autorizzazione dell'API

Di seguito sono riportati alcuni dei modi in cui puoi autorizzare le richieste alle API con i token Amazon Cognito:

Token di accesso

Quando aggiungi un autorizzatore Amazon Cognito a una configurazione di richiesta di metodo API REST, aggiungi gli ambiti di autorizzazione alla configurazione dell'autorizzatore. Con questa configurazione, l'API accetta i token di accesso nell'`Authorization`intestazione e li esamina per verificare gli ambiti accettati.

Token ID

Quando passi un token ID valido a un autorizzatore Amazon Cognito nella tua API REST, API Gateway accetta la richiesta e trasmette il contenuto del token ID al backend dell'API.

Autorizzazioni verificate da Amazon

In Autorizzazioni verificate, hai la possibilità di creare un API-linked archivio di policy. Verified Permissions crea e assegna un autorizzatore Lambda che elabora ID o token di accesso dall'intestazione della richiesta. `Authorization` Questo autorizzatore Lambda passa il token all'archivio delle politiche, dove Verified Permissions lo confronta con le politiche e restituisce una decisione di autorizzazione o rifiuto all'autorizzatore.

Altre risorse

- Controllo e gestione dell'accesso a un'API REST in API Gateway
- Autorizzazione con Amazon Verified Permissions

Machine-to-machine Autorizzazione (M2M)

Amazon Cognito supporta applicazioni che accedono ai dati delle API con identità di macchina. Le identità delle macchine nei pool di utenti sono client riservati che vengono eseguiti su server di applicazioni e si connettono a API remote. Il loro funzionamento avviene senza l'interazione dell'utente: attività pianificate, flussi di dati o aggiornamenti delle risorse. Quando questi client autorizzano le proprie richieste con un token di accesso, eseguono l'autorizzazione da macchina a macchina o M2M. Nell'autorizzazione M2M, un segreto condiviso sostituisce le credenziali dell'utente nel controllo degli accessi.

Un'applicazione che accede a un'API con autorizzazione M2M deve avere un ID client e un client secret. Nel tuo pool di utenti, crea un client per l'app con un client secret. L'identità della macchina può quindi ottenere un token di accesso M2M in due modi: può richiedere la concessione delle credenziali del client da parte di o può richiamare l'operazione GetClientTokenAPI. Endpoint Token

Entrambi gli approcci emettono un token di accesso che autorizza solo gli ambiti personalizzati dei server di risorse. La configurazione differisce: l'endpoint del token richiede un dominio di pool di utenti e si adatta alle applicazioni che utilizzano una libreria OIDC, mentre non `GetClientToken` richiede alcun dominio e funziona tramite l' AWS SDK o l'API. AWS CLI Per ulteriori informazioni sulla configurazione dei client delle app, consulta. Application-specific impostazioni con i client dell'app

Il token di accesso M2M è una dichiarazione verificabile delle operazioni che si desidera consentire all'identità della macchina di richiedere a un'API. Per saperne di più su come i token di accesso autorizzano le richieste API, continua a leggere. Per un'applicazione di esempio, consulta Amazon Cognito e l'autorizzazione da macchina a macchina basata su Amazon Cognito e API Gateway tramite AWS CDK.

L'autorizzazione M2M ha un modello di fatturazione diverso dal modo in cui vengono fatturate gli utenti attivi mensili (MAU). Laddove l'autenticazione utente comporta un costo per utente attivo, la fatturazione M2M riflette i client attivi delle app M2M, indipendentemente dal fatto che utilizzino le credenziali del cliente o il funzionamento dell'API, e il volume totale delle richieste di token. `GetClientToken` Per ulteriori informazioni, consultare Prezzi di Amazon Cognito. Per controllare i costi dell'autorizzazione M2M, ottimizza la durata dei token di accesso e il numero di richieste di token effettuate dalle tue applicazioni. Scopri come utilizzare Gestione della scadenza e della memorizzazione nella cache dei token del pool di utenti la memorizzazione nella cache di API Gateway per ridurre le richieste di nuovi token nell'autorizzazione M2M.

Per informazioni sull'ottimizzazione delle operazioni di Amazon Cognito che aggiungono costi alla bolletta, AWS consulta. Gestione dei costi

Metadati client per le credenziali client da macchina a macchina (M2M)

È possibile trasmettere i metadati dei client nelle richieste M2M. I metadati del client sono informazioni aggiuntive provenienti da un ambiente utente o applicativo che possono contribuire ai risultati di un. Trigger Lambda di pre-generazione del token Nelle operazioni di autenticazione con un user principal, è possibile passare i metadati del client al trigger precedente alla generazione del token nel corpo delle richieste API AdminRespondToAuthChallenge. RespondToAuthChallenge Poiché le applicazioni gestiscono il flusso per la generazione di token di accesso per M2M con richieste dirette aEndpoint Token, hanno un modello diverso. Nel corpo POST delle richieste di token per le credenziali del client, passate un `aws_client_metadata` parametro con l'oggetto di metadati del client URL-encoded () `x-www-form-urlencoded` a string. Per una richiesta di esempio, consulta Credenziali del client con autorizzazione di base. Di seguito è riportato un parametro di esempio che passa le coppie chiave-valore. `{"environment": "dev", "language": "en-US"}`

```
aws_client_metadata=%7B%22environment%22%3A%20%22dev%22,%20%22language%22%3A%20%22en-US%22%7D
```

Ottenere token di accesso M2M con GetClientToken

L'operazione GetClientTokenAPI emette un token di accesso M2M per un client di app riservato tramite l' AWS SDK, o API AWS CLI, senza un dominio del pool di utenti. Fornisce le stesse funzionalità delle credenziali del client concesse dall'endpoint del token; entrambe autorizzano un'applicazione anziché un utente.

> **Note**
>
> Amazon Cognito non utilizza policy basate sull'identità AWS Identity and Access Management (IAM) per l'autorizzazione. `GetClientToken` Non puoi utilizzare le credenziali IAM per autorizzare la richiesta; l'autorizzazione proviene dal segreto del client dell'app.

Per utilizzarlo`GetClientToken`, configura un client dell'app con un client secret e solo il flusso di `ALLOW_CLIENT_TOKEN_AUTH` autenticazione. Questo flusso si esclude a vicenda con i flussi di autenticazione degli utenti. Un client di app può avere fino a due segreti attivi. Associa gli ambiti personalizzati del tuo server di risorse al client dell'app. L' AWS CLI esempio seguente crea un client di app compatibile.

```
aws cognito-idp create-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-name my-m2m-client \
    --generate-secret \
    --explicit-auth-flows ALLOW_CLIENT_TOKEN_AUTH \
    --allowed-o-auth-scopes "solar-system-data/asteroids.add"
```

Per ottenere un token di accesso, chiama `GetClientToken` con l'ID client dell'app, un client secret attivo e gli ambiti personalizzati che desideri includere nel token. `GetClientToken`non accetta un `UserPoolId` parametro; Amazon Cognito identifica il pool di utenti dall'ID client. Se non lo specifichi`Scopes`, Amazon Cognito autorizza gli ambiti configurati per il client dell'app.

```
aws cognito-idp get-client-token \
    --client-id 1example23456789 \
    --secret exampleClientSecret123EXAMPLE \
    --scopes "solar-system-data/asteroids.add"
```

La risposta contiene il token di accesso e i relativi metadati in un oggetto. `ClientAuthenticationResult`

```
{
    "ClientAuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer"
    }
}
```

Presenta il token di accesso al tuo server di risorse, ad esempio, nell'`Authorization`intestazione di una richiesta a un'API REST di Amazon API Gateway. Il server di risorse verifica la firma e la scadenza del token, quindi autorizza la richiesta in base agli ambiti del token.

Per passare dati a un utente Trigger Lambda di pre-generazione del token che personalizza il token, includi una `ClientMetadata` mappa nella richiesta con il parametro. `--client-metadata` Il trigger precedente alla generazione del token deve essere configurato per la versione 3 o successiva dell'evento trigger per ricevere i metadati del client nel flusso M2M. Amazon Cognito non archivia, convalida o crittografa questi dati e li rende disponibili solo per i trigger Lambda, quindi non include informazioni sensibili.

Informazioni sugli ambiti

Un ambito è un livello di accesso che un'app può richiedere a una risorsa. In un token di accesso Amazon Cognito, l'ambito è supportato dal trust che configuri con il tuo pool di utenti: un emittente affidabile di token di accesso con una firma digitale nota. I pool di utenti possono generare token di accesso con scopi che dimostrano che il cliente è autorizzato a gestire alcuni o tutti i propri profili utente o a recuperare dati da un'API di back-end. I pool di utenti di Amazon Cognito emettono token di accesso con l'ambito API riservato dei pool di utenti, gli ambiti personalizzati e gli ambiti OpenID Connect (OIDC).

L'ambito API riservato dei pool di utenti

L'`aws.cognito.signin.user.admin`ambito autorizza le operazioni self-service per l'utente corrente nell'API dei pool di utenti di Amazon Cognito. Autorizza il detentore di un token di accesso a interrogare e aggiornare tutte le informazioni sul portatore con, ad esempio, le operazioni e API. GetUserUpdateUserAttributes Quando autentichi il tuo utente con l'API dei pool di utenti di Amazon Cognito, questo è l'unico ambito che ricevi nel token di accesso. È anche l'unico ambito di cui hai bisogno per leggere e scrivere gli attributi utente che il client dell'app è autorizzato a leggere e

scrivere. Puoi anche richiedere questo ambito nelle richieste a Endpoint Authorize. Questo ambito da solo non è sufficiente per richiedere attributi utente da Endpoint UserInfo. Per i token di accesso che autorizzano l'API dei pool di utenti e le richieste `userInfo` per gli utenti, devi richiedere entrambi gli ambiti `openid` e `aws.cognito.signin.user.admin` in una richiesta `/oauth2/authorize`.

Ambiti personalizzati

Gli ambiti personalizzati autorizzano le richieste alle API esterne protette dai server di risorse. Puoi richiedere ambiti personalizzati con altri tipi di ambiti. Puoi trovare maggiori informazioni sugli ambiti personalizzati in questa pagina.

Ambiti OpenID Connect (OIDC)

Quando autentichi gli utenti con il server di autorizzazione del pool di utenti, incluso l'accesso gestito, devi richiedere gli ambiti. Puoi autenticare gli utenti locali del pool di utenti e gli utenti federati di terze parti nel tuo server di autorizzazione Amazon Cognito. Gli ambiti OIDC autorizzano l'app a leggere le informazioni utente dal pool di utenti. Endpoint UserInfo Il modello OAuth, in cui si interrogano gli attributi utente dall'endpoint `userInfo`, può ottimizzare l'app per un volume elevato di richieste di attributi utente. L'endpoint `userInfo` restituisce gli attributi a un livello di autorizzazione determinato dagli ambiti del token di accesso. Puoi autorizzare il client dell'app a emettere token di accesso con i seguenti ambiti OIDC.

openid

L'ambito minimo per query OpenID Connect (OIDC). Autorizza il token ID, l'attestazione di identificazione univoca `sub` e la possibilità di richiedere altri ambiti.

> **Note**
>
> Quando richiedi l'ambito `openid` e nessun altro, il token ID del pool di utenti e la risposta `userInfo` includono attestazioni per tutti gli attributi utente che il client dell'app è in grado di leggere. Quando richiedi `openid` e altri ambiti OIDC come`profile`, and `email`phone, il contenuto del token ID e della risposta UserInfo è limitato ai vincoli degli ambiti aggiuntivi.
> Ad esempio, una richiesta a Endpoint Authorize con il parametro `scope=openid+email` restituisce un token ID con `sub`, `email` e `email_verified`. Il token di accesso di questa richiesta restituisce gli stessi attributi daEndpoint UserInfo. Una richiesta con parametro `scope=openid` restituisce tutti gli attributi leggibili dal client nel token ID e da `userInfo`.

profile

Autorizza tutti gli attributi utente che il client dell'app è in grado di leggere.

e-mail

Autorizza gli attributi utente `email` e `email_verified`. Amazon Cognito restituisce `email_verified` se un valore è stato impostato in modo esplicito.

telefono

Autorizza gli attributi utente `phone_number` e `phone_number_verified`.

Informazioni sui server di risorse

Un'API del server di risorse potrebbe consentire l'accesso alle informazioni in un database o controllare le risorse IT. Un token di accesso di Amazon Cognito può autorizzare l'accesso alle API che supportano OAuth 2.0. Le REST API Gateway Amazon API sono caratterizzate dal supporto integrato per l'autorizzazione con i token di accesso di Amazon Cognito. L'app trasferisce il token d'accesso nella chiamata API ai server di risorse. Il server di risorse ispeziona il token d'accesso per determinare se gli accessi devono essere concessi.

Amazon Cognito potrebbe apportare future modifiche allo schema dei token di accesso del pool di utenti. Se la tua app analizza il contenuto del token di accesso prima di passarlo a un'API, devi progettare il codice per accettare gli aggiornamenti dello schema.

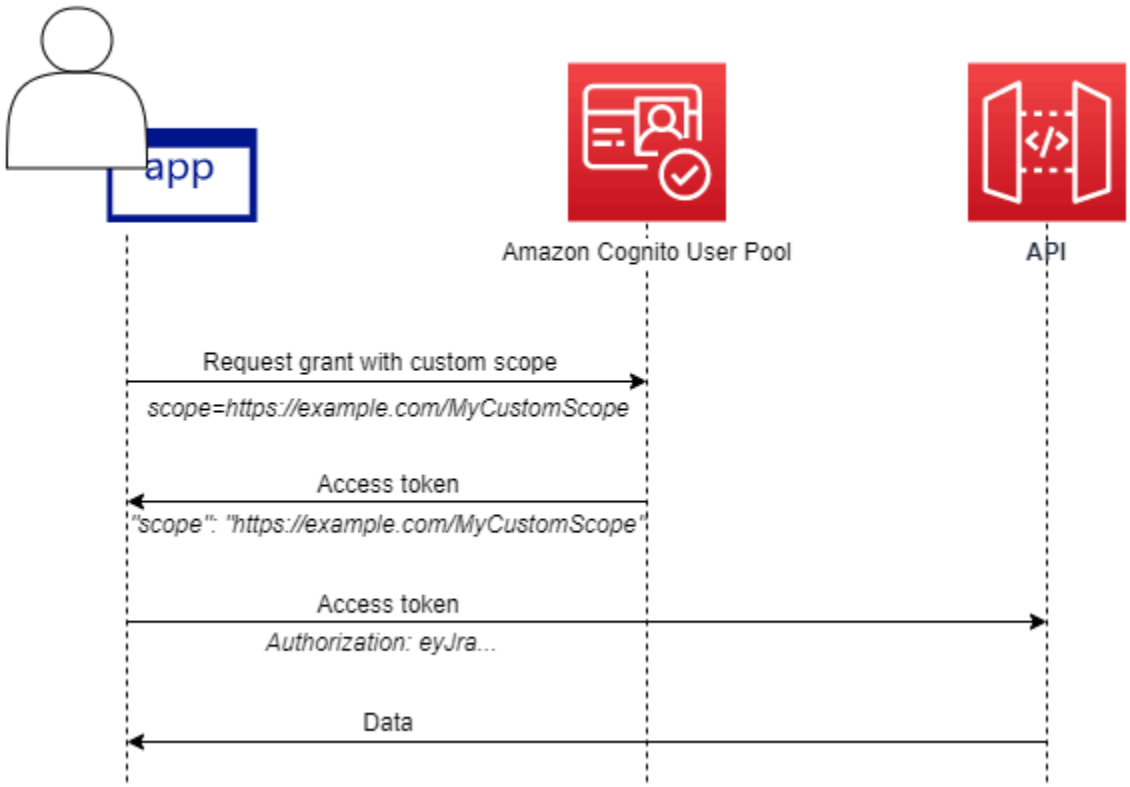
Gli ambiti personalizzati vengono definiti dall'utente ed estendono le funzionalità di autorizzazione di un pool di utenti per includere scopi non correlati all'interrogazione e alla modifica degli utenti e dei relativi attributi. Ad esempio, se disponete di un server di risorse per le foto, potrebbe definire due ambiti: per l'accesso in lettura alle foto e `photos.read` per l'accesso. `photos.write` write/delete Puoi configurare un'API per accettare i token di accesso per l'autorizzazione e la concessione di richieste `HTTP GET` per accedere ai token con `photos.read` nell'attestazione `scope` e richieste `HTTP POST` ai token con `photos.write`. Questi sono ambiti personalizzati.

> **Note**
>
> Per elaborare una qualsiasi richiesta all'interno del token, il tuo server di risorse deve verificare la firma e la data di scadenza dei token d'accesso. Per ulteriori informazioni sulla verifica dei token, consulta Verifica dei token web JSON. Per ulteriori informazioni sulla verifica e sull'utilizzo dei token del pool di utenti in Gateway Amazon API, consulta la sezione

Integrazione dei pool di utenti Amazon Cognito con API Gateway. API Gateway è una buona soluzione per l'ispezione dei token d'accesso e la protezione delle risorse. Per ulteriori informazioni sulle autorizzazioni Lambda di API Gateway, consulta Uso di autorizzazioni Lambda di API Gateway.

Panoramica di

Con Amazon Cognito, puoi creare server di risorse OAuth2.0 e associarvi ambiti personalizzati. Gli ambiti personalizzati in un token di accesso autorizzano azioni specifiche nell'API. Puoi autorizzare qualsiasi app del client nel pool di utenti a emettere ambiti personalizzati da uno qualsiasi dei server di risorse. Associa ambiti personalizzati a un client dell'app e richiedi questi ambiti nelle concessioni del codice di autorizzazione OAuth2.0, nelle concessioni implicite e nelle concessioni delle credenziali client dal Endpoint Token. Per M2M, puoi anche richiedere ambiti personalizzati con l'GetClientTokenoperazione API. Amazon Cognito aggiunge ambiti personalizzati nell'attestazione `scope` in un token di accesso. Un client può utilizzare il token d'accesso piuttosto che il suo server di risorse, facendo in modo che le decisioni legate all'autorizzazione siano basate sugli ambiti presenti nel token. Per ulteriori informazioni sui token d'accesso proposto, consulta Utilizzo dei token con i bacini d'utenza.



Per ottenere un token di accesso con ambiti personalizzati, l'app può richiedere al cliente di riscattare un codice di autorizzazione o di concedere le credenziali del cliente. Endpoint Token Nell'accesso gestito, puoi anche richiedere ambiti personalizzati in un token di accesso da una concessione implicita. Per l'autorizzazione M2M, l'app può anche richiedere ambiti personalizzati con l'operazione API. GetClientToken

> **Note**
>
> Perché sono progettate per l'autenticazione interattiva con il pool di utenti come IdP InitiateAuthe AdminInitiateAuthle richieste producono solo un'`scope`attestazione nel token di accesso con il singolo valore. `aws.cognito.signin.user.admin`

Gestione del server di risorse e degli ambiti personalizzati

Durante la creazione di un server di risorse, è necessario fornire un nome per il server di risorse e un identificatore per il server di risorse. Per ogni ambito che crei nel server di risorse, è necessario fornire nome e descrizione.

- Nome del server di risorse: un nome intuitivo per il server di risorse, ad esempio `Solar system object tracker` o `Photo API`.
- Identificatore del server di risorse: un identificatore univoco per il server di risorse. L'identificatore è qualsiasi nome che desideri associare alla tua API, ad esempio `solar-system-data`. Puoi configurare identificatori più lunghi, ad esempio `https://solar-system-data-api.example.com` come riferimento più diretto ai percorsi URI delle API, ma stringhe più lunghe aumentano la dimensione dei token di accesso.
- Nome ambito: il valore che desideri nell'attestazione `scope`. Ad esempio, `sunproximity.read`.
- Descrizione: una descrizione intuitiva dell'ambito. Ad esempio, `Check current proximity to sun`.

Amazon Cognito può includere ambiti personalizzati nei token di accesso per qualsiasi utente, locale nel pool di utenti o federato, con un provider di identità di terze parti. Puoi scegliere gli ambiti per i token di accesso degli utenti durante i flussi di autenticazione con il server di autorizzazione OAuth 2.0 che include l'accesso gestito. L'autenticazione dell'utente deve iniziare in corrispondenza di Endpoint Authorize con `scope` come uno dei parametri della richiesta. Di seguito è riportato un formato consigliato per i server di risorse. Per un identificatore, usa un nome intuitivo per le API. Per un ambito personalizzato, usa l'azione che autorizzano.

```
resourceServerIdentifier/scopeName
```

Ad esempio, hai scoperto un nuovo asteroide nella fascia di Kuiper e desideri registrarlo tramite la tua API `solar-system-data`. L'ambito che autorizza le operazioni di scrittura nel database degli asteroidi è `asteroids.add`. Quando richiedi il token di accesso che ti autorizza a registrare la tua scoperta, formatta il parametro di richiesta HTTPS come. `scope` `scope=solar-system-data/asteroids.add`

L'eliminazione di un ambito dal server di risorse non elimina la relativa associazione con tutti i client. Invece, l'ambito è contrassegnato come inattivo. Amazon Cognito non aggiunge ambiti inattivi ai token di accesso, ma per il resto procede normalmente se l'app ne richiede uno. Se aggiungi nuovamente l'ambito al tuo server di risorse in un secondo momento, Amazon Cognito lo scrive nuovamente nel token di accesso. Se richiedi un ambito che non hai associato all'app del client, a prescindere che venga eliminato dal server di risorse del pool di utenti, l'autenticazione non va a buon fine.

Puoi utilizzare l'API o Console di gestione AWS la CLI per definire server di risorse e ambiti per il tuo pool di utenti.

Definizione di un server di risorse per il pool di utenti (Console di gestione AWS)

È possibile utilizzare il Console di gestione AWS per definire un server di risorse per il pool di utenti.

Definizione di un server di risorse

1. Accedi alla console Amazon Cognito.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegli il menu Dominio sotto Branding e individua i server di risorse.
4. Seleziona Create a resource server (Crea un server di risorse).
5. Inserisci un nome del server di risorse. Ad esempio, `Photo Server`.
6. Inserisci un Resource server identifier (identificatore del server di risorse). Ad esempio, `com.example.photos`.
7. Inserisci gli Custom scopes (ambiti personalizzati) per le risorse, come `read` e `write`.
8. Per ogni Scope name (nome ambito), inserisci una Description (descrizione), come `view your photos` e `update your photos`.

9. Scegli Crea.

Gli ambiti personalizzati possono essere esaminati nel menu Dominio sotto Resource servers, nella colonna Ambiti personalizzati. Gli ambiti personalizzati possono essere abilitati per i client delle app dal menu App client in Applicazioni. Seleziona un client per l'app, individua le pagine di accesso e scegli Modifica. Aggiungi degli Custom scopes (ambiti personalizzati) e seleziona Save changes (salva modifiche).

Definizione di un server di risorse per il pool di utenti (AWS CLI and AWS API)

Usa i seguenti comandi per specificare le impostazioni dei server di risorse per il tuo bacino d'utenza.

Creazione di un server di risorse

- AWS CLI: `aws cognito-idp create-resource-server`
- AWS API: CreateResourceServer

Ottenimento di informazioni sulle impostazioni dei tuoi server di risorse

- AWS CLI: `aws cognito-idp describe-resource-server`
- AWS API: DescribeResourceServer

Creazione di liste di informazioni su tutti i server di risorse per il tuo bacino d'utenza

- AWS CLI: `aws cognito-idp list-resource-servers`
- AWS API: ListResourceServers

Come eliminare un server di risorse

- AWS CLI: `aws cognito-idp delete-resource-server`
- AWS API: DeleteResourceServer

Aggiornamento delle impostazioni per un server di risorse

- AWS CLI: `aws cognito-idp update-resource-server`
- AWS API: UpdateResourceServer

Associazione di risorse

Con l'associazione delle risorse, nota anche come indicatori di risorse, è possibile richiedere le API-specific concessioni al server di autorizzazione del pool di utenti. Resource binding è un'estensione OAuth 2.0 definita in RFC 8707 che consente ai client di specificare esplicitamente a quale server di risorse intendono accedere durante le richieste di autorizzazione. Con l'associazione delle risorse, le configurazioni delle API possono rifiutare l'accesso a token che non sono specificamente destinati a loro.

> **Note**
>
> Puoi associare i token di accesso solo alle risorse per gli utenti. Non è possibile richiedere un'associazione di risorse con credenziali del cliente concesse da M2M.

Quando utilizzi l'associazione di risorse con i pool di utenti di Amazon Cognito, i client possono includere un `resource` parametro nelle loro richieste di autenticazione al server di autorizzazione del pool di utenti. Il tuo pool di utenti verifica che il valore della risorsa richiesta sia un URL, seguendo le stesse regole dello schema degli URL di callback dei client dell'app:`https://`, `http://` con `localhost` only o uno schema personalizzato come. `myapp://` Amazon Cognito imposta l'URI richiesto come destinatario nella `aud` rivendicazione del token di accesso. Se la risorsa richiesta è un server di risorse del pool di utenti, l'identificatore del server di risorse deve essere in un formato URL. È possibile richiedere una risorsa per richiesta di autenticazione.

Questa funzionalità è esclusiva dell'autenticazione di accesso gestita con il server di autorizzazione OAuth 2.0 del pool di utenti. È possibile richiedere l'associazione delle risorse in termini di concessioni implicite e basate su codice di autorizzazione da. Endpoint Authorize Token-refresh sovvenzioni derivanti dal Endpoint Token riporto del credito dalla richiesta originale. `aud` Al momento non è disponibile nei modelli di autenticazione SDK.

Implementa l'associazione delle risorse con il tuo pool di utenti Amazon Cognito

1. Configura uno o più server di risorse nel tuo pool di utenti con identificatori univoci.
2. Nella richiesta di autorizzazione a`/oauth2/authorize`, richiedi un codice di autorizzazione o una concessione implicita e includi il `resource` parametro. Il valore di `resource` deve essere un identificatore del server di URL-formatted risorse o un URL. Ad esempio, `&resource=https://solar-system-data-api.example.com`.

3. Il server di autorizzazione convalida la richiesta di risorse, completa l'autenticazione e imposta il `aud` claim del token di accesso sull'URL della risorsa richiesta.
4. Per verificare che i token siano stati emessi appositamente per tale scopo, la risorsa che utilizza il token di accesso dell'utente verifica il claim. `aud`

Gestione degli utenti

Dopo che gli utenti hanno effettuato l'accesso, puoi gestire i loro account per tutta la durata dell'applicazione. Questa sezione tratta l'amministrazione quotidiana di un elenco utenti: creazione e importazione di utenti, organizzazione in gruppi e connessione di un pool di utenti a un pool di identità in modo che anche gli utenti possano ottenere le credenziali. AWS

Argomenti

- Gestione degli utenti nel tuo bacino d'utenza
- Aggiornamento della configurazione del pool di utenti e del client dell'app

Gestione degli utenti nel tuo bacino d'utenza

Dopo che hai creato un pool di utenti, puoi creare, confermare e gestire gli account utente. Con i gruppi di bacini d'utenza di Amazon Cognito, è possibile gestire gli utenti e il loro accesso alle risorse mappando i ruoli IAM ai gruppi.

La gestione degli utenti nel pool di utenti di Amazon Cognito implica una varietà di opzioni di configurazione e attività amministrative. I pool di utenti possono scalare fino a milioni di utenti. Un elenco utenti di queste dimensioni richiede strumenti amministrativi altrettanto scalabili e ripetibili. Potresti voler creare molti profili utente, gestire gli utenti inattivi, produrre report di governance e conformità o configurare strumenti self-service dove gli utenti svolgono la maggior parte del lavoro. Dopo aver creato un pool di utenti, puoi controllare il modo in cui gli utenti si iscrivono e confermano i propri account, inclusa la richiesta di verifica dell'e-mail o del numero di telefono. Gli amministratori possono anche creare direttamente account utente e personalizzare i messaggi di benvenuto e i requisiti relativi alle password.

I pool di utenti dispongono di gruppi di utenti, in cui è possibile gestire l'accesso alle risorse in base all'appartenenza di un utente al gruppo. Puoi assegnare ruoli IAM a questi gruppi per gestire l'accesso Servizi AWS con pool di identità. L'appartenenza al gruppo degli utenti è presente sia nell'ID che nei token di accesso. Con queste informazioni, puoi prendere decisioni sul controllo degli accessi in fase di esecuzione nella tua applicazione o con un motore di policy come Amazon Verified Permissions.

I pool di utenti hanno spesso molti utenti. Ti ritroverai spesso a cercare e aggiornare gli account utente. La console e l'API di Amazon Cognito supportano l'interrogazione degli utenti in base ad

attributi standard come nome utente, e-mail e numero di telefono. Gli amministratori possono anche reimpostare le password, disabilitare gli account e visualizzare la cronologia degli eventi degli utenti.

Per la migrazione dei dati utente esistenti, Amazon Cognito offre la possibilità di importare gli utenti da un file CSV e di utilizzare un trigger Lambda per migrare automaticamente gli utenti al primo accesso. Queste opzioni supportano le transizioni degli utenti da altre directory utente al tuo pool di utenti.

È possibile utilizzare le funzionalità di gestione degli utenti nei pool di utenti per avere un controllo preciso sul ciclo di vita degli utenti e sull'esperienza di autenticazione. La combinazione di registrazione self-service, account creati dall'amministratore, gruppi e strumenti di migrazione rende i pool di utenti di Amazon Cognito una directory utente flessibile.

Argomenti

- Configurazione delle policy per la creazione degli utenti
- Registrazione e conferma degli account utente
- Creazione di account utente come amministratore
- Aggiunta di gruppi a un bacino d'utenza
- Gestione e ricerca degli account utente
- Password, ripristino dell'account e politiche relative alle password
- Importazione di utenti in un bacino d'utenza
- Utilizzo degli attributi utente

Configurazione delle policy per la creazione degli utenti

Il pool di utenti può consentire agli utenti di registrarsi oppure puoi crearli come amministratore. Puoi anche controllare in che misura il processo di verifica e conferma dopo la registrazione è affidato agli utenti. Ad esempio, potresti voler esaminare le iscrizioni e accettarle sulla base di un processo di convalida esterno. Questa configurazione, o policy utente creata dall'amministratore, imposta inoltre il periodo di tempo prima che un utente non possa più confermare il proprio account.

Amazon Cognito può soddisfare le esigenze dei clienti pubblici come piattaforma CIAM (Customer Identity and Access Management) per il software. Un pool di utenti che accetta la registrazione e dispone di un client per l'app, con o senza accesso gestito, crea un profilo utente per chiunque su Internet conosca l'ID client dell'app, visibile pubblicamente, e richieda di registrarsi. Un profilo

utente registrato può ricevere token di accesso e di identità e può accedere alle risorse che hai autorizzato per l'app. Prima di attivare l'iscrizione nel pool di utenti, esamina le opzioni e assicurati che la configurazione sia conforme ai tuoi standard di sicurezza. Imposta Abilita la registrazione self-service e `AllowAdminCreateUserOnly`, come descritto nelle seguenti procedure, con attenzione.

Console di gestione AWS

Il Sign-upmenu del pool di utenti contiene alcune impostazioni per la registrazione e la creazione amministrativa degli utenti del pool di utenti.

Configurazione dell'esperienza di registrazione

1. In Cognito-assisted Verifica e conferma, scegli se consentire a Cognito di inviare automaticamente messaggi per verificare e confermare. Con questa impostazione abilitata, Amazon Cognito invia un'e-mail o un messaggio SMS ai nuovi utenti con un codice da presentare al pool di utenti. Ciò conferma la proprietà dell'indirizzo e-mail o del numero di telefono, impostando l'attributo equivalente come verificato e confermando l'account utente per l'accesso. Gli Attributi da verificare scelti determinano i metodi di consegna e le destinazioni dei messaggi di verifica.
2. Verifica delle modifiche degli attributi non è importante quando si creano utenti, ma è relativa alla verifica degli attributi. Puoi consentire agli utenti che hanno modificato ma non ancora verificato i propri attributi di accesso di continuare ad accedere con il nuovo valore dell'attributo o con quello originale. Per ulteriori informazioni, consulta Verifica in caso di modifica dell'e-mail o del numero di telefono da parte dell'utente.
3. Attributi obbligatori mostra gli attributi a cui è necessario fornire un valore prima che un utente possa registrarsi o creare un utente. Puoi impostare gli attributi obbligatori solo quando crei un pool di utenti.
4. Gli Attributi personalizzati sono importanti per il processo di creazione e registrazione degli utenti perché puoi impostare un valore per gli attributi personalizzati non modificabili solo quando crei un utente per la prima volta. Per ulteriori informazioni sugli attributi personalizzati, consulta Attributi personalizzati.
5. Al Self-service momento dell'iscrizione, seleziona Abilita l'autoregistrazione se desideri che gli utenti siano in grado di generare un nuovo account con l'API non `SignUp` autenticata. Se disabiliti l'autoregistrazione, puoi creare nuovi utenti solo come amministratore, nella console Amazon Cognito o AdminCreateUsercon richieste API. In un pool di utenti in cui l'autoregistrazione è inattiva, le richieste SignUpAPI vengono restituite **`NotAuthorizedException`** e l'accesso gestito non mostra il link di registrazione.

Per i pool di utenti in cui intendi creare utenti come amministratore, puoi configurare la durata delle relative password temporanee nell'impostazione del menu Metodi di autenticazione in Le password temporanee impostate dagli amministratori scadono in.

Un altro elemento importante della creazione di utenti come amministratore è il messaggio di invito. Quando crei un nuovo utente, Amazon Cognito invia un messaggio con un link alla tua app in modo che l'utente possa accedere per la prima volta. Personalizza questo modello di messaggio nel menu Metodi di autenticazione in Modelli di messaggi.

Puoi configurare client di app riservati, in genere applicazioni Web, con un segreto del client che impedisce l'iscrizione senza il segreto del client dell'app. Come best practice consigliata in materia di sicurezza, non distribuire i segreti dei client delle app in client di app pubbliche, in genere app per dispositivi mobili. Puoi creare client di app con segreti client nel menu App client della console Amazon Cognito.

Amazon Cognito user pools API

Puoi impostare a livello di codice i parametri per la creazione di utenti in un pool di utenti in una richiesta CreateUserPoolo UpdateUserPoolAPI.

L'AdminCreateUserConfigelemento imposta i valori per le seguenti proprietà di un pool di utenti.

1. Abilitazione dell'iscrizione self-service
2. Il messaggio di invito che invii ai nuovi utenti creati dall'amministratore

L'esempio seguente, quando viene aggiunto al corpo completo della richiesta API, imposta un pool di utenti con l'iscrizione self-service inattiva e un'e-mail di invito di base.

```
"AdminCreateUserConfig": {
      "AllowAdminCreateUserOnly": true,
      "InviteMessageTemplate": {
         "EmailMessage": "Your username is {username} and temporary password is
 {####}.",
         "EmailSubject": "Welcome to ExampleApp",
         "SMSMessage": "Your username is {username} and temporary password is
 {####}."
      }
   }
```

I seguenti parametri aggiuntivi di una richiesta CreateUserPoolo UpdateUserPoolAPI regolano la creazione di nuovi utenti.

AutoVerifiedAttributes

Gli attributi, gli indirizzi e-mail o i numeri di telefono a cui desideri inviare automaticamente un messaggio quando registri un nuovo utente.

Policy

La policy delle password del pool di utenti.

Schema

Gli attributi personalizzati del pool di utenti. Sono importanti per il processo di creazione e registrazione degli utenti perché puoi impostare un valore per gli attributi personalizzati non modificabili solo quando crei un utente per la prima volta.

Questo parametro imposta anche gli attributi richiesti per il pool di utenti. Il testo seguente, quando viene inserito nell'elemento `Schema` di un corpo di una richiesta API completa, imposta l'attributo `email`come richiesto.

```
{
            "Name": "email",
            "Required": true
}
```

Registrazione e conferma degli account utente

Gli account utente vengono aggiunti al bacino d'utenza in uno dei seguenti modi:

- L'utente effettua la registrazione nell'app client del tuo bacino d'utenza. Può trattarsi di un'app per dispositivi mobili o Web.
- Puoi importare l'account utente nel tuo bacino d'utenza. Per ulteriori informazioni, consulta Importazione di utenti nel bacino d'utenza da un file CSV.
- Puoi creare l'account utente all'interno del bacino d'utenza e invitare l'utente a effettuare l'accesso. Per ulteriori informazioni, consulta Creazione di account utente come amministratore.

Gli utenti che effettuano la registrazione devono essere confermati prima di poter eseguire l'accesso. Gli utenti importati e creati sono già confermati, ma devono creare la propria password quando effettuano l'accesso per la prima volta. Le sezioni seguenti spiegano il processo di conferma e la verifica tramite e-mail o telefono.

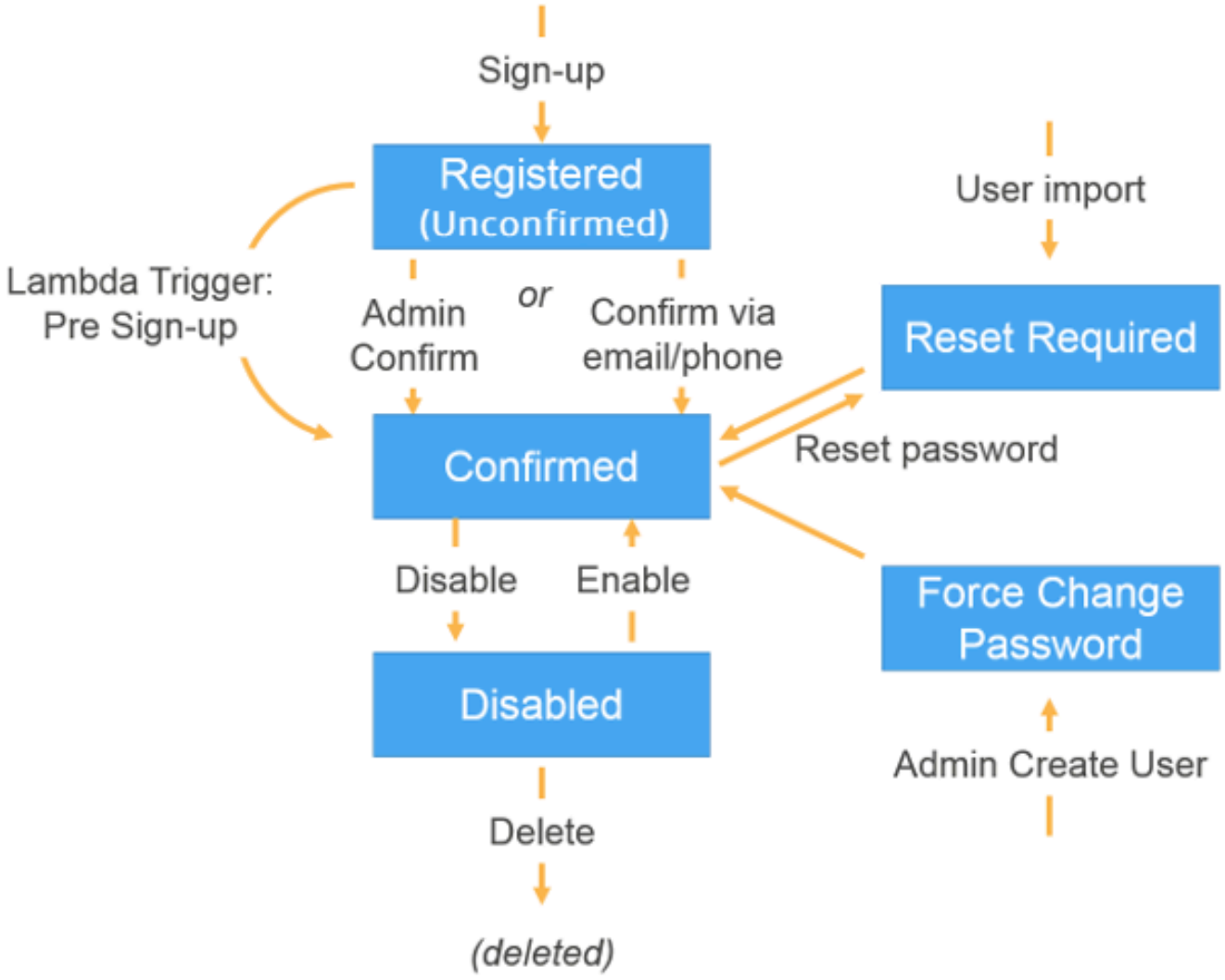
Password al momento dell'iscrizione

Amazon Cognito richiede le password di tutti gli utenti al momento della registrazione, ad eccezione delle seguenti condizioni. Se tutte queste condizioni sono soddisfatte, puoi omettere le password nelle operazioni di registrazione.

1. L'accesso senza password è attivo nel pool di utenti e nel client dell'app.
2. La tua applicazione è personalizzata con moduli di autenticazione in un SDK. AWS L'accesso gestito e l'interfaccia utente ospitata richiedono sempre delle password.
3. Gli utenti forniscono valori di attributo per i metodi di accesso senza password, ad esempio password monouso (OTP) tramite e-mail o SMS, consentiti dall'utente. Ad esempio, se consenti l'accesso tramite e-mail e telefono OTP, gli utenti possono fornire un numero di telefono o un indirizzo e-mail, ma se consenti l'accesso solo tramite e-mail, devono fornire un indirizzo e-mail.
4. Il pool di utenti verifica automaticamente gli attributi che gli utenti possono utilizzare con l'accesso senza password.
5. Per ogni SignUprichiesta specifica, l'utente non fornisce un valore per il parametro Password.

Panoramica sulla conferma dell'account utente

Il seguente diagramma illustra il processo di conferma:



Un account utente può avere differenti stati:

Registrato (non confermato)

L'utente si è registrato correttamente, ma non può effettuare l'accesso fino a quando l'account utente non viene confermato. In questo stato l'utente è abilitato ma non confermato.

I nuovi utenti che effettuano la registrazione iniziano da questo stato.

Confermato

L'account utente viene confermato e l'utente può effettuare l'accesso. Quando un utente inserisce un codice o segue un link e-mail per confermare il proprio account utente, tale e-mail o numero di telefono viene verificato automaticamente. Il codice o il link è valido 24 ore.

Se l'account utente è stato confermato dall'amministratore o da un trigger Lambda di pre-registrazione, potrebbe non esserci un'e-mail verificata o un numero di telefono associato all'account.

Reimpostazione della password obbligatoria

L'account utente viene confermato, ma l'utente deve richiedere un codice e reimpostare la password prima di poter effettuare l'accesso.

Gli account utente importati da un amministratore o uno sviluppatore iniziano da questo stato.

Modifica forzata della password

L'account utente viene verificato e l'utente può accedere utilizzando una password temporanea che dovrà modificare al primo accesso con un nuovo valore prima di fare qualsiasi altra operazione.

Gli account utente creati da un amministratore o uno sviluppatore iniziano da questo stato.

Disabilitato

Prima di poter eliminare un account utente, è necessario disabilitare l'accesso per quest'ultimo.

Altre risorse

- Rilevamento e correzione degli account utente inattivi con Amazon Cognito

Verifica delle informazioni di contatto al momento della registrazione

Quando nuovi utenti effettuano la registrazione nella tua app, è probabile che tu voglia che forniscano almeno un metodo di contatto. Ad esempio, con le informazioni di contatto degli utenti, puoi:

- Inviare una password temporanea quando un utente sceglie di reimpostare la propria password.
- Avvisare gli utenti quando le loro informazioni personali o finanziarie vengono aggiornate.
- Inviare messaggi promozionali, ad esempio offerte speciali o sconti.
- Inviare riepiloghi sugli account o promemoria di fatturazione.

Per i casi d'uso come questi, è importante inviare messaggi a una destinazione verificata. Diversamente potresti inviare i tuoi messaggi a indirizzi e-mail o numeri di telefono non validi non digitati correttamente. Ancora peggio, potresti inviare informazioni riservate a destinatari scaltri che fingono di essere tuoi utenti.

Per essere certo di inviare i messaggi solo alle persone corrette, configura il tuo bacino d'utenza Amazon Cognito in modo che gli utenti debbano fornire le informazioni seguenti al momento della registrazione:

a. Un indirizzo e-mail o numero di telefono.

b. Un codice di verifica che Amazon Cognito invia all'indirizzo e-mail o al numero di telefono. Se sono trascorse 24 ore e il codice o il link dell'utente non è più valido, chiama l'operatore dell'ResendConfirmationCodeAPI per generare e inviare un nuovo codice o collegamento.

Fornendo il codice di verifica, un utente dimostra di avere accesso alla mailbox o al telefono che ha ricevuto il codice. Dopo che l'utente ha fornito il codice, Amazon Cognito aggiorna le informazioni relative all'utente nel bacino d'utenza nel seguente modo:

- Impostando lo stato dell'utente su `CONFIRMED`.
- Aggiornando gli attributi dell'utente per indicare che l'indirizzo e-mail o il numero di telefono sono verificati.

Per visualizzare queste informazioni, puoi utilizzare la console Amazon Cognito. In alternativa, puoi utilizzare l'operazione `AdminGetUser` API, il `admin-get-user` comando con o un'azione corrispondente in uno degli AWS SDK. AWS CLI

Se un utente dispone di un metodo di contatto verificato, Amazon Cognito gli invia automaticamente un messaggio quando richiede un ripristino della password.

Altre azioni che confermano e verificano gli attributi dell'utente

La seguente attività dell'utente verifica gli attributi dell'utente. Non è necessario impostare questi attributi per la verifica automatica: le azioni elencate li contrassegnano come verificati in tutti i casi.

Indirizzo e-mail

1. Completamento con successo dell'autenticazione senza password con una password monouso (OTP) via e-mail.
2. Completamento con successo dell'autenticazione a più fattori (MFA) con un'e-mail OTP.

Numero di telefono

1. Completamento con successo dell'autenticazione senza password con un SMS OTP.
2. Completamento con successo della MFA con un SMS OTP.

Per configurare il bacino d'utenza per richiedere la verifica di e-mail o telefono

Quando verifichi gli indirizzi e-mail e i numeri di telefono degli utenti, assicurati di poter contattare questi ultimi. Completa i seguenti passaggi Console di gestione AWS per configurare il tuo pool di utenti in modo che gli utenti confermino i loro indirizzi e-mail o numeri di telefono.

> **Note**
>
> Se non disponi ancora di un pool di utenti nel tuo account, consulta Crea una nuova applicazione nella console Amazon Cognito.

Per configurare il bacino d'utenza

1. Passa alla console di Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Dal riquadro di navigazione, scegli User Pools (Bacini d'utenza). Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli il Sign-upmenu e individua la verifica degli attributi e la conferma dell'account utente. Scegli Modifica.
4. Nella sezione Cognito-assisted Verifica e conferma, scegli se consentire a Cognito di inviare automaticamente messaggi per la verifica e la conferma. Con questa impostazione abilitata, Amazon Cognito invia messaggi agli attributi di contatto utente selezionati quando un utente si iscrive o quando crei un profilo utente. Per verificare gli attributi e confermare i profili utente per l'accesso, Amazon Cognito invia un codice o un link nei messaggi agli utenti. Gli utenti devono quindi inserire il codice nell'interfaccia utente in modo che l'app possa confermarli in una richiesta API `ConfirmSignUp` o `AdminConfirmSignUp`.

   > **Note**
   >
   > Puoi anche disabilitare Cognito-assisted la verifica e la conferma e utilizzare azioni API autenticate o trigger Lambda per verificare gli attributi e confermare gli utenti. Scegliendo questa opzione, Amazon Cognito non invia alcun codice di verifica quando gli utenti effettuano la registrazione. Scegli questa opzione se utilizzi un flusso di autenticazione personalizzato che verifica almeno un metodo di contatto senza usare i codici di verifica di Amazon Cognito. Ad esempio, è possibile utilizzare un trigger Lambda di pre-registrazione che verifica automaticamente gli indirizzi e-mail che appartengono a un determinato dominio.

Se non vengono verificate le informazioni di contatto degli utenti, in alcuni casi essi potrebbero non essere in grado di utilizzare l'app. Si noti che tutti gli utenti necessitano di informazioni di contatto verificate per:

- Reimpostare la password utente — Quando un utente effettua un'operazione nell'app che chiama l'operazione API `ForgotPassword`, Amazon Cognito invia una password temporanea all'indirizzo e-mail o al numero di telefono dell'utente. Amazon Cognito invia questa password solo se l'utente ha almeno un metodo di contatto verificato.
- Accesso tramite indirizzo e-mail o numero di telefono come alias — Configurando il bacino d'utenza per consentire questi alias, un utente sarà in grado di accedere con un alias solo se è verificato. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

5. Scegli gli attributi da verificare:

   Invia un messaggio SMS, verifica il numero di telefono

   Amazon Cognito invia un codice di verifica tramite messaggio SMS all'accesso dell'utente. Scegli questa opzione se in genere comunichi con gli utenti tramite SMS. Ad esempio, si vorranno utilizzare numeri di telefono verificati se si inviano notifiche di consegna, conferme di appuntamenti o avvisi. I numeri di telefono degli utenti sono l'attributo verificato quando gli account vengono confermati; è necessario intraprendere ulteriori azioni per verificare e comunicare con gli indirizzi e-mail degli utenti.

   Invia messaggio e-mail, verifica l'indirizzo email

   Amazon Cognito invia un codice di verifica tramite messaggio SMS all'accesso dell'utente. Scegliere questa opzione se in genere si comunica con gli utenti tramite e-mail. Ad esempio, si vorranno utilizzare indirizzi e-mail verificati se si inviano estratti conto, riepiloghi di ordini o offerte speciali. Gli indirizzi e-mail degli utenti sono l'attributo verificato quando gli account vengono confermati; è necessario intraprendere ulteriori azioni per verificare e comunicare con i numeri di telefono degli utenti.

   Invia messaggio SMS se il numero di telefono è disponibile, altrimenti invia un messaggio e-mail

   Scegliere questa opzione se non si necessita che tutti gli utenti dispongano del medesimo metodo di contatto verificato. In questo caso, la pagina di registrazione nell'app potrebbe richiedere agli utenti di verificare solo il metodo di contatto preferito. Quando Amazon Cognito un codice di verifica, lo invia al metodo di contatto fornito nella richiesta `SignUp` dall'app. Se un utente fornisce sia un indirizzo e-mail che un numero di telefono e l'app offre entrambi i

metodi di contatto nella richiesta `SignUp`, Amazon Cognito invierà il codice di verifica solo al numero di telefono.

Se occorre che gli utenti verifichino sia un indirizzo e-mail che un numero di telefono, scegliere questa opzione. Amazon Cognito verifica un metodo di contatto quando l'utente effettua la registrazione e l'app deve verificare l'altro metodo di contatto dopo che l'utente ha effettuato l'accesso. Per ulteriori informazioni, consulta Se si richiede agli utenti di confermare sia gli indirizzi e-mail che i numeri di telefono.

6. Scegli Salva modifiche.

Flusso di autenticazione con verifica di e-mail o telefono

Se il bacino d'utenza richiede agli utenti di verificare le proprie informazioni di contatto, l'app deve facilitare il seguente flusso quando un utente effettua la registrazione:

1. Un utente si iscrive alla tua app inserendo un nome utente, un numero di telefono, un indirizzo and/or email e possibilmente altri attributi.
2. Il servizio Amazon Cognito riceve la richiesta di registrazione dall'app. Dopo aver verificato che la richiesta contenga tutti gli attributi necessari per la registrazione, il servizio completa il processo di registrazione e invia un codice di conferma al telefono (tramite SMS) o all'indirizzo e-mail dell'utente. Il codice è valido per 24 ore.
3. Il servizio informa l'app che la registrazione è stata completata e che l'account utente è in attesa di conferma. La risposta contiene informazioni su dove è stato inviato il codice di conferma. A questo punto l'account utente è in stato non confermato e l'indirizzo e-mail e il numero di telefono dell'utente non sono verificati.
4. L'app è ora in grado di richiedere all'utente di inserire il codice di conferma. Non è necessario che l'utente inserisca il codice immediatamente. Tuttavia, l'utente non sarà in grado di effettuare l'accesso fino a quando non avrà inserito il codice di conferma.
5. L'utente inserisce il codice di conferma nell'app.
6. L'app chiama `ConfirmSignUp` per inviare il codice al servizio Amazon Cognito, il quale verifica il codice e, se è corretto, imposta l'account utente sullo stato confermato. Dopo aver confermato correttamente l'account utente, il servizio Amazon Cognito contrassegna automaticamente l'attributo utilizzato per la conferma (indirizzo e-mail o numero di telefono) come verificato. A meno che il valore di questo attributo non venga modificato, l'utente non dovrà verificarlo nuovamente.
7. A questo punto l'account utente è in stato confermato e l'utente può effettuare l'accesso.

Se si richiede agli utenti di confermare sia gli indirizzi e-mail che i numeri di telefono

Amazon Cognito verifica solo uno dei metodi di contatto quando un utente effettua la registrazione. Nei casi in cui Amazon Cognito deve scegliere tra la verifica di un indirizzo e-mail o di un numero di telefono, sceglie di verificare il numero di telefono inviando un codice di verifica tramite SMS. Ad esempio, se configuri il tuo bacino d'utenza per consentire agli utenti di verificare gli indirizzi e-mail o i numeri di telefono e se la tua app fornisce entrambi questi attributi al momento della registrazione, Amazon Cognito verifica solo il numero di telefono. Dopo che un utente ha verificato il proprio numero di telefono, Amazon Cognito imposta lo stato dell'utente su `CONFIRMED` e all'utente sarà consentito accedere all'app.

Dopo che l'utente ha effettuato l'accesso, la tua app può fornire l'opzione di verifica del metodo di contatto che non è stato verificato durante la registrazione. Per verificare il secondo metodo, la tua app chiama l'operazione API `VerifyUserAttribute`. Tieni presente che questa operazione richiede un parametro `AccessToken` e che Amazon Cognito fornisce solo i token di accesso agli utenti autenticati. Pertanto, puoi verificare il secondo metodo di contatto solo dopo che l'utente ha effettuato l'accesso.

Se hai bisogno che i tuoi utenti verifichino sia gli indirizzi e-mail che i numeri di telefono, effettua quanto segue:

1. Configurare il bacino d'utenza per consentire agli utenti di verificare l'indirizzo e-mail o il numero di telefono.
2. Nel flusso di registrazione dell'app, richiedere agli utenti di fornire un indirizzo e-mail e un numero di telefono. Chiamare l'operazione API `SignUp` e fornire l'indirizzo e-mail e il numero di telefono per il parametro `UserAttributes`. A questo punto, Amazon Cognito invia un codice di verifica al telefono dell'utente.
3. Nell'interfaccia dell'app, presentare una pagina di conferma in cui l'utente possa immettere il codice di verifica. Confermare l'utente chiamando l'operazione API `ConfirmSignUp`. A questo punto, lo stato dell'utente è `CONFIRMED` e il suo numero di telefono verificato, ma non lo è l'indirizzo e-mail.
4. Presentare la pagina di accesso e autenticare l'utente chiamando l'operazione API `InitiateAuth`. Dopo che l'utente è stato autenticato, Amazon Cognito restituisce un token di accesso all'app.
5. Chiama l'operazione API `GetUserAttributeVerificationCode`. Specificare i seguenti parametri nella richiesta.

- `AccessToken`: il token di accesso restituito da Amazon Cognito quando l'utente ha effettuato l'accesso.
- `AttributeName`: specifica `"email"` come valore dell'attributo.

Amazon Cognito invia un codice di verifica all'indirizzo e-mail dell'utente.

6. Presentare una pagina di conferma in cui l'utente possa immettere il codice di verifica. Quando l'utente invia il codice, chiamare l'operazione API `VerifyUserAttribute`. Specificare i seguenti parametri nella richiesta.

   - `AccessToken`: il token di accesso restituito da Amazon Cognito quando l'utente ha effettuato l'accesso.
   - `AttributeName`: specifica `"email"` come valore dell'attributo.
   - `Code`: il codice di verifica fornito dall'utente.

   A questo punto, l'indirizzo e-mail è verificato.

Permettere agli utenti di registrarsi ma confermarli come un amministratore del pool di utenti

Potrebbe essere necessario impedire al pool di utenti di inviare automaticamente messaggi di verifica nel pool di utenti, ma consentire comunque a chiunque di registrarsi per un account. Questo modello lascia spazio, ad esempio, alla revisione umana delle nuove richieste di registrazione e alla convalida ed elaborazione batch delle registrazioni. Puoi confermare nuovi account utente nella console Amazon Cognito o con il funzionamento dell' IAM-authenticated API. AdminConfirmSignUp Puoi confermare gli account utente come un amministratore a prescindere che il pool di utenti invii o meno messaggi di verifica.

Con questa tecnica, puoi solo confermare l'iscrizione self-service di un utente. Per confermare un utente creato come amministratore, crea una richiesta AdminSetUserPasswordAPI con `Permanent` set to`True`.

1. Un utente si iscrive alla tua app inserendo un nome utente, un numero di telefono, un indirizzo and/or email e possibilmente altri attributi.
2. Il servizio Amazon Cognito riceve la richiesta di registrazione dall'app. Dopo aver verificato che la richiesta contenga tutti gli attributi necessari per la registrazione, il servizio completa il

processo di registrazione e indica all'app che la registrazione è stata completata ed è in attesa di conferma. A questo punto l'account utente è in stato non confermato. L'utente non può effettuare l'accesso fino a quando l'account non è confermato.

3. Conferma l'account dell'utente. È necessario accedere Console di gestione AWS o firmare la richiesta API con AWS le credenziali per confermare l'account.

   a. Per confermare un utente nella console Amazon Cognito, accedi al menu Utenti, scegli l'utente che desideri confermare e dal menu Azioni seleziona Conferma.

   b. Per confermare un utente nell' AWS API o nella CLI, crea una richiesta AdminConfirmSignUpAPI o admin-confirm-sign-up in. AWS CLI

4. A questo punto l'account utente è in stato confermato e l'utente può effettuare l'accesso.

Calcolo dei valori SecretHash

Assegnare un segreto del cliente al client dell'app riservata come una best practice. Quando si assegna un segreto del client al client dell'app, le richieste API del pool di utenti di Amazon Cognito devono includere un hash che includa il segreto del client nel corpo della richiesta. Per convalidare la conoscenza del segreto del client per le operazioni API nei seguenti elenchi, concatenare il segreto del client con l'ID del client dell'app e il nome dell'utente, quindi codificare in base64 tale stringa.

Quando l'app concede l'accesso agli utenti a un client con un hash segreto, puoi utilizzare il valore di qualsiasi attributo di accesso al pool di utenti come elemento nome utente dell'hash segreto. Quando l'app richiede nuovi token in un'operazione di autenticazione con `REFRESH_TOKEN_AUTH`, il valore dell'elemento nome utente dipende dagli attributi di accesso. Se il pool di utenti non ha `username` come attributo di accesso, imposta il valore hash segreto del valore nome utente della richiesta `sub` dell'utente dal token di accesso o token ID. Quando `username` è un attributo di accesso, imposta il valore hash segreto del nome utente dalla richiesta `username`.

Le seguenti API dei pool di utenti di Amazon Cognito accettano un valore hash del segreto del client in un parametro `SecretHash`.

- ConfirmForgotPassword
- ConfirmSignUp
- ForgotPassword
- ResendConfirmationCode
- SignUp

Inoltre, le seguenti API accettano un valore hash del segreto del client in un parametro `SECRET_HASH`, nei parametri di autenticazione o in una risposta di richiesta di verifica.

Operazione API	Parametro principale per SECRET_HASH
InitiateAuth	AuthParameters
AdminInitiateAuth	AuthParameters
RespondToAuthChallenge	ChallengeResponses
AdminRespondToAuthChallenge	ChallengeResponses

Il valore hash del segreto è un codice di autenticazione dei messaggi mediante algoritmi hash con chiave (HMAC) codificato in Base64 e calcolato utilizzando la chiave segreta del client di un pool di utenti e nome utente, oltre all'ID client nel messaggio. Il seguente pseudocodice mostra come viene calcolato questo valore. In questo pseudocodice, + indica la concatenazione, `HMAC_SHA256` rappresenta una funzione che produce un valore HMAC utilizzando HMACSHA256 e rappresenta una funzione che produce la versione dell'output hash. `Base64` Base-64-encoded

```
Base64 ( HMAC_SHA256 ( "Client Secret Key", "Username" + "Client Id" ) )
```

Per una panoramica dettagliata su come calcolare e utilizzare il `SecretHash` parametro, consulta Come posso risolvere gli errori «Impossibile verificare l'hash segreto per il client» dall'API dei miei pool di utenti di Amazon Cognito<client-id>? nel Knowledge Center. AWS

Puoi utilizzare i seguenti esempi di codice nel codice dell'app lato server.

Shell

```
echo -n "[username][app client ID]" | openssl dgst -sha256 -hmac [app client secret]
 -binary | openssl enc -base64
```

Java

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
```

```
public static String calculateSecretHash(String userPoolClientId, String
 userPoolClientSecret, String userName) {
    final String HMAC_SHA256_ALGORITHM = "HmacSHA256";

    SecretKeySpec signingKey = new SecretKeySpec(
            userPoolClientSecret.getBytes(StandardCharsets.UTF_8),
            HMAC_SHA256_ALGORITHM);
    try {
        Mac mac = Mac.getInstance(HMAC_SHA256_ALGORITHM);
        mac.init(signingKey);
        mac.update(userName.getBytes(StandardCharsets.UTF_8));
        byte[] rawHmac =
 mac.doFinal(userPoolClientId.getBytes(StandardCharsets.UTF_8));
        return Base64.getEncoder().encodeToString(rawHmac);
    } catch (Exception e) {
        throw new RuntimeException("Error while calculating ");
    }
}
```

Python

```
import sys
import hmac, hashlib, base64
username = sys.argv[1]
app_client_id = sys.argv[2]
key = sys.argv[3]
message = bytes(sys.argv[1]+sys.argv[2],'utf-8')
key = bytes(sys.argv[3],'utf-8')
secret_hash = base64.b64encode(hmac.new(key, message,
 digestmod=hashlib.sha256).digest()).decode()
print("SECRET HASH:",secret_hash)
```

Conferma degli account utente senza la verifica dell'e-mail o del numero di telefono

Il trigger Lambda di pre-registrazione può essere utilizzato per confermare automaticamente gli account utente al momento della registrazione, senza richiedere un codice di conferma e verificare l'e-mail o il numero di telefono. Gli utenti che si confermano in questo modo possono effettuare l'accesso immediatamente senza la necessità di ricevere un codice.

Tramite questo trigger, puoi anche contrassegnare l'e-mail o il numero di telefono di un utente come verificati.

> **Note**
>
> Sebbene questo approccio sia utile per gli utenti che stanno appena cominciando, ti consigliamo di auto verificare almeno l'e-mail o il numero di telefono. In caso contrario, può accadere che l'utente non sia in grado di recuperare la password nel caso la dimentichi.

Se non richiedi che l'utente riceva e inserisca un codice di conferma durante la registrazione e non si esegue la verifica automatica di indirizzo e-mail e numero di telefono nel trigger Lambda di pre-registrazione, rischi di non avere un indirizzo e-mail o un numero di telefono verificato per quell'account utente. L'utente può verificare l'indirizzo e-mail o il numero di telefono in un secondo momento. Tuttavia, se l'utente dimentica la password e non dispone di un indirizzo e-mail o di un numero di telefono verificato, rimane bloccato fuori dall'account, in quanto il flusso di password dimenticata richiede un indirizzo e-mail o un numero di telefono verificato per inviare all'utente un codice di verifica.

Verifica in caso di modifica dell'e-mail o del numero di telefono da parte dell'utente

Nei pool di utenti configurati con più nomi di accesso, gli utenti possono inserire un numero di telefono o un indirizzo e-mail come nome utente al momento dell'accesso. Quando aggiornano il loro indirizzo e-mail o il numero di telefono nella tua app, Amazon Cognito può inviare loro immediatamente un messaggio con un codice che verifica la loro proprietà del nuovo valore dell'attributo. Per abilitare l'invio automatico di questi codici di verifica, consulta. Configurazione della verifica di e-mail o telefono

Gli utenti che ricevono un codice di verifica devono fornire tale codice ad Amazon Cognito in una VerifyUserAttributerichiesta. Dopo aver fornito il codice, il relativo attributo viene contrassegnato come verificato. In genere, quando gli utenti aggiornano l'indirizzo e-mail o il numero di telefono, è necessario verificare che siano proprietari del nuovo valore prima che possano utilizzarlo per accedere e ricevere messaggi. I pool di utenti dispongono di un'opzione configurabile che determina se gli utenti devono verificare gli aggiornamenti del proprio indirizzo e-mail o numero di telefono.

Questa opzione è la proprietà `AttributesRequireVerificationBeforeUpdate` del pool di utenti. Configuralo in una UpdateUserPoolrichiesta CreateUserPoolo o con l'impostazione Mantieni attivo il valore dell'attributo originale quando è in sospeso un aggiornamento nel Sign-upmenu della console Amazon Cognito.

Il modo in cui il tuo pool di utenti tratta gli aggiornamenti degli indirizzi e-mail e dei numeri di telefono dipende dalla configurazione del nome utente del tuo pool di utenti. I nomi utente del pool di utenti

possono essere in una configurazione di attributi di nome utente in cui i nomi di accesso sono indirizzo e-mail, numero di telefono o entrambi. Possono anche essere in una configurazione di attributi di alias in cui l'`username`attributo è un nome di accesso insieme all'indirizzo e-mail, al numero di telefono o al nome utente preferito come nomi di accesso alternativi. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

Puoi anche utilizzare un trigger Lambda per messaggi personalizzati per personalizzare il messaggio di verifica. Per ulteriori informazioni, consulta Trigger Lambda di messaggi personalizzati. Quando l'indirizzo e-mail o il numero di telefono di un utente non sono verificati, l'applicazione deve informare l'utente che deve verificare l'attributo e fornire un pulsante o un collegamento per consentire agli utenti di inserire il codice di verifica.

La tabella seguente descrive in che modo `AttributesRequireVerificationBeforeUpdate` le impostazioni degli alias determinano il risultato quando gli utenti modificano il valore dei propri attributi di accesso.

Configurazione del nome utente	Comportamento quando gli utenti devono verificare nuovi attributi	Comportamento quando agli utenti non è richiesto di verificare nuovi attributi
Attributi nome utente	L'attributo originale rimane verificato, idoneo per l'accesso e al valore originale. Quando l'utente verifica un nuovo valore, Amazon Cognito aggiorna il valore dell'attr ibuto, lo contrassegna come verificato e lo rende idoneo per l'accesso.	Amazon Cognito aggiorna l'attributo a un nuovo valore. Il nuovo valore è idoneo per l'accesso. Quando l'utente verifica un nuovo valore, Amazon Cognito lo contrasse gna come verificato.
Attributi alias	L'attributo originale rimane verificato, idoneo per l'accesso e al valore originale. Quando l'utente verifica un nuovo valore, Amazon Cognito aggiorna il valore dell'attr ibuto, lo contrassegna come	Amazon Cognito aggiorna l'attributo a un nuovo valore. Né il valore originale né quello nuovo dell'attributo sono idonei per l'accesso. Quando l'utente verifica un nuovo valore, Amazon Cognito aggiorna il valore dell'attr

Configurazione del nome utente	Comportamento quando gli utenti devono verificare nuovi attributi	Comportamento quando agli utenti non è richiesto di verificare nuovi attributi
	verificato e lo rende idoneo per l'accesso.	ibuto, lo contrassegna come verificato e lo rende idoneo per l'accesso.

Esempio 1

L'utente 1 accede all'applicazione con l'indirizzo e-mail `user1@example.com` e dispone del nome utente `user1` (attributi alias). Il tuo pool di utenti è configurato per verificare gli aggiornamenti degli attributi di accesso e per inviare automaticamente messaggi di verifica. Richiedono di aggiornare il proprio indirizzo e-mail a`user1+foo@example.com`. Ricevono un'e-mail di verifica all'indirizzo `user1+foo@example.com` e possono accedere nuovamente solo con l'indirizzo e-mail`user1@example.com`. Successivamente, inseriscono il codice di verifica e possono accedere nuovamente solo con l'indirizzo e-mail`user1+foo@example.com`.

Esempio 2

L'utente 2 accede all'applicazione con l'indirizzo e-mail `user2@example.com` e dispone di un nome utente (attributi alias). Il tuo pool di utenti è configurato per non verificare gli aggiornamenti degli attributi di accesso e per inviare automaticamente messaggi di verifica. Richiedono di aggiornare il proprio indirizzo e-mail a`user2+bar@example.com`. Ricevono un'email di verifica all'indirizzo `user2+bar@example.com` e non possono accedere nuovamente. Successivamente, inseriscono il codice di verifica e possono accedere nuovamente solo con l'indirizzo e-mail`user2+bar@example.com`.

Esempio 3

L'utente 3 accede all'applicazione con l'indirizzo e-mail `user3@example.com` e non dispone di un nome utente (attributi del nome utente). Il tuo pool di utenti è configurato per non verificare gli aggiornamenti degli attributi di accesso e per inviare automaticamente messaggi di verifica. Richiedono di aggiornare il proprio indirizzo e-mail a`user3+baz@example.com`. Ricevono un'email di verifica all'indirizzo`user3+baz@example.com`, ma possono accedere immediatamente senza che venga intrapresa alcuna azione aggiuntiva con il codice di verifica.

Processo di conferma e verifica degli account utente creati dagli amministratori o dagli sviluppatori

Gli account utente creati da un amministratore o da uno sviluppatore sono già in stato confermato, per cui agli utenti non viene richiesto di inserire un codice di conferma. Il messaggio di invito che il servizio Amazon Cognito invia a questi utenti include il nome utente e una password temporanea. L'utente deve modificare la password prima di effettuare l'accesso. Per ulteriori informazioni, consulta la Personalizzare e-mail ed SMS di Creazione di account utente come amministratore e il trigger di messaggio personalizzato in Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda.

Processi di conferma e verifica per gli account utente importati

Gli account utente creati utilizzando la funzionalità di importazione degli utenti nella CLI o nell'API (vediImportazione di utenti nel bacino d'utenza da un file CSV) sono già nello stato confermato, quindi agli utenti non è richiesto di inserire un codice di conferma. Console di gestione AWS Non viene inviato alcun messaggio di invito. Tuttavia, gli account utente importati prevedono che gli utenti richiedano prima un codice chiamando l'API `ForgotPassword` e che in seguito creino una password utilizzando il codice inviato chiamando l'API `ConfirmForgotPassword` prima di effettuare l'accesso. Per ulteriori informazioni, consulta Necessità degli utenti importati di ripristinare le password.

L'e-mail o il numero di telefono devono essere contrassegnati come verificati quando l'account utente viene importato, in modo che non venga richiesta la verifica quando l'utente effettua l'accesso.

Invio di e-mail durante il test dell'App

Amazon Cognito invia agli utenti messaggi e-mail quando creano e gestiscono i propri account nell'app client per il pool di utenti. Se configuri il bacino d'utenza per richiedere la verifica e-mail, Amazon Cognito invia un'e-mail quando:

- Un utente effettua la registrazione.
- Un utente aggiorna il suo indirizzo e-mail.
- Un utente esegue un'operazione che chiama l'operazione API `ForgotPassword`.
- Crei un account utente come amministratore.

A seconda dell'azione che attiva l'e-mail, l'e-mail contiene un codice di verifica o una password temporanea. I tuoi utenti devono ricevere queste e-mail e comprendere il messaggio. In caso contrario, potrebbero non essere in grado di effettuare l'accesso e utilizzare la tua app.

Per essere che le e-mail vengano inviate senza problemi e che il messaggio sia corretto, prova le operazioni nella tua app attivando le consegne e-mail da Amazon Cognito. Ad esempio, utilizzando la pagina di registrazione nella tua app oppure utilizzando l'operazione API `SignUp`, puoi attivare un'e-mail effettuando la registrazione con un indirizzo e-mail di prova. Quando esegui il testing in questo modo, ricorda quanto segue:

> **Importante**
>
> Quando utilizzi un indirizzo e-mail per testare le operazioni che attivano le e-mail da Amazon Cognito, non utilizzare un indirizzo e-mail fittizio (ossia, senza una casella di posta associata). Utilizza un indirizzo e-mail reale che riceverà l'e-mail da Amazon Cognito senza creare un mancato recapito permanente.
> Un mancato recapito permanente si verifica quando Amazon Cognito non è in grado di consegnare le e-mail alla casella di posta del destinatario, cosa che accade sempre se la casella di posta è inesistente.
> Amazon Cognito limita il numero di e-mail che possono essere inviate da AWS account che subiscono ripetutamente hard bounce.

Quando provi le operazioni che attivano le e-mail, utilizza uno dei seguenti indirizzi e-mail per evitare gli hard bounce:

- Un indirizzo per un account e-mail di tua proprietà che utilizzi per il testing. Quando utilizzi il tuo indirizzo e-mail, riceverai un'e-mail inviata da Amazon Cognito. Con questa e-mail, puoi utilizzare il codice di verifica per testare l'esperienza di registrazione nella tua app. Se hai personalizzato il messaggio e-mail per il tuo bacino d'utenza, puoi controllare che le personalizzazioni siano corrette.
- L'indirizzo del simulatore di mailbox, success@simulator.amazonses.com. Se utilizzi l'indirizzo del simulatore, Amazon Cognito invia correttamente l'e-mail ma non sarai in grado di visualizzarla. Questa opzione è utile quando non è necessario utilizzare il codice di verifica né verificare il messaggio e-mail.
- L'indirizzo del simulatore di mailbox con un'etichetta arbitraria aggiuntiva, come success+user1@simulator.amazonses.com o success+user2@simulator.amazonses.com. Amazon

Cognito invia correttamente e-mail a questi indirizzi ma non sarai in grado di visualizzarle. Questa opzione è utile quando si desidera testare il processo di registrazione aggiungendo più utenti di prova al bacino d'utenza e ogni utente di prova ha un indirizzo e-mail univoco.

Configurazione della verifica di e-mail o telefono

Puoi scegliere le impostazioni per la verifica tramite e-mail o telefono nel menu Metodi di autenticazione. Per ulteriori informazioni sull'autenticazione a più fattori (Multi-Factor Authentication, MFA), consulta MFA con SMS.

Amazon Cognito utilizza Amazon SNS per inviare messaggi SMS. Se non hai mai inviato un messaggio SMS da Amazon Cognito o da altri in Servizio AWS precedenza, Amazon SNS potrebbe inserire il tuo account nella sandbox SMS. Ti consigliamo di inviare un messaggio di prova a un numero di telefono verificato prima di rimuovere il tuo account dalla sandbox e spostarlo alla produzione. Inoltre, se intendi inviare messaggi SMS a numeri di telefono di destinazione negli Stati Uniti, devi ottenere un ID mittente o di origine da Amazon Pinpoint. Per configurare il pool di utenti Amazon Cognito per i messaggi SMS, consulta Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.

Amazon Cognito è in grado di verificare automaticamente gli indirizzi e-mail e i numeri di telefono. Per effettuare questa verifica, Amazon Cognito invia un codice di verifica oppure un link di verifica. Per gli indirizzi e-mail, Amazon Cognito invia il codice o il link tramite un messaggio e-mail. Puoi scegliere un tipo di codice o link di verifica quando modifichi il modello di messaggio di verifica nel menu Modelli di messaggio nella console Amazon Cognito. Per ulteriori informazioni, consulta Personalizzazione dei messaggi di verifica delle e-mail.

Per i numeri di telefono, Amazon Cognito invia il codice con un SMS.

Amazon Cognito deve verificare un numero di telefono o un indirizzo e-mail per confermare gli utenti e aiutarli a recuperare le password dimenticate. In alternativa, puoi confermare automaticamente gli utenti con il trigger Lambda di pre-registrazione o utilizzare AdminConfirmSignUpl'operazione API. Per ulteriori informazioni, consulta Registrazione e conferma degli account utente.

Il codice o link di verifica è valido 24 ore.

Se scegli di richiedere la verifica per un indirizzo e-mail o un numero di telefono, Amazon Cognito invia automaticamente il codice di verifica o il link quando un utente effettua la registrazione. Se il bacino d'utenza ha un Trigger Lambda del mittente di SMS personalizzato o Trigger Lambda del mittente di e-mail personalizzato configurato, tale funzione viene invece richiamata.

Note

- L'utilizzo degli SMS per verificare i numeri di telefono viene addebitato separatamente da Amazon SNS. Per l'invio di messaggi e-mail, non viene addebitato alcun costo. Per le informazioni sui prezzi Amazon SNS, consulta la pagina Prezzi Worldwide SMS. Per la lista dei paesi in cui è disponibile la messaggistica SMS, consulta la pagina Supported regions and countries (regioni e paesi supportati).
- Quando esegui il test delle operazioni nella tua app per la generazione di messaggi e-mail provenienti da Amazon Cognito, utilizza un indirizzo e-mail reale a cui Amazon Cognito può inviare messaggi senza incorrere in un mancato recapito permanente. Per ulteriori informazioni, consulta the section called “Invio di e-mail durante il test dell'App”.
- Il flusso della reimpostazione della password richiede l'indirizzo e-mail o il numero di telefono dell'utente per verificarne l'identità.

Important

Se un utente si registra sia con un numero di telefono sia con un indirizzo e-mail e le impostazioni del pool di utenti richiedono la verifica di entrambi gli attributi, Amazon Cognito invia un codice di verifica tramite SMS al numero di cellulare. Amazon Cognito non ha ancora verificato l'indirizzo e-mail, quindi l'app deve chiamare GetUserper vedere se un indirizzo e-mail è in attesa di verifica. Se richiede una verifica, l'app deve chiamare GetUserAttributeVerificationCodeper avviare il flusso di verifica dell'e-mail. Quindi deve inviare il codice di verifica VerifyUserAttributechiamando.

Puoi modificare la quota di spesa per i messaggi SMS per uno Account AWS e per singoli messaggi. I limiti si applicano solo al costo di invio di messaggi SMS. Per ulteriori informazioni, vedi Cosa sono le quote di spesa a livello di account e di messaggio e come funzionano? in Amazon SNS FAQs.

Amazon Cognito invia messaggi SMS utilizzando le risorse Amazon SNS nel luogo in cui è stato creato Regione AWS il pool di utenti o in una regione alternativa legacy di Amazon SNS, come indicato nella tabella seguente. L'eccezione è rappresentata dai bacini d'utenza Amazon Cognito nella regione Asia Pacifico (Seoul). Questi bacini d'utenza utilizzano la tua configurazione Amazon SNS nella regione Asia Pacifico (Tokyo). Per ulteriori informazioni, consulta Seleziona Regione AWS per i messaggi SMS.

Regione di Amazon Cognito	Regione alternativa Amazon SNS legacy
Stati Uniti orientali (Ohio)	Stati Uniti orientali (Virginia settentrionale)
Asia Pacifico (Mumbai)	Asia Pacifico (Singapore)
Asia Pacifico (Seul)	Asia Pacifico (Tokyo)
Canada (Centrale)	Stati Uniti orientali (Virginia settentrionale)
Europa (Francoforte)	Europa (Irlanda)
Europa (Londra)	Europa (Irlanda)

Esempio: se il tuo bacino d'utenza Amazon Cognito si trova in Asia Pacifico (Mumbai) e hai aumentato il tuo limite di spesa in ap-southeast-1, forse non ti conviene richiedere un aumento separato in ap-south-1. Piuttosto, puoi utilizzare le tue risorse Amazon SNS in Asia Pacifico (Singapore).

Verifica degli aggiornamenti a indirizzi e-mail e numeri di telefono

Un attributo di indirizzo e-mail o un numero di telefono può diventare attivo e non verificato subito dopo che l'utente ha modificato il relativo valore. Amazon Cognito può anche richiedere che l'utente verifichi il nuovo valore prima che Amazon Cognito aggiorni l'attributo. Quando è necessario che gli utenti verifichino innanzitutto il nuovo valore, possono utilizzare il valore originale per l'accesso e per ricevere messaggi fino a quando non verificheranno il nuovo valore.

Quando gli utenti possono utilizzare il proprio indirizzo e-mail o numero di telefono come alias di accesso nel pool di utenti, il nome di accesso per un attributo aggiornato dipende dall'eventuale richiesta di verifica degli attributi aggiornati. Quando si richiede che gli utenti verifichino un attributo aggiornato, un utente può accedere con il valore originale dell'attributo fino a quando non verifica il nuovo valore. Quando non si richiede che gli utenti verifichino un attributo aggiornato, un utente non può accedere o ricevere messaggi con il valore nuovo o originale fino a quando non verifica il nuovo valore.

Ad esempio, il pool di utenti consente l'accesso con un alias di indirizzo e-mail e richiede che gli utenti verifichino il proprio indirizzo e-mail al momento dell'aggiornamento. Sue, che ha effettuato l'accesso come `sue@example.com`, vuole cambiare il suo indirizzo email in `sue2@example.com` ma per errore immette **`s`**`sue2@example.com`. Sue non riceve l'e-mail di verifica, quindi non può

verificare `ssue2@example.com`. Sue accede come `sue@example.com` e invia nuovamente il modulo nell'app per aggiornare il suo indirizzo e-mail in `sue2@example.com`. Riceve il messaggio e-mail, fornisce il codice di verifica all'app e inizia ad accedere come `sue2@example.com`.

Quando un utente aggiorna un attributo e il pool di utenti verifica i nuovi valori degli attributi

- Possono accedere con il valore dell'attributo originale prima di aver confermato il codice per verificare il nuovo valore.
- Possono accedere solo con il valore dell'attributo originale dopo aver confermato il codice per verificare il nuovo valore.
- Se `phone_number_verified` imposti `email_verified` o `true` abiliti una richiesta AdminUpdateUserAttributesAPI, possono accedere prima di aver confermato il codice che Amazon Cognito gli ha inviato.

Quando un utente aggiorna un attributo e il pool di utenti non verifica i nuovi valori degli attributi

- Non possono accedere né ricevere messaggi utilizzando il valore dell'attributo originale.
- Non possono accedere né ricevere messaggi diversi da un codice di conferma utilizzando il nuovo valore dell'attributo prima di aver confermato il codice per verificare il nuovo valore.
- Se `phone_number_verified` imposti `email_verified` o `true` abiliti una richiesta AdminUpdateUserAttributesAPI, possono accedere prima di aver confermato il codice che Amazon Cognito gli ha inviato.

Richiedere la verifica dell'attributo quando gli utenti aggiornano l'indirizzo e-mail o numero di telefono

1. Accedi alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Nel menu Registrazione, scegli Modifica sotto Verifica degli attributi e conferma dell'account utente.
4. Scegli Keep original attribute value active when an update is pending (Mantieni attivo il valore dell'attributo originale quando un aggiornamento è in sospeso).
5. In Active attribute values when an update is pending (Valori di attributo attivi quando un aggiornamento è in sospeso), scegli gli attributi che gli utenti devono di verificare prima che Amazon Cognito aggiorni il valore.

6. Scegli Save changes (Salva modifiche).

Per richiedere la verifica dell'aggiornamento degli attributi con l'API Amazon Cognito, puoi impostare il `AttributesRequireVerificationBeforeUpdate` parametro in una UpdateUserPoolrichiesta.

Autorizzazione di Amazon Cognito a inviare SMS per tuo conto

Per poter inviare SMS agli utenti per tuo conto, Amazon Cognito ha bisogno della tua autorizzazione. Per concedere tale autorizzazione, puoi creare un ruolo AWS Identity and Access Management (IAM). Nel menu Metodi di autenticazione della console Amazon Cognito in SMS, scegli Modifica per impostare un ruolo.

Configurazione MFA, autenticazione, verifica e messaggi di invito

Con Amazon Cognito, puoi personalizzare l'autenticazione, la verifica e i messaggi di invito degli utenti tramite SMS ed e-mail per migliorare la sicurezza e l'esperienza utente della tua applicazione. Per alcuni messaggi puoi scegliere tra verifiche basate su codice e verifiche dei link con un clic. Questo argomento spiega come personalizzare le comunicazioni di autenticazione e verifica nella console Amazon Cognito.

Nel menu Modelli di messaggi, puoi personalizzare:

- I tuoi modelli di messaggi di posta elettronica e SMS per l'autenticazione con password monouso (OTP) e a più fattori (MFA)
- I messaggi di verifica degli SMS e delle e-mail
- Il tipo di verifica dell'e-mail, ovvero codice o link

> **Note**
>
> Amazon Cognito invia link con il tuo modello basato su link nei messaggi di verifica quando gli utenti si iscrivono o inviano nuovamente un codice di conferma. Le e-mail relative alle operazioni di aggiornamento degli attributi e di reimpostazione della password utilizzano il modello di codice.

- I messaggi di invito degli utenti
- FROM e indirizzi REPLY-TO e-mail per le e-mail che passano attraverso il tuo pool di utenti

> **Note**
>
> I modelli di messaggi di verifica via SMS ed e-mail vengono visualizzati solo se hai scelto di richiedere la verifica del numero di telefono e dell'e-mail. Analogamente, il modello di messaggio MFA SMS viene visualizzato solo se le impostazioni MFA sono obbligatorie o facoltative.

Argomenti

- Modelli dei messaggi
- Personalizzazione dei messaggi MFA per e-mail e SMS
- Personalizzazione dei messaggi di verifica delle e-mail
- Personalizzazione dei messaggi di invito degli utenti
- Personalizzazione dell'indirizzo e-mail
- Autorizzazione di Amazon Cognito a inviare e-mail di Amazon SES per tuo conto (da un indirizzo e-mail FROM personalizzato)

Modelli dei messaggi

Puoi utilizzare i modelli di messaggio per inserire segnaposti nei tuoi messaggi. Amazon Cognito sostituisce i segnaposto con i valori corrispondenti. Puoi fare riferimento ai segnaposto Universal Template nei modelli di messaggio di qualsiasi tipo, anche se questi valori non saranno presenti in tutti i tipi di messaggio.

Segnaposto universali modello

Description	Token	Tipo di messaggio
Codice di verifica	`{####}`	Messaggi di verifica, conferma e MFA
Password temporanea	`{####}`	Forgot-password e messaggi di invito
Nome utente	`{username}`	Invito e messaggi di sicurezza avanzati

Una delle risposte automatiche disponibili con protezione dalle minacce consiste nel notificare all'utente che Amazon Cognito ha rilevato attività potenzialmente dannose. Puoi utilizzare i segnaposti dei modelli di sicurezza avanzata per:

- Includere dettagli specifici su un evento come l'indirizzo IP, la città, il paese, l'ora di accesso e il nome del dispositivo. La protezione dalle minacce di Amazon Cognito può analizzare questi dettagli.
- Verificare se un link con un clic è valido.
- Usa l'ID evento, il token di feedback e il nome utente per creare il tuo link personalizzato con un clic.

> **Note**
>
> Per generare link con un clic e utilizzare i segnaposti `{one-click-link-valid}` e `{one-click-link-invalid}` nei modelli di e-mail di sicurezza avanzati, devi disporre già di un dominio configurato per il pool di utenti.

La protezione dalle minacce aggiunge i seguenti segnaposto che puoi inserire nei modelli di messaggio. Questi segnaposto si applicano ai messaggi di autenticazione adattiva, notifiche che Amazon Cognito invia agli utenti le cui sessioni sono state valutate per un livello di rischio. Per configurare i modelli di messaggio con queste variabili, aggiorna la Full-functionconfigurazione della protezione dalle minacce nella console Amazon Cognito o invia i modelli in una SetRiskConfigurationrichiesta.

Segnaposti dei modelli di sicurezza avanzata

Description	Token
IP address (Indirizzo IP)	`{ip-address}`
City	`{city}`
Paese	`{country}`
Log-in tempo	`{login-time}`
Nome dispositivo	`{device-name}`

Description	Token
One-click il link è valido	`{one-click-link-valid}`
One-click il link non è valido	`{one-click-link-invalid}`
ID evento	`{event-id}`
Token del feedback	`{feedback-token}`

Personalizzazione dei messaggi MFA per e-mail e SMS

Per personalizzare gli SMS e i messaggi e-mail per l'autenticazione a più fattori (MFA), modifica il messaggio MFA dal menu Modelli di messaggi nella console dei pool di utenti di Amazon Cognito.

> **Important**
>
> Il tuo messaggio personalizzato deve contenere il segnaposto `{####}`. Questo segnaposto verrà sostituito con il codice di autenticazione al momento dell'invio del messaggio.

Amazon Cognito imposta una lunghezza massima per i messaggi SMS, incluso il codice di autenticazione, di 140 UTF-8 caratteri.

Personalizzazione dei messaggi di verifica degli SMS

Per personalizzare il messaggio SMS per la verifica del numero di telefono, modifica il modello di messaggio di verifica dal menu Modelli di messaggi del tuo pool di utenti.

> **Important**
>
> Il tuo messaggio personalizzato deve contenere il segnaposto `{####}`. Questo segnaposto verrà sostituito con il codice di verifica al momento dell'invio del messaggio.

La lunghezza massima del messaggio, incluso il codice di verifica, è di 140 UTF-8 caratteri.

Personalizzazione dei messaggi di verifica delle e-mail

Per verificare l'indirizzo e-mail di un utente nel bacino d'utenza con Amazon Cognito, puoi inviare all'utente un messaggio e-mail con un link selezionabile, oppure un codice da inserire.

Per personalizzare l'oggetto e il contenuto del messaggio per i messaggi di verifica dell'indirizzo e-mail, modifica il modello di messaggio di verifica nel menu Modelli di messaggio del tuo pool di utenti. Puoi scegliere un Tipo di verifica di Codice o Collegamento quando modifichi il modello Messaggi di verifica.

Quando scegli Codice come tipo di verifica, il messaggio personalizzato deve contenere il segnaposto `{####}`. Quando invii il messaggio, il segnaposto viene sostituito con il codice di verifica.

Quando scegli Collegamento come tipo di verifica, il messaggio personalizzato deve includere un segnaposto nel formato `{##Verify Your Email##}`. Puoi modificare la stringa di testo tra i caratteri segnaposto, ad esempio `{##Click here##}`. Il segnaposto viene sostituito da un link di verifica con il testo Verifica il tuo indirizzo e-mail.

Il link per un messaggio di verifica e-mail indirizza l'utente a un URL come nell'esempio seguente.

```
https://<your user pool domain>/confirmUser/?
client_id=abcdefg12345678&user_name=emailtest&confirmation_code=123456
```

La lunghezza massima del messaggio, incluso il codice di verifica (se presente), è di 20.000 UTF-8 caratteri. È possibile utilizzare i tag HTML in questo messaggio per formattare i contenuti.

Personalizzazione dei messaggi di invito degli utenti

Puoi personalizzare il messaggio di invito utente che Amazon Cognito invia ai nuovi utenti tramite SMS o messaggio e-mail modificando il modello Messaggi di invito nel menu Modelli di messaggio.

> **Important**
>
> Il tuo messaggio personalizzato deve contenere i segnaposti `{username}` e `{####}`. Quando Amazon Cognito invia il messaggio di invito, sostituisce questi segnaposto con il nome utente e la password dell'utente.

La lunghezza massima di un messaggio SMS, incluso il codice di verifica, è di 140 caratteri. UTF-8 La lunghezza massima di un messaggio e-mail, incluso il codice di verifica, è di 20.000 UTF-8 caratteri. È possibile utilizzare tag HTML nei messaggi e-mail per formattare il contenuto.

Personalizzazione dell'indirizzo e-mail

Per impostazione predefinita, i messaggi e-mail che Amazon Cognito invia agli utenti nel bacino d'utenza provengono dall'indirizzo no-reply@verificationemail.com. Puoi scegliere di specificare FROM e indirizzi REPLY-TO e-mail personalizzati anziché no-reply@verificationemail.com.

Per personalizzare il FROM e gli indirizzi REPLY-TO e-mail

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli il menu Metodi di autenticazione. Alla voce E-mail, scegli Modifica.
4. Scegli una Regione SES.
5. Scegli un indirizzo e-mail FROM dall'elenco degli indirizzi e-mail verificati con Amazon SES nella Regione SES selezionata. Per utilizzare un indirizzo e-mail proveniente da un dominio verificato, configura le impostazioni e-mail nell' AWS API AWS Command Line Interface o. Per ulteriori informazioni, consulta Verifica indirizzi e-mail e domini in Amazon SES nella Guida per gli sviluppatori di Amazon SES.
6. Scegli un Set di configurazione dall'elenco dei set di configurazione nella Regione SES selezionata.
7. Inserisci un nome mittente FROM cordiale per i tuoi messaggi e-mail nel formato `John Stiles <johnstiles@example.com>`.
8. Per personalizzare l'indirizzo REPLY-TO e-mail, inserisci un indirizzo e-mail valido nel campo dell'indirizzo REPLY-TO e-mail.

Autorizzazione di Amazon Cognito a inviare e-mail di Amazon SES per tuo conto (da un indirizzo e-mail FROM personalizzato)

Puoi configurare Amazon Cognito per inviare e-mail da un indirizzo e-mail FROM personalizzato anziché dal relativo indirizzo predefinito. Per utilizzare un indirizzo personalizzato, devi concedere ad Amazon Cognito l'autorizzazione ad inviare un messaggio e-mail da un'identità verificata di Amazon SES. Nella maggior parte dei casi, per concedere tale autorizzazione crea una policy di autorizzazione all'invio. Per ulteriori informazioni, consulta Utilizzo dell'autorizzazione di invio con Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

Quando configuri un bacino d'utenza per utilizzare Amazon SES per i messaggi e-mail, Amazon Cognito crea il ruolo `AWSServiceRoleForAmazonCognitoIdpEmailService` nel tuo account per concedere l'accesso ad Amazon SES. Non è necessario alcuna policy di autorizzazione per l'invio

quando viene utilizzato il `AWSServiceRoleForAmazonCognitoIdpEmailService` ruolo collegato ai servizi. È necessario aggiungere una policy di autorizzazione per l'invio solo quando si utilizzano entrambe le funzionalità e-mail predefinite nel bacino d'utenza e un'identità Amazon SES verificata come indirizzo FROM.

Per ulteriori informazioni sul ruolo collegato al servizio creato da Amazon Cognito, consulta Utilizzo di ruoli collegati ai servizi per Amazon Cognito.

L'esempio seguente di policy di autorizzazione di invio garantisce ad Amazon Cognito la possibilità limitata di utilizzare un'identità verificata di Amazon SES. Amazon Cognito può inviare e-mail solo quando lo fa per conto sia del bacino d'utenza nella condizione `aws:SourceArn` sia dell'account nella condizione `aws:SourceAccount`. Per altri esempi, consulta Esempi di policy di autorizzazione di invio di Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

> **Note**
>
> In questo esempio, il valore "Sid" è una stringa arbitraria che identifica in modo univoco l'istruzione. Per ulteriori informazioni sulla sintassi della policy, consulta Policy di autorizzazione di invio di Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "stmnt1234567891234",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "email.cognito-idp.amazonaws.com"
                ]
            },
            "Action": [
                "SES:SendEmail",
                "SES:SendRawEmail"
            ],
```

```
            "Resource": "arn:aws:ses:us-
east-1:111122223333:identity/support@example.com",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:cognito-idp:us-
east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        }
    ]
}
```

La console Amazon Cognito aggiunge una politica simile per te quando selezioni un'identità Amazon SES dal menu a discesa. Se utilizzi la CLI o l'API per configurare il bacino d'utenza, è necessario allegare una policy strutturata in modo simile all'esempio precedente alla tua identità di Amazon SES.

Creazione di account utente come amministratore

I pool di utenti non sono solo un elenco di utenti per la gestione delle identità e degli accessi dei clienti (CIAM), in cui chiunque su Internet può registrarsi per creare un profilo utente nell'applicazione. Puoi disabilitare l'iscrizione in modalità self-service. Potresti già conoscere i tuoi clienti e voler ammettere solo quelli che sono stati autorizzati in anticipo. Puoi creare barriere di autenticazione manuale intorno alla tua applicazione con un provider di identità SAML 2.0 o OIDC privato, importando gli utenti, selezionando gli utenti al momento dell'iscrizione o creando utenti con operazioni amministrative sull'API. Il flusso di lavoro per la creazione amministrativa degli utenti può essere programmatico, con il provisioning degli utenti dopo la registrazione in un altro sistema, oppure può essere basato su test o caso per caso nella console Amazon Cognito.

Quando crei utenti come amministratore, Amazon Cognito imposta una password temporanea per loro e invia un messaggio di benvenuto o di invito. Possono seguire il link nel messaggio di invito e accedere per la prima volta, impostare una password e confermare il proprio account. La pagina che segue descrive come creare nuovi utenti e configurare il messaggio di benvenuto. Per ulteriori informazioni sulla creazione di utenti con l'API dei pool di utenti e un AWS SDK o CDK, consulta. AdminCreateUser

> **Gli utenti con password temporanee iniziano con lo stato FORCE_CHANGE_PASSWORD**
>
> Quando crei un utente con una password temporanea, Amazon Cognito inserisce l'utente nello stato. `FORCE_CHANGE_PASSWORD` Un utente con questo stato non può accedere finché non imposta una password permanente in risposta a una `NEW_PASSWORD_REQUIRED` sfida.

Per spostare un utente creato dall'amministratore `CONFIRMED` allo stato senza effettuare il primo accesso, chiama l'operazione AdminSetUserPasswordAPI con set to. `Permanent True` Questa richiesta imposta una password permanente e conferma l'utente.

Dopo aver creato il pool di utenti, puoi creare utenti utilizzando l'API Console di gestione AWS Amazon Cognito AWS Command Line Interface o l'API Amazon Cognito. Puoi creare il profilo di un nuovo utente in un bacino d'utenza e inviargli un messaggio di benvenuto con le istruzioni di registrazione tramite SMS o e-mail.

Di seguito sono riportati alcuni esempi di come gli amministratori possono gestire gli utenti nei pool di utenti.

- Crea un nuovo profilo utente nella console Amazon Cognito o con il funzionamento dell'`AdminCreateUser`API.
- Rendi disponibili i flussi di autenticazione con nome utente e password, senza password, passkey e personalizzati per il tuo pool di utenti e il client dell'app.
- Impostazione dei valori degli attributi utente.
- Creazione di attributi personalizzati.
- Imposta il valore degli attributi personalizzati immutabili nelle richieste API. `AdminCreateUser` Questa funzionalità non è disponibile nella console Amazon Cognito.
- Specificate una password temporanea, create un utente senza password o consentite ad Amazon Cognito di generare automaticamente una password.
- Crea nuovi utenti e conferma automaticamente i loro account, verifica i loro indirizzi e-mail o verifica i loro numeri di telefono.
- Specificate SMS e messaggi di invito e-mail personalizzati per i nuovi utenti tramite i trigger Console di gestione AWS o Lambda come messaggio personalizzato, mittente SMS personalizzato e mittente e-mail personalizzato.
- Specificare se i messaggi di invito vengono inviati tramite SMS, e-mail, o entrambi.

- Inviare nuovamente il messaggio di benvenuto a un utente esistente chiamando l'API `AdminCreateUser`, specificando `RESEND` per il parametro `MessageAction`.
- Sopprimi l'invio del messaggio di invito quando l'utente viene creato.
- Specificate un limite di scadenza fino a 90 giorni per i nuovi account utente.
- Consentire agli utenti di registrarsi o richiedere che i nuovi utenti vengano aggiunti solo dall'amministratore.

Gli amministratori possono inoltre accedere agli utenti con AWS credenziali in un'applicazione lato server. Per ulteriori informazioni, consulta Modelli di autorizzazione per l'autenticazione tramite API e SDK.

Flussi di autenticazione degli utenti e creazione di utenti

La creazione amministrativa degli utenti presenta opzioni che differiscono in base alla configurazione del pool di utenti. I flussi di autenticazione, o metodi disponibili per gli utenti per l'accesso e l'MFA, possono modificare il modo in cui crei gli utenti e i messaggi che invii loro. Di seguito sono riportati alcuni flussi di autenticazione disponibili nei pool di utenti.

- Nome utente e password
- Passkey
- Sign-in con terze parti IdPs
- Senza password con password monouso (OTP) per e-mail e SMS
- Multi-factor autenticazione con e-mail, SMS e OTP dell'app di autenticazione
- Autenticazione personalizzata con trigger Lambda

Per ulteriori informazioni su come configurare questi fattori di accesso, consulta. Autenticazione con pool di utenti Amazon Cognito

Creare utenti senza password

Se hai abilitato l'accesso senza password per il tuo pool di utenti, puoi creare utenti senza password. Per creare un utente senza password, è necessario fornire i valori degli attributi per un fattore di accesso senza password disponibile. Ad esempio, se nel pool di utenti è disponibile l'accesso OTP senza password tramite posta elettronica, è possibile creare un utente senza password e senza un attributo di indirizzo e-mail. Se gli unici flussi di autenticazione disponibili per i nuovi utenti richiedono

una password, ad esempio passkey o username-password, è necessario creare o generare una password temporanea per ogni nuovo utente.

Per creare un nuovo utente senza password

- Scegli Non impostare una password nella console Amazon Cognito
- Ometti o lascia vuoto il `TemporaryPassword` parametro della tua richiesta API `AdminCreateUser`

Gli utenti senza password vengono confermati automaticamente

Normalmente i nuovi utenti ricevono una password temporanea e entrano in uno `FORCE_CHANGE_PASSWORD` stato al momento della creazione. Quando si creano utenti senza password, questi entrano immediatamente in uno `CONFIRMED` stato. Non puoi inviare nuovamente i codici di conferma a questi utenti nello `CONFIRMED` stato.

I messaggi di invito cambiano per gli utenti senza password.

Per impostazione predefinita, Amazon Cognito invia un messaggio di invito ai nuovi utenti con il seguente messaggio: `Your username is {userName} and your password is {####}.` Quando crei utenti senza password, il messaggio riporta `Your username is {userName}.` Personalizza il tuo messaggio di invito per indicare se intendi impostare le password per gli utenti. Ometti la variabile `{####}` password nei modelli di autenticazione senza password.

Non è possibile generare automaticamente le password quando sono disponibili fattori privi di password

Se hai configurato il tuo pool di utenti per supportare l'accesso senza password OTP tramite e-mail o telefono, non puoi generare automaticamente una password. Per ogni utente che disporrà di una password, è necessario impostare una password temporanea al momento della creazione del profilo.

Gli utenti senza password devono avere valori per tutti gli attributi obbligatori

Quando crei un utente senza password, la tua richiesta ha esito positivo solo se l'utente fornisce valori per tutti gli attributi che hai contrassegnato come obbligatori nel tuo pool di utenti. Ciò si applica a qualsiasi attributo obbligatorio, non solo al numero di telefono e agli attributi e-mail richiesti per la consegna OTP.

Creazione di utenti che forniranno i valori degli attributi obbligatori in un secondo momento

Potresti voler richiedere gli attributi nel tuo pool di utenti, ma raccoglierli dopo aver creato gli utenti a livello amministrativo, durante l'interazione dell'utente nell'applicazione. Gli amministratori possono omettere i valori per gli attributi obbligatori quando creano utenti con password temporanee. Non è possibile omettere i valori degli attributi obbligatori per gli utenti senza password.

Gli utenti con valori mancanti per gli attributi obbligatori e una password temporanea ricevono una sfida NEW_PASSWORD_REQUIRED al primo accesso. Possono quindi fornire un valore per gli attributi obbligatori mancanti nel parametro. `requiredAttributes` È possibile creare utenti con password e senza attributi obbligatori solo se tutti gli attributi obbligatori sono modificabili. Gli utenti possono completare l'accesso con `NEW_PASSWORD_REQUIRED` sfide e valori degli attributi obbligatori solo se gli attributi richiesti sono scrivibili dal client dell'app con cui accedono.

Quando imposti una password permanente per un utente creato dall'amministratore, il suo stato cambia *`CONFIRMED`* e il pool di utenti non richiede loro una nuova password o gli attributi obbligatori al primo accesso.

Creazione di un nuovo utente in Console di gestione AWS

Puoi impostare i requisiti della password utente, configurare i messaggi di invito e verifica inviati agli utenti e aggiungere nuovi utenti con la console Amazon Cognito.

Impostare una policy per le password e abilitare l'auto-registrazione

È possibile configurare le impostazioni per password dalla complessità minima e decidere se gli utenti possono registrarsi utilizzando le API pubbliche nel bacino d'utenza.

Configurare una policy per le password

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli il menu Metodi di autenticazione e individua la politica delle password. Scegli Modifica.
4. Impostare la Password policy mode (modalità di policy per le password) su Custom (personalizzata).
5. Scegli una lunghezza minima della password. Per i limiti al requisito di lunghezza della password, consulta User pools resource quotas (Quote di risorse del pool di utenti).
6. Scegli un requisito di Password complexity (complessità delle password).

7. Scegli la durata della validità della password impostata dagli amministratori.
8. Scegli Save changes (salva modifiche).

Consenti l'iscrizione self-service

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli il Sign-upmenu e individua la Self-serviceregistrazione. Seleziona Edit (Modifica).
4. Scegli se abilitare l'autoregistrazione. Self-registrationviene in genere utilizzato con client di app pubbliche che devono registrare nuovi utenti nel pool di utenti senza distribuire un client secret o credenziali API AWS Identity and Access Management (IAM).

   > **Disattivare dell'auto-registrazione**
   >
   > Se non abiliti l'auto-registrazione, i nuovi utenti devono essere creati mediante operazioni API amministrative utilizzando le credenziali API IAM o tramite accesso con i provider federati.

5. Scegli Save changes (Salva modifiche).

Personalizzare e-mail ed SMS

Personalizzare i messaggi utente

Puoi personalizzare i messaggi inviati da Amazon Cognito ai tuoi utenti che ricevono un invito ad accedere, si registrano per un account utente o a cui viene richiesta l'autenticazione a più fattori (MFA) all'accesso.

> **Note**
>
> Viene inviato un Invitation message (messaggio di invito) quando crei un utente nel bacino d'utenza e lo inviti a effettuare l'accesso. Amazon Cognito invia le informazioni di accesso iniziali all'indirizzo e-mail o al numero di telefono dell'utente.
> Quando un utente effettua la registrazione per un account utente nel bacino d'utenza, viene inviato un messaggio di verifica. Amazon Cognito invia un codice all'utente. Quando l'utente fornisce il codice ad Amazon Cognito, verifica le informazioni di contatto e conferma il proprio account per l'accesso. I codici di verifica sono validi 24 ore.

Quando si abilita la MFA via SMS nel bacino d'utenza e un utente che ha configurato la MFA via SMS accede e gli viene richiesto di effettuare la MFA, viene inviato un Messaggio MFA.

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli il menu Modelli di messaggio e seleziona Messaggio di verifica, Messaggio di invito o Messaggio MFA e scegli Modifica.
4. Personalizza i messaggi per il tipo di messaggio scelto.

> **Note**
>
> Tutte le variabili nei modelli di messaggio devono essere incluse quando si personalizza il messaggio. Se la variabile, ad esempio {####}, non è inclusa, l'utente avrà informazioni insufficienti per completare l'azione del messaggio.
> Per ulteriori informazioni, vedi Message templates (modelli di messaggi).

5. a. Verification messages (Messaggi di verifica)

    i. Scegli un Verification type (tipo di verifica) per le e-mail. Un Code (codice) di verifica invia un codice numerico che l'utente dovrà inserire. Un Link di verifica invia un link su cui l'utente può fare clic per verificare le informazioni di contatto. Il testo nella variabile per un messaggio Link viene visualizzato come testo del collegamento ipertestuale. Ad esempio, viene visualizzato un modello di messaggio che utilizza la variabile {##Click here##} Click here (clicca qui) nel testo dell'e-mail.

    ii. Specificare un Email subject (oggetto dell'e-mail) per i messaggi via e-mail.

    iii. Inserisci un modello di e-mail personalizzato per i messaggi via e-mail. È possibile personalizzare questo modello in formato HTML.

    iv. Inserisci un modello personalizzato di SMS per i messaggi SMS.

    v. Scegli Save changes (salva modifiche).

    b. Invitation messages (Messaggi di invito)

    i. Specificare un Email subject (oggetto dell'e-mail) per i messaggi via e-mail.

    ii. Inserisci un modello di e-mail personalizzato per i messaggi via e-mail. È possibile personalizzare questo modello in formato HTML.

    iii. Inserisci un modello personalizzato di SMS per i messaggi SMS.

iv. Scegli Save changes (salva modifiche).

c. MFA messages (Messaggi MFA)

i. Inserisci un modello personalizzato di SMS per i messaggi SMS.

ii. Scegli Save changes (salva modifiche).

Creazione di un utente

Creazione di un utente

Puoi creare nuovi utenti per il tuo bacino d'utenza dalla console Amazon Cognito. In genere, gli utenti possono accedere dopo aver impostato una password. Per effettuare l'accesso con un indirizzo e-mail l'utente deve verificare l'attributo `email`. Per accedere con un numero di telefono, l'utente deve verificare l'attributo `phone_number`. Per confermare gli account come amministratore, puoi anche utilizzare l'API AWS CLI o o creare profili utente con un provider di identità federato. Per ulteriori informazioni, consulta la documentazione di riferimento dell'API di Amazon Cognito.

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli il menu Utenti e scegli Crea un utente.
4. Esamina i User pool sign-in and security requirements (Requisiti di accesso e sicurezza del bacino d'utenza) per informazioni sui requisiti delle password, sui metodi di recupero dell'account disponibili e sugli attributi alias per il bacino d'utenza.
5. Scegli come desideri inviare l'Invitation message (Messaggio d'invito). Puoi scegliere messaggio SMS, messaggio e-mail o entrambi. Per eliminare il messaggio di invito, scegli Non inviare un invito.

> **Note**
>
> Prima di poter inviare messaggi di invito, configura un mittente e un messaggio Regione AWS con Amazon Simple Notification Service e Amazon Simple Email Service nel menu Metodi di autenticazione del tuo pool di utenti. Messaggi e velocità dati del destinatario. Amazon SES fattura separatamente i messaggi e-mail e Amazon SNS fattura separatamente i messaggi SMS.

6. Scegli uno Username (nome utente) per il nuovo utente.

7. Scegli se selezionare Create a password (Crea una password) o se vuoi che Amazon Cognito esegua l'operazione Generate a password (Genera una password) per l'utente. L'opzione per generare una password non è disponibile se nel pool di utenti è disponibile l'accesso senza password. Qualsiasi password temporanea deve rispettare i criteri delle password del bacino d'utenza.
8. Scegli Crea.
9. Scegli il menu Utenti e scegli la voce Nome utente per l'utente. Aggiungi e modifica User attributes (Attributi utente) e Group memberships (Appartenenze a gruppi). Esamina User event history (Cronologia eventi dell'utente).

Aggiunta di gruppi a un bacino d'utenza

Il supporto per i gruppi nei bacini d'utenza di Amazon Cognito ti permette di creare e gestire gruppi e di aggiungere e rimuovere gli utenti dai gruppi. Utilizza i gruppi per creare raccolte di utenti e gestire le loro autorizzazioni o rappresentare diversi tipi di utenti. Puoi assegnare un ruolo AWS Identity and Access Management (IAM) a un gruppo per definire le autorizzazioni per i membri di un gruppo.

Puoi utilizzare i gruppi per creare una raccolta di utenti in un bacino d'utenza, operazione che spesso viene svolta per impostare le autorizzazioni di tali utenti. Ad esempio, è possibile creare gruppi separati per utenti che siano lettori, collaboratori e revisori del tuo sito Web e della tua app. Utilizzando il ruolo IAM associato a un gruppo, puoi impostare anche autorizzazioni diverse per questi diversi gruppi, in modo che solo i collaboratori possano inserire dei contenuti in Amazon S3 e solo gli editor possano pubblicare dei contenuti tramite un'API in Amazon API Gateway.

Amazon Cognito crea un gruppo di utenti per ogni OIDC, SaML e social identity provider (IdP) che aggiungi al tuo pool di utenti. Il nome del gruppo è nel formato`[user pool ID]_[IdP name]`, ad esempio o. `us-east-1_EXAMPLE_MYSSO` `us-east-1_EXAMPLE_Google` Ogni profilo utente IdP univoco generato automaticamente viene aggiunto automaticamente a questo gruppo. Gli utenti collegati non vengono aggiunti automaticamente a questo gruppo, ma puoi aggiungere i loro profili al gruppo con un processo separato.

È possibile creare e gestire gruppi in un pool di utenti da Console di gestione AWS, le API e la CLI. In qualità di sviluppatore (utilizzando AWS le credenziali), puoi creare, leggere, aggiornare, eliminare ed elencare i gruppi per un pool di utenti. Puoi inoltre aggiungere e rimuovere gli utenti dai gruppi.

Non sono previsti costi aggiuntivi per l'utilizzo di gruppi all'interno di un bacino d'utenza. Per ulteriori informazioni, consulta Prezzi di Amazon Cognito.

Assegnazione di ruoli IAM ai gruppi

È possibile utilizzare i gruppi per controllare le autorizzazioni per le risorse utilizzando un ruolo IAM. I ruoli IAM includono policy di attendibilità e policy di autorizzazione. La policy di attendibilità del ruolo specifica chi può utilizzare il ruolo. Le policy di autorizzazioni specificano le operazioni e le risorse a cui i membri del gruppo possono accedere. Quando crei un ruolo IAM, imposta la policy di attendibilità del ruolo per consentire agli utenti del gruppo di assumere il ruolo. Nelle policy delle autorizzazioni del ruolo specificare le autorizzazioni che si desidera concedere al gruppo.

Quando crei un gruppo in Amazon Cognito, specifichi un ruolo IAM fornendo l'ARNdel ruolo. Quando i membri del gruppo accedono utilizzando Amazon Cognito, possono ricevere credenziali temporanee dai pool di identità. Le autorizzazioni sono determinate dal ruolo IAM associato.

I singoli utenti possono essere in più gruppi. In qualità di sviluppatore, hai a disposizione le seguenti opzioni per scegliere automaticamente il ruolo IAM quando un utente si trova in più gruppi:

- Puoi assegnare i valori di priorità per ciascun gruppo. Viene scelto il gruppo con la precedenza migliore (inferiore) e viene applicato il ruolo IAM associato.
- L'app può anche scegliere tra i ruoli disponibili quando richiede AWS le credenziali per un utente tramite un pool di identità, specificando un ruolo ARN nel parametro. GetCredentialsForIdentity`CustomRoleARN` Il ruolo IAM specificato deve corrispondere a un ruolo disponibile per l'utente.

Assegnazione dei valori di priorità ai gruppi

Un utente può appartenere a più di un gruppo. Nei token di accesso e ID dell'utente, l'attestazione `cognito:groups` contiene l'elenco di tutti i gruppi a cui un utente appartiene. L'attestazione `cognito:roles` contiene l'elenco dei ruoli corrispondenti ai gruppi.

Poiché un utente può appartenere a più di un gruppo, puoi assegnare una priorità a ciascun gruppo. Questo è un numero non negativo che specifica la priorità di questo gruppo rispetto agli altri gruppi ai quali un utente appartiene nel bacino d'utenza. Zero è il valore di priorità massimo. I gruppi con valori di priorità più bassi prevalgono sui gruppi con valori di priorità più alti o nulli. Se un utente appartiene a due o più gruppi, al gruppo con il valore di precedenza più basso viene applicato il ruolo IAM all'`cognito:preferred_role`attestazione nel token ID dell'utente.

Due gruppi possono avere lo stesso valore di priorità. In questo caso nessun gruppo prevale sull'altro. Se due gruppi con lo stesso valore di priorità hanno lo stesso ruolo ARN, tale ruolo viene

utilizzato nell'attestazione `cognito:preferred_role` nei token ID per gli utenti di ciascun gruppo. Se i due gruppi hanno diversi ruoli ARN, l'attestazione `cognito:preferred_role` non è impostata nei token ID degli utenti.

Utilizzo di gruppi per controllare l'autorizzazione con Amazon API Gateway

Puoi utilizzare i gruppi di un bacino d'utenza per controllare l'autorizzazione con Amazon API Gateway. I gruppi di cui un utente è membro sono inclusi sia nel token ID che nel token di accesso da un bacino d'utenza nell'attestazione `cognito:groups`. Puoi inviare ID o token di accesso con richieste ad Amazon API Gateway e utilizzare un autorizzazione del bacino d'utenza Amazon Cognito per un'API REST. Per ulteriori informazioni, consulta la sezione Control access to a REST API using Amazon Cognito user pools as authorizer (Controllo degli accessi a un'API REST utilizzando pool di utenti di Amazon Cognito come autorizzazione) nella Guida per sviluppatori di API Gateway.

È inoltre possibile autorizzare l'accesso a un'API HTTP di Amazon API Gateway con un'autorizzazione JWT personalizzata. Per ulteriori informazioni consulta la sezione Controlling access to HTTP APIs with JWT authorizers (Controllo dell'accesso alle API HTTP con le autorizzazioni JWT) nella Guida per gli sviluppatori dell'API Gateway.

Limitazioni per i gruppi

I gruppi di utenti sono soggetti alle seguenti limitazioni:

- Il numero di gruppi che puoi creare è limitato dalle quote del servizio Amazon Cognito.
- I gruppi non possono essere annidati.
- Non puoi cercare gli utenti in un gruppo.
- Non puoi cercare i gruppi per nome, ma puoi farne un elenco.

Creare un nuovo gruppo in Console di gestione AWS

Per creare un nuovo gruppo, utilizza la procedura seguente.

Per creare un nuovo gruppo

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Gruppi, quindi scegli Crea un gruppo.

5. Nella pagina Create a group (Crea gruppo), sotto Group name (nome gruppo) inserisci un nome per il gruppo.
6. Facoltativamente, è possibile fornire ulteriori informazioni su questo gruppo utilizzando uno dei seguenti campi:
   - Description (Descrizione) - Inserisci i dettagli sull'utilizzo futuro di questo nuovo gruppo.
   - Precedence (Priorità) - Amazon Cognito valuta e applica tutte le autorizzazioni di gruppo per un determinato utente in base ai gruppi a cui appartengono hanno un valore di precedenza inferiore. Viene scelto il gruppo con la precedenza inferiore e viene applicato il ruolo IAM associato. Per ulteriori informazioni, consulta Assegnazione dei valori di priorità ai gruppi.
   - Ruolo IAM - È possibile assegnare un ruolo IAM al gruppo quando è necessario controllare le autorizzazioni per le risorse. In caso di integrazione di un bacino d'utenza con un pool di identità, l'impostazione IAM role (ruolo IAM) determina il ruolo da assegnare nel token ID dell'utente se il pool di identità è configurato per la scelta del ruolo dal token. Per ulteriori informazioni, consulta Assegnazione di ruoli IAM ai gruppi.
   - Add users to this group (Aggiungi utenti a questo gruppo) - Aggiungi utenti esistenti come membri di questo gruppo dopo la creazione.
7. Scegli Create (Crea) per confermare.

Gestione e ricerca degli account utente

I pool di utenti possono contenere milioni di utenti. Lavorare con un set di dati di queste dimensioni è una sfida per gli amministratori. Amazon Cognito dispone di strumenti per trovare e modificare i profili utente. I metodi principali per trovare gli utenti sono il menu Utenti della console Amazon Cognito e con. ListUsers Tra i metodi che recuperano informazioni sugli utenti, queste sono le opzioni che non hanno un impatto sui costi, a differenza, ad esempio, di. AdminGetUser

Questa sezione della guida contiene informazioni sulla ricerca e l'aggiornamento dei profili utente in un pool di utenti.

Visualizzazione degli attributi utente

Utilizza la procedura seguente per visualizzare gli attributi utente nella console Amazon Cognito.

Per visualizzare gli attributi utente

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.

2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Utenti e seleziona un utente nell'elenco.
5. Nella pagina dei dettagli dell'utente, alla voce User attributes (attributi utente), è possibile visualizzare quali attributi siano associati all'utente.

Reimpostare una password utente

Utilizza la procedura seguente per reimpostare la password utente nella console Amazon Cognito.

Per reimpostare una password utente

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Utenti e seleziona un utente nell'elenco.
5. Nella pagina dei dettagli dell'utente, scegli Actions (Operazioni),Reset password (reimposta password).
6. Nella finestra di dialogo Reset password (reimposta password), controlla le informazioni e, quando sei pronto, scegli Reimposta.

   L'operazione genera un codice di conferma che viene inviato all'utente e ne disabilita la password corrente cambiando lo stato dell'utente in `RESET_REQUIRED`. Il codice Reset password (Reimposta password) è valido per 1 ora.

Abilita, disabilita ed elimina gli account utente

È possibile eliminare i profili utente non utilizzati o, se si desidera impedire temporaneamente l'accesso, disabilitarli. Gli utenti possono eliminare i propri account, ma solo gli amministratori del pool di utenti possono abilitare e disabilitare gli account utente.

Effetto dell'eliminazione

Gli utenti non possono accedere con account utente eliminati e per riottenere l'accesso devono registrarsi o essere creati di nuovo.

Effetto della disabilitazione degli account

Quando disabiliti un account utente, Amazon Cognito invalida automaticamente tutte le sessioni autenticate, disattiva l'account utente per l'accesso e revoca i relativi token di accesso e aggiornamento. Amazon Cognito restituisce un `invalid_request` errore con il messaggio `User is not enabled` quando un utente tenta di accedere a un account che hai disabilitato. Questo comportamento non cambia con le impostazioni di divulgazione dell'esistenza degli utenti per il client dell'app. È possibile disabilitare gli account utente locali e i profili locali degli account utente federati. Quando gli utenti accedono con l'accesso gestito o la classica interfaccia utente ospitata, disattivano il loro account e quindi provano ad accedere nuovamente con il cookie del browser che mantiene la sessione autenticata, Amazon Cognito li reindirizza alla pagina di accesso.

Effetto dell'attivazione degli account

Gli utenti possono accedere immediatamente agli account dopo averli abilitati. Gli account utente sono abilitati per impostazione predefinita. Gli attributi e le password degli utenti rimangono gli stessi di prima della disattivazione dell'account. I token revocati dall'applicazione, indipendentemente dal fatto che tu abbia disabilitato l'account utente o revocato separatamente il token di aggiornamento, rimangono non validi dopo aver abilitato l'account utente che possedeva il token.

Delete a user account (console)

Per eliminare un account utente

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Utenti e seleziona il pulsante di opzione accanto al nome utente di un utente nell'elenco.
5. Scegli Elimina.
6. Scegli Disabilita l'accesso utente.
7. Scegli Elimina.

Delete a user account (API)

Gli utenti possono eliminare i propri account con il funzionamento dell' access-token-authorizedDeleteUserAPI self-service. Di seguito è riportato un esempio di corpo della `DeleteUser` richiesta.

```
{
```

```
   "AccessToken": "eyJra456defEXAMPLE"
}
```

Gli amministratori possono eliminare gli account utente con l'operazione API autorizzata da IAM AdminDeleteUser. Di seguito è riportato un esempio `AdminDeleteUser` di corpo della richiesta.

```
{
   "Username": "testuser",
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Disable a user account (console)

Per disabilitare un account utente

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Utenti e seleziona il nome utente di un utente nell'elenco.
5. Nella pagina dei dettagli dell'utente, scegli Azioni, Disabilita l'accesso utente.
6. Nella finestra di dialogo che viene creata, scegli Disabilita.

Disable a user account (API)

Gli amministratori possono disabilitare gli account utente con l'operazione API autorizzata da IAM AdminDisableUser. Di seguito è riportato un esempio `AdminDisableUser` di corpo della richiesta.

```
{
   "Username": "testuser",
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Enable a user account (console)

Per abilitare un account utente

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.

2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Utenti e seleziona il nome utente di un utente nell'elenco.
5. Nella pagina dei dettagli dell'utente, scegli Azioni, Abilita l'accesso utente.
6. Nella finestra di dialogo che viene creata, scegli Abilita.

Enable a user account (API)

Gli amministratori possono abilitare gli account utente con l'operazione API autorizzata da IAM AdminEnableUser. Di seguito è riportato un esempio `AdminEnableUser` di corpo della richiesta.

```
{
   "Username": "testuser",
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Ricerca degli attributi utente

Se hai già creato un bacino d'utenza, puoi effettuare la ricerca dal pannello Users (Utenti) nella Console di gestione AWS. Puoi anche utilizzare l'ListUsers API Amazon Cognito, che accetta un parametro Filter.

Puoi cercare i seguenti attributi standard. Gli attributi personalizzati non possono essere cercati.

- username (distinzione tra maiuscole e minuscole)
- email
- phone_number
- name
- given_name
- family_name
- preferred_username
- cognito: user_status (denominato Status (Stato) nella console) (senza distinzione tra maiuscole e minuscole)
- status (denominato Enabled (Abilitato) nella console) (distinzione tra maiuscole e minuscole)

- sub

> **Note**
>
> È inoltre possibile fare un elenco degli utenti con un filtro lato client. Il filtro lato server non corrisponde a più di 1 attributo. Per la ricerca avanzata, utilizza un filtro lato client con il parametro `--query` dell'operazione `list-users` nella AWS Command Line Interface. Quando utilizzi un filtro lato client, ListUsers restituisce un elenco impaginato di zero o più utenti. È possibile ricevere più pagine di fila senza risultati. Ripetere la query con ogni token di impaginazione restituito fino a quando non si riceve un valore del token di impaginazione nullo, quindi rivedere il risultato combinato.
> Per ulteriori informazioni sul filtraggio lato server e lato client, consulta Filtraggio dell'output nella Guida per l'utente. AWS CLI AWS Command Line Interface

Ricerca di utenti con Console di gestione AWS

Se hai già creato un bacino d'utenza, puoi effettuare la ricerca dal pannello Users (Utenti) nella Console di gestione AWS.

Console di gestione AWS le ricerche sono sempre ricerche con prefisso («inizia con»).

Per cercare un utente nella console Amazon Cognito

1. Passa alla console Amazon Cognito. È possibile che ti vengano richieste le credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Utenti e inserisci il nome utente nel campo di ricerca. Attenzione: alcuni valori di attributo tengono conto di maiuscole/minuscole (ad esempio lo User name (Nome utente).

   È inoltre possibile trovare utenti modificando il filtro di ricerca per restringere l'ambito alle altre proprietà dell'utente, ad esempio E-mail,numero di telefono, oppure cognome.

Ricerca di utenti tramite l'API **`ListUsers`**

Per cercare utenti dalla tua app, utilizza l'API Amazon CognitoListUsers . Quest'API utilizza i parametri seguenti:

- `AttributesToGet`: una matrice di stringhe, in cui ogni stringa è il nome di un attributo utente da restituire a ogni utente nei risultati di ricerca. Per recuperare tutti gli attributi, non includere un parametro `AttributesToGet` o richiesta `AttributesToGet` con un valore della stringa letterale `null`.
- `Filter`: una stringa del filtro del modulo"`AttributeName` `Filter-Type` "`AttributeValue`"". Le virgolette all'interno della stringa del filtro devono essere sfuggite utilizzando la barra rovesciata (\). Ad esempio, `"family_name = \"Reddy\""`. Se la stringa del filtro è vuota, `ListUsers` riporta tutti gli utenti al bacino d'utenza.
  - `AttributeName`: il nome dell'attributo da cercare. Puoi cercare solo un attributo per volta.

    > **Note**
    >
    > Puoi cercare solo gli attributi standard. Gli attributi personalizzati non possono essere cercati. Questo perché si può effettuare la ricerca solo per gli attributi indicizzati, e gli attributi personalizzati non possono essere indicizzati.

  - `Filter-Type`: per una corrispondenza esatta, utilizza =, ad esempio, `given_name = "Jon"`. Per una corrispondenza del prefisso ("inizia con"), utilizza ^=, ad esempio, `given_name ^= "Jon"`.
  - `AttributeValue`: il valore di attributo che deve essere abbinato a ogni utente.
- `Limit`: numero massimo di utenti da restituire.
- `PaginationToken`: un token per ottenere più risultati da una ricerca precedente. Il token di impaginazione di Amazon Cognito scade dopo un'ora.
- `UserPoolId`: l'ID del bacino d'utenza per il bacino d'utenza sul quale eseguire la ricerca.

Tutte le ricerche non fanno la distinzione tra maiuscole e minuscole. I risultati di ricerca vengono ordinati in base all'attributo denominato dalla stringa `AttributeName`, in ordine crescente.

Esempi di utilizzo dell'API **`ListUsers`**

L'esempio seguente illustra la restituzione di tutti gli utenti e include tutti gli attributi.

```
{
    "AttributesToGet": null,
    "Filter": "",
    "Limit": 10,
```

```
    "UserPoolId": "us-east-1_samplepool"
}
```

L'esempio seguente illustra la restituzione di tutti gli utenti il cui numero di telefono inizia per "+1312" e include tutti gli attributi.

```
{
    "AttributesToGet": null,
    "Filter": "phone_number ^= \"+1312\"",
    "Limit": 10,
    "UserPoolId": "us-east-1_samplepool"
}
```

L'esempio seguente illustra la restituzione dei primi 10 utenti il cui cognome è "Reddy". Per ogni utente, i risultati di ricerca includono il nome specificato dall'utente, il numero di telefono e l'indirizzo e-mail. Se non ci sono più di 10 utenti che corrispondono all'interno del bacino d'utenza, la risposta include un token di paginazione.

```
{
    "AttributesToGet": [
        "given_name",
        "phone_number",
        "email"
    ],
    "Filter": "family_name = \"Reddy\"",
    "Limit": 10,
    "UserPoolId": "us-east-1_samplepool"
}
```

Mentre nell'esempio precedente viene restituito un token di paginazione, in quello seguente vengono restituiti i 10 utenti successivi che corrispondono alla stessa stringa del filtro.

```
{
    "AttributesToGet": [
        "given_name",
        "phone_number",
        "email"
    ],
```

```
    "Filter": "family_name = \"Reddy\"",
    "Limit": 10,
    "PaginationToken": "pagination_token_from_previous_search",
    "UserPoolId": "us-east-1_samplepool"
}
```

Password, ripristino dell'account e politiche relative alle password

A tutti gli utenti che accedono a un pool di utenti, anche gli utenti federati, vengono assegnate delle password ai rispettivi profili utente. Gli utenti locali e gli utenti collegati devono fornire una password al momento dell'accesso. Gli utenti federati non utilizzano le password del pool di utenti, ma accedono con il proprio provider di identità (IdP). Puoi consentire agli utenti di reimpostare le proprie password, reimpostare o modificare le password in qualità di amministratore e impostare politiche per la complessità e la cronologia delle password.

Amazon Cognito non memorizza le password degli utenti in testo normale. Invece, memorizza un hash della password di ogni utente con un sale specifico per l'utente. Per questo motivo, non è possibile recuperare le password esistenti dai profili utente nei pool di utenti. È consigliabile non archiviare le password utente in testo semplice da nessuna parte. Esegui la reimpostazione delle password quando gli utenti dimenticano le password.

Reimpostazione e ripristino della password

Gli utenti dimenticano le password. Potresti volere che siano in grado di reimpostare la password da soli oppure potresti voler richiedere che un amministratore reimposti la password per loro. I pool di utenti di Amazon Cognito offrono opzioni per entrambi i modelli. Questa parte della guida illustra le impostazioni del pool di utenti e le operazioni API per la reimpostazione della password.

Il funzionamento dell'ForgotPasswordAPI e l'opzione di accesso gestito Password dimenticata? invia agli utenti un codice che, quando confermano di avere il codice corretto, dà loro l'opportunità di impostare una nuova password. ConfirmForgotPassword Si tratta del modello self-service di recupero delle password.

Recupero di utenti non verificati

Puoi inviare messaggi di ripristino agli utenti che hanno verificato il proprio indirizzo e-mail o numero di telefono. Se non dispongono di un'e-mail o di un telefono di ripristino confermati, un amministratore del pool di utenti può contrassegnare il proprio indirizzo e-mail o numero di telefono come verificato. Modifica gli attributi utente dell'utente nella console Amazon Cognito e seleziona la casella di controllo accanto a Segna il numero di telefono come verificato o

Contrassegna l'indirizzo e-mail come verificato. Puoi anche impostare `email_verified` o su `phone_number_verified` true in una AdminUpdateUserAttributesrichiesta. Per i nuovi utenti, l'operazione ResendConfirmationCodeAPI invia un nuovo codice al loro indirizzo e-mail o numero di telefono e possono completare la conferma e la verifica in modalità self-service.

Reimposta le password come amministratore

Le operazioni AdminSetUserPassworde AdminResetUserPasswordAPI sono i metodi di reimpostazione della password avviati dall'amministratore. `AdminSetUserPassword`imposta una password temporanea o permanente e `AdminResetUserPassword` invia agli utenti un codice per la reimpostazione della password nello stesso modo di. `ForgotPassword`

Configura la reimpostazione e il ripristino della password

Amazon Cognito seleziona automaticamente le opzioni di ripristino dell'account tra gli attributi richiesti e le opzioni di accesso che scegli quando crei un pool di utenti nella console. Puoi modificare queste impostazioni predefinite.

Il metodo MFA preferito dall'utente influenza i metodi che può utilizzare per recuperare la password. Gli utenti la cui MFA preferita è tramite messaggio di posta elettronica non possono ricevere un codice di reimpostazione della password tramite e-mail. Gli utenti la cui MFA preferita è tramite SMS non possono ricevere un codice di reimpostazione della password tramite SMS.

Le impostazioni di recupero della password devono fornire un'opzione alternativa quando gli utenti non sono idonei al metodo preferito di reimpostazione della password. Ad esempio, i meccanismi di ripristino potrebbero avere l'e-mail come priorità assoluta e la MFA e-mail potrebbe essere un'opzione nel tuo pool di utenti. In questo caso, aggiungi il ripristino SMS-message dell'account come seconda opzione o utilizza le operazioni amministrative dell'API per reimpostare le password di tali utenti.

Amazon Cognito risponde alle richieste di reimpostazione della password di utenti che non dispongono di un metodo di ripristino valido con una risposta di errore. `InvalidParameterException`

> **Note**
>
> Gli utenti non possono ricevere codici MFA e di reimpostazione della password allo stesso indirizzo e-mail o numero di telefono. Se utilizzano password monouso (OTP) dai messaggi e-mail per MFA, devono utilizzare i messaggi SMS per il ripristino dell'account. Se utilizzano le OTP dei messaggi SMS per l'MFA, devono utilizzare i messaggi e-mail per il ripristino dell'account. Nei pool di utenti con MFA, gli utenti potrebbero non essere in grado di

completare il ripristino self-service della password se dispongono di attributi per il proprio indirizzo e-mail ma non dispongono del numero di telefono o del numero di telefono ma non dell'indirizzo e-mail.
Per evitare che gli utenti non possano reimpostare le proprie password nei pool di utenti con questa configurazione, imposta gli `phone_number` attributi `email` e come richiesto.
In alternativa, puoi configurare processi che raccolgono e impostano sempre tali attributi quando gli utenti si registrano o quando gli amministratori creano profili utente. Quando gli utenti dispongono di entrambi gli attributi, Amazon Cognito invia automaticamente i codici di reimpostazione della password alla destinazione che non è il fattore MFA dell'utente.

La procedura seguente configura il ripristino degli account in modalità self-service in un pool di utenti.

Configure self-service password reset (API/SDK)

Il `AccountRecoverySetting` parametro è il parametro del pool di utenti che imposta i metodi che gli utenti possono utilizzare per recuperare la password nelle richieste ForgotPasswordAPI o quando selezionano Password dimenticata? nell'accesso gestito. `ForgotPassword`invia un codice di ripristino a un'email o a un numero di telefono verificato. Il codice di ripristino è valido per un'ora. Quando si specifica AccountRecoverySetting per il bacino d'utenza, Amazon Cognito sceglie la destinazione di distribuzione del codice in base alla priorità impostata.

Quando si definisce `AccountRecoverySetting` e un utente dispone di SMS MFA configurato, SMS non può essere utilizzato come meccanismo di recupero degli account. La priorità per questa impostazione è determinata con 1 la massima priorità. Amazon Cognito invia una verifica solo a uno dei metodi specificati. L'esempio seguente `AccountRecoverySetting` imposta gli indirizzi e-mail come destinazione principale per i codici di ripristino dell'account, ricorrendo al messaggio SMS se l'utente non dispone di un attributo di indirizzo e-mail.

```
"AccountRecoverySetting": {
   "RecoveryMechanisms": [
      {
         "Name": "verified_email",
         "Priority": 1
      },
      {
         "Name": "verified_phone_number",
         "Priority": 2
      }
   ]
```

```
}
```

Il valore `admin_only` disattiva il ripristino self-service dell'account, richiedendo invece agli utenti di contattare l'amministratore per la reimpostazione della password. Non è possibile utilizzare `admin_only` con altri meccanismi di recupero degli account. Quanto segue e

```
"AccountRecoverySetting": {
   "RecoveryMechanisms": [
      {
         "Name": "admin_only",
         "Priority": 1
      }
   ]
}
```

Se non lo specifichi`AccountRecoverySetting`, Amazon Cognito invia prima il codice di ripristino a un numero di telefono verificato e a un indirizzo e-mail verificato se gli utenti non dispongono di un attributo di numero di telefono.

Per ulteriori informazioni su `AccountRecoverySetting`, consulta CreateUserPool e UpdateUserPool.

Configure self-service password reset (console)

Configura le opzioni di ripristino dell'account e di reimpostazione della password dal menu del Sign-intuo pool di utenti.

Per configurare il ripristino dell'account utente

1. Accedi alla console Amazon Cognito.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il Sign-inmenu. Individua il recupero dell'account utente e scegli Modifica
5. Per consentire agli utenti di reimpostare le proprie password, scegli Abilita il ripristino dell'account in modalità self-service.
6. Configura il metodo di consegna per i codici di recupero delle password che il tuo pool di utenti invia agli utenti. In Metodo di consegna per i messaggi di recupero dell'account utente, seleziona un'opzione disponibile. Come procedura consigliata, scegli un'opzione che abbia un metodo secondario per l'invio dei messaggi, ad esempio e-mail se disponibile, altrimenti

SMS. Con un metodo di consegna secondario, Amazon Cognito può inviare codici agli utenti in un modo che richiede loro di utilizzare un supporto diverso per la reimpostazione della password rispetto all'MFA.

7. Seleziona Salva modifiche.

Come comportarsi in caso di password dimenticata

In una data ora, permettiamo tra 5 e 20 tentativi per un utente di richiedere o inserire un codice di reimpostazione della password come parte delle operazioni password-dimenticata e conferma-password-dimenticata. Il valore esatto dipende dai parametri di rischio associati alle richieste. Notare che questo comportamento è soggetto a modifiche.

Aggiunta di requisiti password del bacino d'utenza

Le password complesse e sicure sono una best practice di sicurezza per il tuo pool di utenti. Soprattutto nelle applicazioni aperte a Internet, le password deboli possono esporre le credenziali degli utenti a sistemi che indovinano le password e cercano di accedere ai dati. Più una password è complessa, più è difficile indovinarla. Amazon Cognito offre strumenti aggiuntivi per gli amministratori attenti alla sicurezza, come la protezione dalle minacce e gli ACL AWS WAF Web, ma la politica delle password è un elemento centrale della sicurezza della directory utenti.

Le password per gli utenti locali nei pool di utenti di Amazon Cognito non scadono automaticamente. Come best practice, registra l'ora, la data e i metadati delle reimpostazioni delle password degli utenti in un sistema esterno. Con un registro esterno dell'età della password, l'applicazione o un trigger Lambda possono cercare l'età della password di un utente e richiederne la reimpostazione dopo un determinato periodo.

Puoi configurare il tuo pool di utenti in modo che richieda una complessità minima della password conforme ai tuoi standard di sicurezza. Le password complesse hanno una lunghezza minima di almeno otto caratteri. Includono anche una combinazione di caratteri maiuscoli, numerici e speciali.

Con i livelli di funzionalità Essentials o Plus, puoi anche impostare una politica per il riutilizzo delle password. È possibile impedire a un utente di reimpostare la propria password con una nuova password che corrisponda alla password corrente o con una qualsiasi delle 23 password precedenti aggiuntive, per un totale massimo di 24.

> **Note**
>
> Amazon Cognito non applica la tua politica sulle password quando importi utenti con hash di password. Verifica la sicurezza della password nel tuo sistema di origine prima di generare hash per l'importazione.

Impostazione di una policy delle password di un pool di utenti

1. Crea un pool di utenti e vai al passaggio Configura i requisiti di sicurezza oppure accedi a un pool di utenti esistente e vai al menu Metodi di autenticazione.
2. Vai a Policy delle password.
3. Scegli una Modalità policy delle password. I Valori predefiniti di Cognito configurano il pool di utenti con le impostazioni minime consigliate. Puoi anche scegliere una policy delle password Personalizzata.
4. Imposta una Lunghezza minima della password. Tutti gli utenti devono registrarsi o essere creati con una password la cui lunghezza sia maggiore o uguale a questo valore. Puoi impostare questo valore minimo fino a 99, ma gli utenti possono impostare password lunghe fino a 256 caratteri.
5. Configura le regole di complessità delle password in Requisiti delle password. Scegli i tipi di caratteri (numeri, caratteri speciali, lettere maiuscole e minuscole) che devono essere presenti almeno in una occorrenza nelle password di ogni utente.

   È possibile richiedere almeno uno dei seguenti caratteri nelle password. Dopo che Amazon Cognito ha verificato che le password contengano i caratteri minimi richiesti, le password degli utenti possono contenere caratteri aggiuntivi di qualsiasi tipo fino alla lunghezza massima della password.

   - Lettere maiuscole e minuscole dell'alfabeto latino di base
   - Numeri
   - Possono includere i seguenti caratteri speciali.

     ```
     ^ $ * . [ ] { } ( ) ? " ! @ # % & / \ , > < ' : ; | _ ~ ` = + -
     ```

   - Non-leading, caratteri spaziali non finali.
6. Imposta un valore per Scadenza delle password temporanee impostate dagli amministratori. Trascorso questo periodo di tempo, un nuovo utente creato con una richiesta API

`AdminCreateUser` nella console di Amazon Cognito non può effettuare l'accesso e impostare una nuova password. Dopo aver effettuato l'accesso con la password temporanea, i relativi account utente non scadono mai. Per aggiornare la durata della password nell'API dei pool di utenti di Amazon Cognito, imposta un valore per TemporaryPasswordValidityDays nella tua richiesta CreateUserPoolo UpdateUserPoolAPI.

7. Imposta un valore per Impedisci l'uso di password precedenti, se disponibile. Per utilizzare questa funzionalità, scegli il livello di funzionalità Essentials o Plus nel tuo pool di utenti. Il valore di questo parametro è il numero di password precedenti a cui una nuova password non può corrispondere quando un utente reimposta la propria password.

Per reimpostare l'accesso per un account utente scaduto, esegui una delle seguenti operazioni:

- Invia una nuova password temporanea e reimposta il periodo di scadenza con una richiesta AdminCreateUserAPI `MessageAction` impostata su. `RESEND`
- Elimina la CA e creane una nuova.
- Genera un nuovo codice di conferma in una richiesta AdminResetUserPasswordAPI.

Importazione di utenti in un bacino d'utenza

Sono disponibili due metodi per importare o migrare gli utenti dalla directory o dal database di utenti esistente nei bacini d'utenza di Amazon Cognito. Puoi eseguire la migrazione di utenti durante la procedura di accesso utilizzando Amazon Cognito per la prima volta con un trigger Lambda di migrazione utenti. Con questo approccio, gli utenti possono continuare a utilizzare le password esistenti senza doverle reimpostare dopo la migrazione al pool di utenti. In alternativa, è possibile migrare gli utenti in blocco caricando un file CSV che contiene gli attributi del profilo utente di tutti gli utenti. Le seguenti sezioni descrivono questi due approcci.

Altre risorse

- Approcci per la migrazione degli utenti ai pool di utenti di Amazon Cognito
- AWS RE:Inforce 2023 - Migrazione ad Amazon Cognito

Argomenti

- Importazione di utenti con un trigger Lambda per la migrazione di utenti
- Importazione di utenti nel bacino d'utenza da un file CSV

Importazione di utenti con un trigger Lambda per la migrazione di utenti

Questo approccio ti consente di eseguire senza problemi la migrazione degli utenti dalla directory degli utenti esistente ai bacini d'utenza, quando un utente accede per la prima volta con la tua app o richiede la reimpostazione della password. Aggiungi una funzione Trigger Lambda di migrazione utenti al bacino d'utenza per ricevere i metadati sugli utenti che tentano di accedere e ottenere le informazioni del profilo utente da un'origine di identità esterna. Per i dettagli e un codice di esempio su questo trigger Lambda, inclusi i parametri di richiesta e risposta, consulta Parametri del trigger Lambda di migrazione utenti.

Prima di avviare la migrazione degli utenti, crea una funzione Lambda di migrazione degli utenti nel tuo Account AWS e imposta la funzione Lambda come trigger di migrazione degli utenti nel bacino d'utenza. Aggiungi una policy di autorizzazione alla funzione Lambda che consenta solo al principale dell'account del servizio Amazon Cognito `cognito-idp.amazonaws.com` di richiamare la funzione Lambda e solo nel contesto del proprio bacino d'utenza. Per ulteriori informazioni, consulta Utilizzo delle policy basate su risorse per AWS Lambda (policy della funzione Lambda).

Sign-in processo

1. L'utente apre l'app e accede con l'API dei pool di utenti di Amazon Cognito o tramite accesso gestito. Per ulteriori informazioni su come facilitare l'accesso con le API di Amazon Cognito, consulta Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili.
2. L'app invia il nome utente e la password a Amazon Cognito. Se la tua app ha un'interfaccia utente di accesso personalizzata creata con un AWS SDK, l'app deve utilizzare InitiateAutho AdminInitiateAuthcon il `USER_PASSWORD_AUTH` flusso or. `ADMIN_USER_PASSWORD_AUTH` Quando l'app utilizza uno di questi flussi, l'SDK invia la password al server.

> **Note**
>
> Prima di aggiungere un trigger di migrazione degli utenti, attiva il flusso `USER_PASSWORD_AUTH` o `ADMIN_USER_PASSWORD_AUTH` nelle impostazioni del client di app. Devi utilizzare questi flussi al posto del flusso `USER_SRP_AUTH` di default. Amazon Cognito deve inviare una password alla funzione Lambda in modo che possa verificare l'autenticazione dell'utente nell'altra directory. Una SRP oscura la password dell'utente nella funzione Lambda.

3. Amazon Cognito verifica se il nome utente inviato corrisponde a un nome utente o alias nel pool di utenti. Puoi impostare l'indirizzo e-mail, il numero di telefono o il nome utente preferito dell'utente come alias nel tuo pool di utenti. Se l'utente non esiste, Amazon Cognito invia i parametri, tra cui nome utente e password, alla tua Trigger Lambda di migrazione utenti funzione.
4. La funzione Trigger Lambda di migrazione utenti controlla o autentica l'utente nella directory o nel database degli utenti esistente. La funzione restituisce gli attributi utente archiviati da Amazon Cognito nel profilo dell'utente nel bacino d'utenza. Puoi restituire un `username` parametro solo se il nome utente inviato corrisponde a un attributo alias. Se desideri che gli utenti continuino a usare le loro password esistenti, la funzione imposta l'attributo `finalUserStatus` su `CONFIRMED` nella risposta Lambda. L'app deve restituire tutti i parametri `"response"` mostrati in Parametri del trigger Lambda di migrazione utenti.

   > **Important**
   >
   > Non registrare l'intero oggetto evento di richiesta nel codice Lambda di migrazione degli utenti in quanto questo oggetto evento di richiesta include la password dell'utente. Se non si eliminano i registri, le password vengono visualizzate in Registri. CloudWatch

5. Amazon Cognito crea il profilo utente nel bacino d'utenza e restituisce i token per il client dell'app.
6. L'app esegue l'acquisizione del token, accetta l'autenticazione dell'utente e procede al contenuto richiesto.

Dopo aver eseguito la migrazione degli utenti, utilizza `USER_SRP_AUTH` per l'accesso. Il protocollo Secure Remote Password (SRP) non invia la password in rete e offre notevoli vantaggi dal punto di vista della sicurezza grazie al flusso `USER_PASSWORD_AUTH` usato durante la migrazione.

In caso di errori durante la migrazione, compresi i problemi del dispositivo client o di rete, l'app riceve le risposte di errore dall'API dei bacini d'utenza di Amazon Cognito. In questo caso, Amazon Cognito potrebbe non creare l'account utente nel bacino d'utenza. L'utente deve quindi tentare di eseguire nuovamente l'accesso. Se l'errore dell'accesso si ripete, tenta di reimpostare la password dell'utente con il flusso della password dimenticata dell'app.

Il flusso della password dimenticata richiama anche la funzione Trigger Lambda di migrazione utenti con un'origine eventi `UserMigration_ForgotPassword`. Poiché l'utente non invia una password quando richiede la reimpostazione della password, Amazon Cognito non include una password nell'evento che invia alla funzione Lambda. La funzione può solo cercare l'utente nella directory degli utenti esistente e restituire gli attributi da aggiungere al profilo utente nel bacino d'utenza. Dopo

che la funzione ha completato l'invocazione e restituito la risposta ad Amazon Cognito, il pool di utenti invia un codice di reimpostazione della password tramite e-mail o SMS. Nella tua app, richiedi all'utente il codice di conferma e una nuova password, quindi invia tali informazioni ad Amazon Cognito in ConfirmForgotPassworduna richiesta API. Puoi anche utilizzare le pagine integrate per il flusso di password dimenticata nell'accesso gestito.

Risorse aggiuntive

- Approcci per la migrazione degli utenti ai pool di utenti di Amazon Cognito

Importazione di utenti nel bacino d'utenza da un file CSV

Se disponi di un archivio di identità esterno e hai tempo a disposizione per preparare il pool di utenti per nuovi utenti locali, l'importazione in blocco di utenti da un file con valori separati da virgole (CSV) può essere un'opzione semplice ed economica per la migrazione a un pool di utenti di Amazon Cognito. L'importazione di un file CSV consiste nel scaricare e compilare un file modello, per poi consegnarlo al gruppo di utenti in un processo di importazione. Puoi utilizzare un'importazione CSV per creare rapidamente utenti di prova. Puoi anche compilare il file a livello di codice con richieste API di lettura sul tuo archivio di identità esterno, quindi analizzarne i dettagli e gli attributi in operazioni di scrittura sul file.

Per impostazione predefinita, il processo di importazione imposta i valori per tutti gli attributi utente tranne la password. Questo significa che i tuoi utenti devono cambiare le loro password dopo il primo accesso. Gli utenti sono in uno `RESET_REQUIRED` stato quando vengono importati con questo metodo.

In alternativa, puoi importare gli utenti con gli hash delle password esistenti. Quando specifichi un algoritmo di hashing delle password durante la creazione del processo di importazione e includi gli hash delle password nel tuo file CSV, Amazon Cognito importa gli utenti con le password esistenti. Questi utenti vengono creati con uno `CONFIRMED` stato e possono accedere immediatamente senza reimpostare le password. Per ulteriori informazioni, consulta Importazione di utenti con hash di password.

Il modo più semplice per importare utenti da un file CSV consiste nell'attivare l'accesso senza password nel tuo pool di utenti. Grazie agli attributi dell'indirizzo e-mail e del numero di telefono e alla corretta configurazione del pool di utenti, gli utenti possono accedere con password monouso (OTP) e-mail o SMS subito dopo il completamento del processo di importazione. Per ulteriori informazioni, consulta Necessità degli utenti importati di ripristinare le password.

Puoi anche impostare le password degli utenti con una richiesta AdminSetUserPasswordAPI che imposta il parametro su. `Permanent true` L'importazione di file CSV non contribuisce al numero di utenti attivi mensili (MAU) fatturati nel tuo pool di utenti. Tuttavia, le operazioni di reimpostazione della password generano MAU. Per gestire i costi quando importi un gran numero di utenti con password che potrebbero non essere immediatamente attivi, configura l'applicazione in modo che richieda agli utenti una nuova password quando accedono e ricevono la sfida. `RESET_REQUIRED`

> **Note**
>
> La data di creazione per ogni utente si riferisce al momento in cui tale utente è stato importato nel bacino d'utenza. La data di creazione non è uno degli attributi importati.

Passaggi per creare un processo di importazione degli utenti

1. Crea un ruolo Amazon CloudWatch Logs nella console AWS Identity and Access Management (IAM).
2. Crea il file .csv di importazione degli utenti;
3. Crea ed esegui Web; Facoltativamente, specifica un algoritmo di hashing delle password per importare gli utenti con gli hash delle password esistenti.
4. Carica il file .csv di importazione degli utenti;
5. Avvia ed esegui Web;
6. Utilizzato CloudWatch per controllare il registro degli eventi.
7. Se non hai importato gli hash delle password, richiedi agli utenti importati di reimpostare le proprie password.

Altre risorse

- Esportazione dei profili utente di Cognito: architettura di riferimento per l'esportazione degli account utente tra pool di utenti

Argomenti

- Creazione del ruolo CloudWatch Logs IAM
- Creazione di un file CSV di importazione degli utenti
- Creazione ed esecuzione del processo di importazione del bacino d'utenza di Amazon Cognito

- Visualizzazione dei risultati di importazione del pool di utenti nella CloudWatch console
- Necessità degli utenti importati di ripristinare le password
- Importazione di utenti con hash di password

Creazione del ruolo CloudWatch Logs IAM

Se utilizzi la CLI o l'API di Amazon Cognito, devi creare un ruolo IAM. CloudWatch La procedura seguente descrive come creare un ruolo IAM che Amazon Cognito può utilizzare per scrivere i risultati del processo di importazione in Logs. CloudWatch

> **Note**
>
> Quando crei un processo di importazione nella console di Amazon Cognito, puoi creare il ruolo IAM contemporaneamente. Quando scegli l'operazione Create a new IAM role (Crea un nuovo ruolo IAM), Amazon Cognito applica automaticamente la policy di attendibilità e la policy IAM appropriate al ruolo.

Per creare il ruolo IAM di CloudWatch Logs per l'importazione del pool di utenti (AWS CLI, API)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Crea un nuovo ruolo IAM per un Servizio AWS. Per istruzioni dettagliate, consulta Creazione di un ruolo per un servizio Servizio AWS nella Guida per l'utente di AWS Identity and Access Management .

   a. Quando selezioni un Use Case (Caso d'uso) per Trusted entity type (Tipo di entità attendibile), scegli qualsiasi servizio. Amazon Cognito non è attualmente elencato nei casi d'uso dei servizi.

   b. Nella schermata Add permissions (Aggiungi autorizzazioni), scegli Create policy (Crea policy) e inserisci la seguente istruzione della policy. Sostituiscilo *`REGION`* con il tuo pool Regione AWS di utenti, ad esempio`us-east-1`. Sostituiscilo *`ACCOUNT`* con il tuo Account AWS ID, ad esempio`111122223333`.

      JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
cognito/*"
            ]
        }
    ]
}
```

3. Poiché non hai scelto Amazon Cognito come entità attendibile quando hai creato il ruolo, ora devi modificare manualmente la relazione di trust del ruolo. Seleziona Roles (Ruoli) nel pannello di navigazione della console IAM, quindi scegli il nuovo ruolo creato.
4. Seleziona la scheda Relazioni di attendibilità
5. Seleziona Modifica policy di attendibilità.
6. Incolla la seguente istruzione della policy in Edit trust policy (Modifica policy di attendibilità), sostituendo qualsiasi testo esistente:

   JSON

```
{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "cognito-idp.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    }
```

7. Scegli Update policy (Aggiorna policy).
8. Nota l'ARN del ruolo . Quando si crea processo di importazione, è necessario fornire l'ARN.

In alternativa, puoi creare lo stesso ruolo con la AWS CLI. I seguenti comandi creano un ruolo che Amazon Cognito può assumere, allegano una politica che consente la scrittura nei CloudWatch log e restituiscono il ruolo ARN. *REGION*Sostituiscilo con il tuo pool Regione AWS di utenti e *ACCOUNT* con il tuo ID. Account AWS

Per creare il ruolo CloudWatch Logs IAM per l'importazione del pool di utenti (AWS CLI)

1. Crea il ruolo con una politica di fiducia che consenta ad Amazon Cognito di assumerlo.

```
aws iam create-role \
    --role-name CognitoUserImportRole \
    --assume-role-policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": { "Service": "cognito-idp.amazonaws.com" },
                "Action": "sts:AssumeRole"
            }
        ]
    }'
```

2. Allega una politica che consenta al ruolo di scrivere i risultati di importazione in Logs. CloudWatch

```
aws iam put-role-policy \
    --role-name CognitoUserImportRole \
    --policy-name CognitoUserImportLogsPolicy \
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "logs:CreateLogGroup",
                    "logs:CreateLogStream",
                    "logs:PutLogEvents",
                    "logs:DescribeLogStreams"
```

```
                ],
                "Resource": "arn:aws:logs:REGION:ACCOUNT:log-group:/aws/cognito/*"
            }
        ]
    }'
```

3. Recupera il ruolo ARN da utilizzare quando crei *ROLE_ARN* il tuo processo di importazione.

```
aws iam get-role \
    --role-name CognitoUserImportRole \
    --query 'Role.Arn' --output text
```

Creazione di un file CSV di importazione degli utenti

Prima di poter importare gli utenti esistenti nel pool di utenti, è necessario creare un file CSV contenente gli utenti che si desidera importare e i relativi attributi. Dal pool di utenti, è possibile recuperare un file di importazione utente con intestazioni che riflettono lo schema di attributi del pool di utenti. È quindi possibile inserire le informazioni utente che soddisfano i requisiti di formattazione in Formattazione del file CSV.

Download dell'intestazione del file CSV (console)

Utilizza la seguente procedura per scaricare il file di intestazione CSV.

Per eseguire il download dell'intestazione del file CSV

1. Passa alla console Amazon Cognito. È possibile che ti vengano richieste le AWS credenziali.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Utenti.
5. Nella sezione Import users (Importa utenti), scegli Create an import job (Crea un processo di importazione).
6. In Upload CSV (Carica CSV), seleziona il link template.csv e scarica il file CSV.

Scaricamento dell'intestazione del file CSV (AWS CLI)

Per ottenere un elenco delle intestazioni corrette, dal menu Utenti in Importa utenti, seleziona Crea processo di importazione. Nella finestra di dialogo che segue, seleziona il `template.csv` link per scaricare un file modello con gli attributi del tuo pool di utenti.

Puoi anche eseguire il seguente comando CLI, *USER_POOL_ID* dov'è l'identificatore del pool di utenti per il pool di utenti in cui importerai gli utenti:

```
aws cognito-idp get-csv-header --user-pool-id "USER_POOL_ID"
```

Risposta di esempio:

```
{
    "CSVHeader": [
        "name",
        "given_name",
        "family_name",
        "middle_name",
        "nickname",
        "preferred_username",
        "profile",
        "picture",
        "website",
        "email",
        "email_verified",
        "gender",
        "birthdate",
        "zoneinfo",
        "locale",
        "phone_number",
        "phone_number_verified",
        "address",
        "updated_at",
        "cognito:mfa_enabled",
        "cognito:username"
    ],
    "UserPoolId": "USER_POOL_ID"
}
```

Formattazione del file CSV

L'aspetto del file di intestazione CSV di importazione degli utenti scaricato è il seguente: Include anche eventuali attributi personalizzati aggiunti al pool di utenti.

```
cognito:username,name,given_name,family_name,middle_name,nickname,preferred_username,profile,pi
```

Modificare il file CSV, in modo che includa questa intestazione e i valori di attributo per gli utenti. Il file viene formattato in base ai seguenti criteri:

> **Note**
>
> Per ulteriori informazioni sui valori di attributo, ad esempio il formato corretto per i numeri di telefono, consulta Utilizzo degli attributi utente.

- La prima riga nel file è la riga di intestazione scaricata, che contiene i nomi di attributo degli utenti.
- L'ordine delle colonne nel file CSV non è rilevante.
- Ogni riga, dopo la prima, contiene i valori di attributo per un utente.
- Tutte le colonne nell'intestazione devono essere presenti, ma non è necessario che tu fornisca valori in ogni colonna;
- I seguenti attributi sono necessari:
  - cognito:username
  - email_verified o phone_number_verified
    - Almeno uno degli attributi verificato automaticamente deve essere `true` per ogni utente. Un attributo verificato automaticamente è un indirizzo e-mail o un numero di telefono a cui Amazon Cognito invia automaticamente un codice quando un nuovo utente si unisce al pool di utenti.
    - Il pool di utenti deve avere almeno un attributo verificato automaticamente, email_verified o phone_number_verified. Se il bacino d'utenza non ha attributi verificati automaticamente, il processo di importazione non verrà avviato.
    - Se il bacino d'utenza ha un attributo verificato automaticamente, tale attributo deve essere verificato per ogni utente. Ad esempio, se il bacino d'utenza ha solo phone_number come attributo verificato automaticamente, il valore phone_number_verified deve essere `true` per ogni utente.

> **Note**
>
> Affinché gli utenti reimpostino le password, devono avere una e-mail o un numero di telefono verificati. Amazon Cognito invia un messaggio contenente un codice di reimpostazione della password all'e-mail o al numero di telefono specificati nel file CSV. Se il messaggio viene inviato al numero di telefono, è inviato tramite SMS. Per

ulteriori informazioni, consulta Verifica delle informazioni di contatto al momento della registrazione.

  - e-mail (se email_verified è `true`)
  - phone_number (se phone_number_verified è `true`)
  - Tutti gli attributi che hai contrassegnato come richiesto quando hai creato il bacino d'utenza
- I valori di attributo che sono stringhe dovrebbero non essere tra virgolette;
- Se il valore di un attributo contiene una virgola, è necessario inserire una barra rovesciata (\) prima della virgola. Questo perché i campi in un file CSV sono separati da virgole.
- Il contenuto del file CSV deve essere in UTF-8 formato senza l'ordine dei byte.
- Il campo cognito:username è obbligatorio e deve essere univoco all'interno del tuo bacino d'utenza. Può essere qualsiasi stringa Unicode. Tuttavia, non può contenere spazi o schede;
- I valori della data di nascita, se presenti, devono essere nel formato. *`mm/dd/yyyy`* Ciò significa, ad esempio, che una data di nascita corrispondente al 1 febbraio 1985 deve essere codificata come **`02/01/1985`**.
- Il campo cognito:mfa_enabled deve corrispondere ai requisiti MFA del tuo pool di utenti. Se hai impostato l'autenticazione a più fattori (MFA) come obbligatoria nel tuo pool di utenti, questo campo deve `true` essere o vuoto per tutti gli utenti. Se hai impostato la modalità MFA come disattivata, questo campo deve essere `false` o vuoto per tutti gli utenti. Un valore vuoto imposta MFA-enabled lo stato degli utenti importati sullo stato richiesto dal pool di utenti. È possibile importare utenti in un pool di MFA-required utenti senza un fattore MFA valido, indipendentemente dal fatto che si imposti un `cognito:mfa_enabled` valore. Gli utenti in questo stato hanno la MFA attiva ma non possono accedere finché non configurano un attributo e-mail, un attributo numero di telefono o un TOTP e tale configurazione non è un fattore MFA valido nel tuo pool di utenti.
- La lunghezza massima è 16.000 caratteri.
- Il file CSV può avere una dimensione massima di 100 MB.
- Il numero massimo di righe (utenti) nel file è 500.000. In questo valore massimo non è inclusa la riga di intestazione.
- Il valore del campo updated_at dovrebbe essere un periodo di tempo espresso in secondi, per esempio: **`1471453471`**.
- Qualsiasi spazio bianco iniziale o finale nel valore di un attributo viene tagliato.

L'elenco seguente è un esempio di file di importazione CSV per un pool di utenti senza attributi personalizzati. Lo schema del pool di utenti potrebbe differire da questo esempio. In tal caso, è necessario fornire valori di test nel modello CSV scaricato dal pool di utenti.

```
cognito:username,name,given_name,family_name,middle_name,nickname,preferred_username,profile,pi
John,,John,Doe,,,,,,,johndoe@example.com,TRUE,,02/01/1985,,,+12345550100,TRUE,123 Any
 Street,,FALSE
Jane,,Jane,Roe,,,,,,,janeroe@example.com,TRUE,,01/01/1985,,,+12345550199,TRUE,100 Main
 Street,,FALSE
```

Creazione ed esecuzione del processo di importazione del bacino d'utenza di Amazon Cognito

Questa sezione descrive come creare ed eseguire il processo di importazione del pool di utenti utilizzando la console Amazon Cognito e AWS Command Line Interface ()AWS CLI.

Argomenti

- Importazione di utenti da un file CSV (console)
- Importazione di utenti (AWS CLI)

Importazione di utenti da un file CSV (console)

Nella procedura seguente viene descritto come importare gli utenti dal file CSV.

Per importare gli utenti da un file CSV (console)

1. Passa alla console Amazon Cognito. È possibile che ti vengano richieste le credenziali AWS .
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Utenti.
5. Nella sezione Import users (Importa utenti), scegli Create an import job (Crea un processo di importazione).
6. Nella pagina Create import job (Crea processo di importazione), inserisci un Job name (Nome processo).
7. Scegli l'operazione Create a new IAM role (Crea un nuovo ruolo IAM) o Use an existing IAM role (Utilizza un ruolo IAM esistente).

    a. Se hai scelto Create a new IAM role (Crea un nuovo ruolo IAM), inserisci un nome per il nuovo ruolo. Amazon Cognito crea automaticamente un ruolo con le autorizzazioni e

il rapporto di fiducia corretti. Il principale IAM che crea il processo di importazione deve disporre delle autorizzazioni per creare ruoli IAM.

b. Se hai scelto Use an existing IAM role (Utilizza un ruolo IAM esistente), scegli un ruolo dall'elenco in IAM role selection (Selezione del ruolo IAM). Questo ruolo deve disporre delle autorizzazioni e della policy di attendibilità descritte in Creazione del ruolo CloudWatch Logs IAM.

8. In Carica CSV, scegli Scegli file e allega il file CSV che hai preparato.
9. Scegli Create job (Crea processo) per inviare il processo, ma avvialo in un secondo momento. Scegli Create and start job (Crea e avvia il processo) per inviare il processo e avviarlo immediatamente.
10. Se hai creato il processo ma non l'hai avviato, puoi avviarlo in un secondo momento. Nel menu Utenti, sotto Importa utenti, scegli il processo di importazione, quindi seleziona Avvia. Puoi anche inviare una richiesta StartUserImportJobAPI da un AWS SDK.
11. Monitora lo stato di avanzamento del processo di importazione degli utenti nel menu Utenti sotto Importa utenti. Se il processo non va a buon fine, puoi selezionare il valore Status (Stato). Per ulteriori dettagli, seleziona Visualizza CloudWatch i log per maggiori dettagli ed esamina eventuali problemi nella console CloudWatch Logs.

Importazione di utenti (AWS CLI)

I seguenti comandi della CLI sono disponibili per importare gli utenti in un bacino d'utenza:

- `create-user-import-job`
- `get-csv-header`
- `describe-user-import-job`
- `list-user-import-jobs`
- `start-user-import-job`
- `stop-user-import-job`

Per l'elenco delle opzioni della riga di comando per questi comandi, utilizza l'opzione `help` della riga di comando. Ad esempio:

```
aws cognito-idp get-csv-header help
```

Creazione di un processo di importazione degli utenti

Dopo aver creato il file CSV, crea un processo di importazione utenti eseguendo il seguente comando CLI, *JOB_NAME* dove è il nome che scegli per il processo*USER_POOL_ID*, l'ID del pool di utenti per il pool di utenti a cui vengono aggiunti i nuovi utenti *ROLE_ARN* e il ruolo ARN in cui hai ricevuto: Creazione del ruolo CloudWatch Logs IAM

```
aws cognito-idp create-user-import-job --job-name "JOB_NAME" --user-pool-id
 "USER_POOL_ID" --cloud-watch-logs-role-arn "ROLE_ARN"
```

Il valore *PRE_SIGNED_URL* restituito nella risposta è valido per 15 minuti. Trascorso tale periodo, scade ed è necessario creare un nuovo processo di importazione degli utenti per ottenere un nuovo URL.

Example risposta:

```
{
    "UserImportJob": {
        "Status": "Created",
        "SkippedUsers": 0,
        "UserPoolId": "USER_POOL_ID",
        "ImportedUsers": 0,
        "JobName": "JOB_NAME",
        "JobId": "JOB_ID",
        "PreSignedUrl": "PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn": "ROLE_ARN",
        "FailedUsers": 0,
        "CreationDate": 1470957431.965
    }
}
```

Valori di stato per un processo di importazione degli utenti

Nelle risposte ai comandi di importazione degli utenti, visualizzerai uno dei seguenti valori `Status`:

- `Created`: Il processo è stato creato, ma non avviato.
- `Pending`: Stato di transizione. Hai avviato il processo, ma questo non ha ancora iniziato a importare gli utenti;
- `InProgress`: Il processo è stato avviato e gli utenti sono in fase di importazione.
- `Stopping`: Hai interrotto il processo, ma questo non ha ancora cessato l'importazione degli utenti.

- `Stopped`: Hai interrotto il processo e quest'ultimo ha cessato l'importazione degli utenti.
- `Succeeded`: La fase ha avuto esito positivo.
- `Failed`: Il processo è stato interrotto a causa di un errore.
- `Expired`: Hai creato un processo, ma non lo hai avviato entro 24-48 ore. Tutti i dati associati al processo sono stati eliminati e il processo non può essere avviato.

Caricamento del file CSV

Utilizza il seguente comando `curl` per caricare il file CSV contenente i dati dell'utente nell'URL prefirmato che hai ottenuto dalla risposta del comando `create-user-import-job`.

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"
 "PRE_SIGNED_URL"
```

Nell'output di questo comando, cerca la frase `"We are completely uploaded and fine"`. Questa frase indica che il file è stato caricato correttamente. I tuoi pool di utenti non conservano le informazioni nei file di importazione dopo aver eseguito i processi di importazione. Una volta completati o scaduti, Amazon Cognito elimina il file CSV caricato.

Descrizione di un processo di importazione degli utenti

Per ottenere una descrizione del processo di importazione degli utenti, utilizza il seguente comando, dove si *`USER_POOL_ID`* trova l'ID del pool di utenti e *`JOB_ID`* l'ID del lavoro restituito quando hai creato il processo di importazione degli utenti.

```
aws cognito-idp describe-user-import-job --user-pool-id "USER_POOL_ID" --job-id
 "JOB_ID"
```

Example Risposta di esempio:

```
{
    "UserImportJob": {
        "Status": "Created",
        "SkippedUsers": 0,
        "UserPoolId": "USER_POOL_ID",
        "ImportedUsers": 0,
        "JobName": "JOB_NAME",
        "JobId": "JOB_ID",
```

```
        "PreSignedUrl": "PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn":"ROLE_ARN",
        "FailedUsers": 0,
        "CreationDate": 1470957431.965
    }
}
```

Nell'output di esempio precedente, *PRE_SIGNED_URL* è l'URL in cui hai caricato il file CSV. *ROLE_ARN*È l'ARN del ruolo CloudWatch Logs che hai ricevuto quando hai creato il ruolo.

Elenco dei tuoi processi di importazione degli utenti

Per elencare i tuoi processi di importazione degli utenti, utilizza il comando seguente:

```
aws cognito-idp list-user-import-jobs --user-pool-id "USER_POOL_ID" --max-results 2
```

Example Risposta di esempio:

```
{
    "UserImportJobs": [
        {
            "Status": "Created",
            "SkippedUsers": 0,
            "UserPoolId": "USER_POOL_ID",
            "ImportedUsers": 0,
            "JobName": "JOB_NAME",
            "JobId": "JOB_ID",
            "PreSignedUrl":"PRE_SIGNED_URL",
            "CloudWatchLogsRoleArn":"ROLE_ARN",
            "FailedUsers": 0,
            "CreationDate": 1470957431.965
        },
        {
            "CompletionDate": 1470954227.701,
            "StartDate": 1470954226.086,
            "Status": "Failed",
            "UserPoolId": "USER_POOL_ID",
            "ImportedUsers": 0,
            "SkippedUsers": 0,
            "JobName": "JOB_NAME",
            "CompletionMessage": "Too many users have failed or been skipped during the
 import.",
```

```
            "JobId": "JOB_ID",
            "PreSignedUrl":"PRE_SIGNED_URL",
            "CloudWatchLogsRoleArn":"ROLE_ARN",
            "FailedUsers": 5,
            "CreationDate": 1470953929.313
        }
    ],
    "PaginationToken": "PAGINATION_TOKEN"
}
```

I processi sono elencati in ordine cronologico dall'ultimo creato al primo. La *PAGINATION_TOKEN* stringa dopo il secondo processo indica che ci sono risultati aggiuntivi per questo comando list. Per elencare i risultati aggiuntivi, utilizza l' opzione `--pagination-token` come segue:

```
aws cognito-idp list-user-import-jobs --user-pool-id "USER_POOL_ID" --max-results 10 --
pagination-token "PAGINATION_TOKEN"
```

Avvio di un processo di importazione degli utenti

Per avviare un processo di importazione degli utenti, utilizza il comando seguente:

```
aws cognito-idp start-user-import-job --user-pool-id "USER_POOL_ID" --job-id "JOB_ID"
```

Può essere attivo soltanto un processo di importazione alla volta per account.

Example Risposta di esempio:

```
{
    "UserImportJob": {
        "Status": "Pending",
        "StartDate": 1470957851.483,
        "UserPoolId": "USER_POOL_ID",
        "ImportedUsers": 0,
        "SkippedUsers": 0,
        "JobName": "JOB_NAME",
        "JobId": "JOB_ID",
        "PreSignedUrl":"PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn": "ROLE_ARN",
        "FailedUsers": 0,
        "CreationDate": 1470957431.965
    }
```

```
}
```

Arresto di un processo di importazione degli utenti

Per arrestare un processo di importazione degli utenti mentre è in corso, utilizza il comando seguente. Una volta che hai interrotto il processo, non puoi riavviarlo.

```
aws cognito-idp stop-user-import-job --user-pool-id "USER_POOL_ID" --job-id "JOB_ID"
```

Example Risposta di esempio:

```
{
    "UserImportJob": {
        "CompletionDate": 1470958050.571,
        "StartDate": 1470958047.797,
        "Status": "Stopped",
        "UserPoolId": "USER_POOL_ID",
        "ImportedUsers": 0,
        "SkippedUsers": 0,
        "JobName": "JOB_NAME",
        "CompletionMessage": "The Import Job was stopped by the developer.",
        "JobId": "JOB_ID",
        "PreSignedUrl":"PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn": "ROLE_ARN",
        "FailedUsers": 0,
        "CreationDate": 1470957972.387
    }
}
```

Visualizzazione dei risultati di importazione del pool di utenti nella CloudWatch console

Puoi visualizzare i risultati del tuo processo di importazione nella CloudWatch console Amazon.

Argomenti

- Visualizzazione dei risultati
- Interpretazione dei risultati

Visualizzazione dei risultati

I seguenti passaggi descrivono come visualizzare i risultati dell'importazione del bacino d'utenza.

Visualizzazione dei risultati dell'importazione del bacino d'utenza

1. Accedi Console di gestione AWS e apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Scegliere Logs (Log).
3. Scegli il gruppo di log per i tuoi processi di importazione del bacino d'utenza. Il nome del gruppo di log è del tipo `/aws/cognito/userpools/`*`USER_POOL_ID`*`/`*`USER_POOL_NAME`*.
4. Scegli il log per il processo di importazione degli utenti che hai appena eseguito. Il nome del registro è nel formato*`JOB_ID`*/*`JOB_NAME`*. I risultati nel log si riferiscono ai i tuoi utenti per numero di linea. Nessun dato degli utenti viene scritto nel log. Per ogni utente, viene visualizzata una linea simile a quella riportata di seguito:

   - `[SUCCEEDED] Line Number 5956 - The import succeeded.`
   - `[SKIPPED] Line Number 5956 - The user already exists.`
   - `[FAILED] Line Number 5956 - The User Record does not set any of the auto verified attributes to true. (Example: email_verified to true).`

Interpretazione dei risultati

Lo stato degli utenti importati con successo è impostato su "PasswordReset».

Nei seguenti casi, l'utente non verrà importato, ma il processo di importazione proseguirà:

- Nessun attributo verificato automaticamente è impostato su `true`.
- I dati dell'utente non corrispondono allo schema;
- Non è stato possibile importare l'utente a causa di un errore interno.

Nei seguenti casi, il processo di importazione fallirà:

- Il ruolo Amazon CloudWatch Logs non può essere assunto, non ha la politica di accesso corretta o è stato eliminato.
- Il bacino d'utenza è stato eliminato.
- Amazon Cognito non è in grado di analizzare il file .csv.

Necessità degli utenti importati di ripristinare le password

Se il tuo pool di utenti offre solo l'accesso basato su password, gli utenti devono reimpostare le password dopo l'importazione. La prima volta che accedono, possono inserire qualsiasi password. Amazon Cognito richiede loro di inserire una nuova password nella risposta API alla richiesta di accesso della tua applicazione.

Se il tuo pool di utenti dispone di fattori di autenticazione senza password, Amazon Cognito utilizza per impostazione predefinita quelli per gli utenti importati. Non viene loro richiesta una nuova password e possono accedere immediatamente con un'e-mail o un SMS OTP senza password. Puoi anche richiedere agli utenti di impostare una password in modo che possano completare altri metodi di accesso come nome utente-password e passkey. Le seguenti condizioni si applicano all'accesso senza password dopo l'importazione dell'utente.

1. È necessario importare gli utenti con un attributo che corrisponda a un fattore di accesso senza password disponibile. Se gli utenti possono accedere con un indirizzo e-mail, è necessario importare un attributo. `email` Se si tratta di un numero di telefono, è necessario importare un `phone_number` attributo. Se entrambi, importa un valore per entrambi gli attributi.
2. Normalmente, gli utenti effettuano l'importazione in uno `RESET_REQUIRED` stato in cui devono reimpostare la password. Se vengono importati con la possibilità di accedere con un fattore senza password, Amazon Cognito imposta il loro stato su. `CONFIRMED`

Per ulteriori informazioni sull'autenticazione senza password, incluso come configurarla e come costruire il flusso di autenticazione nella tua applicazione, consulta. Autenticazione con pool di utenti Amazon Cognito

La procedura seguente descrive l'esperienza utente in un meccanismo di accesso personalizzato con utenti locali `RESET_REQUIRED` dopo l'importazione di un file CSV. Se i tuoi utenti accedono con accesso gestito, chiedi loro di selezionare la password dimenticata? opzione, fornisci il codice contenuto nell'e-mail o nel messaggio di testo e imposta una password.

Necessità degli utenti importati di ripristinare le password

1. Nell'app, tenta di accedere silenziosamente per l'utente corrente con `InitiateAuth` utilizzando una password casuale.
2. Amazon Cognito restituisce `NotAuthorizedException` quando `PreventUserExistenceErrors` è abilitato. In caso contrario, restituisce `PasswordResetRequiredException`.

3. L'app effettua una richiesta API `ForgotPassword` e reimposta la password dell'utente.

    a. L'app invia il nome utente in una richiesta API `ForgotPassword`.

    b. Amazon Cognito invia un codice all'indirizzo e-mail o al numero di telefono verificato. La destinazione dipende dai valori forniti per `email_verified` e `phone_number_verified` nel file CSV. La risposta alla richiesta `ForgotPassword` indica la destinazione del codice.

    > **Note**
    >
    > Il pool di utenti deve essere configurato per verificare le e-mail o i numeri di telefono. Per ulteriori informazioni, consulta Registrazione e conferma degli account utente.

    c. L'app visualizza un messaggio all'utente per verificare la posizione di invio del codice e richiede all'utente di inserire il codice e una nuova password.

    d. L'utente inserisce il codice e la nuova password nell'app;

    e. L'app invia il codice e la nuova password in una richiesta API `ConfirmForgotPassword`.

    f. L'app reindirizza l'utente per l'accesso.

Importazione di utenti con hash di password

Quando esegui la migrazione degli utenti da un sistema di autenticazione esistente ad Amazon Cognito, puoi importare gli hash delle loro password insieme agli attributi utente. I tuoi utenti possono quindi accedere immediatamente con le password esistenti, senza dover reimpostare la password. Dopo che gli utenti accedono per la prima volta, Amazon Cognito migra in modo trasparente le proprie credenziali al sistema di autenticazione nativo di Amazon Cognito.

Come funziona l'importazione di hash delle password

Con l'importazione dell'hash delle password, puoi migrare gli utenti preservando la loro capacità di accedere con le password esistenti. Il processo di importazione funziona nel modo seguente:

1. Si crea un processo di importazione e si specifica l'algoritmo di hashing delle password utilizzato dal sistema di origine per creare gli hash delle password.
2. Includi la `password_hash` colonna nel tuo file CSV con il valore hash della password di ogni utente.
3. Amazon Cognito importa gli utenti con gli hash delle password e li crea con uno stato. `CONFIRMED` Gli utenti possono accedere immediatamente.

4. Quando un utente accede per la prima volta, Amazon Cognito verifica la password in base all'hash importato. Una volta completata la verifica, Amazon Cognito migra le credenziali dell'utente al sistema di autenticazione nativo.
5. Tutti gli accessi successivi utilizzano il sistema di autenticazione nativo di Amazon Cognito.

> **Important**
>
> Tutti gli hash delle password in un singolo processo di importazione devono utilizzare lo stesso algoritmo. L'algoritmo viene specificato a livello di processo quando si crea il processo di importazione.

> **Note**
>
> Fino a quando un utente non completa il primo accesso e Amazon Cognito non migra le proprie credenziali, non puoi utilizzare l'autenticazione Secure Remote Password (SRP) per quell'utente. Utilizza i nostri `ADMIN_USER_PASSWORD_AUTH` flussi `USER_PASSWORD_AUTH` per gli utenti con hash di password importati che non hanno ancora effettuato l'accesso.

Algoritmi di hashing delle password supportati

Amazon Cognito supporta i seguenti algoritmi di hashing delle password per l'importazione:

`BCRYPT`

La funzione hash adattiva bcrypt. Amazon Cognito estrae tutti i parametri richiesti (sale, fattore di costo) dalla stringa hash.

Format: `$2<a/b/x/y>$[cost]$[22-char salt][31-char hash]`

Esempio: `$2b$10$CtA.Rcu/szzn9U00wpUjOuN3vrgJRZycv4aOzcP3GzqzO8UDPEFq6`

Fattore di costo massimo: 12

`SCRYPT`

La funzione di derivazione delle chiavi basata su password di scrypt. Amazon Cognito estrae tutti i parametri richiesti dalla stringa hash.

Format: `N$r$p$hexSalt$hexHash`

Esempio: `65536$8$1$304dbaef7c5e828dc19c98f0600d18fe$4f69c498c12cd102d057356facf8d77e8d42407090491ea32c5b038f5a18c099`

Parametri massimi: N (CPU/memory costo) = 65536, r (dimensione del blocco) = 8, p (parallelismo) = 1

`ARGON2ID`

L'algoritmo di hashing delle password Argon2id. Amazon Cognito estrae tutti i parametri richiesti dalla stringa hash.

Format: `$argon2id$v=N$m=M,t=T,p=P$salt$hash`

Esempio: `$argon2id$v=19$m=19456,t=2,p=1$ko/G5o1ms+ML08P95sQ8DA$AkVbvWSOqz7Hs3qthhWKxicOWnGLN+MBmpwc3emi5VA`

Parametri massimi: m (memoria in KB) = 19456, t (iterazioni) = 2, p (parallelismo) = 1

`PBKDF2_SHA256`

Password-Based Funzione di derivazione chiave 2 con. SHA-256 Amazon Cognito estrae tutti i parametri richiesti dalla stringa hash.

Format: `$pbkdf2-sha256$iterations$salt$hash`

Esempio: `$pbkdf2-sha256$600000$1XZlmwLQ2hhM3JYuCPiArQ$Pfheg9Zi/v5lXU4yyLA0WFUYEd/rlaVbzrM9oMD6IrA`

Iterazioni massime: 600000

> **Note**
>
> Tutti gli algoritmi supportati sono autodescrittivi, il che significa che Amazon Cognito può estrarre tutti i parametri richiesti (come sale, fattore di costo e iterazioni) direttamente dalla stringa hash. Devi solo specificare il nome dell'algoritmo quando crei il processo di importazione.

Se l'hash della password ha valori di parametro che superano i limiti massimi sopra elencati, l'importazione non riesce per quell'utente. Controlla la configurazione dell'hashing delle password del sistema di origine prima dell'importazione per garantire la compatibilità.

Aggiungere gli hash delle password al file CSV

Quando scarichi il modello CSV per l'importazione da parte dell'utente, il modello include una colonna. `password_hash` Compila questa colonna con i valori hash delle password per gli utenti che desideri importare con le password esistenti.

Regole di formattazione CSV per l'importazione degli hash delle password

- La `password_hash` colonna è obbligatoria quando il processo di importazione specifica un algoritmo di hashing delle password. Se lasci il valore vuoto per un utente specifico, Amazon Cognito importa quell'utente senza password e imposta l'utente sullo stato. `RESET_REQUIRED`
- Se il processo di importazione specifica un algoritmo di hashing delle password ma il file CSV non include una `password_hash` colonna, il processo ha esito negativo.
- Se il valore hash di una password non è valido o non corrisponde al formato previsto per l'algoritmo specificato, l'importazione non riesce per quell'utente. Amazon Cognito non crea l'utente e registra l'errore nei log di Amazon CloudWatch Logs.
- I valori hash delle password fanno distinzione tra maiuscole e minuscole. Assicurati che corrispondano al formato esatto previsto dall'algoritmo.

Example Esempio di CSV con hash delle password

L'esempio seguente mostra un file CSV che importa utenti con hash di password bcrypt:

```
cognito:username,email,email_verified,phone_number,phone_number_verified,password_hash
alejandro_rosalez,alejandro_rosalez@example.com,TRUE,+12345550100,TRUE,$2b$10$CtA.Rcu/
szzn9U00wpUjOuN3vrgJRZycv4aOzcP3GzqzO8UDPEFq6
mary_major,mary_major@example.com,TRUE,+12345550199,TRUE,$2b$10$CtA.Rcu/
szzn9U00wpUjOuN3vrgJRZycv4aOzcP3GzqzO8UDPEFq6
```

Creazione di un processo di importazione con hash di password (AWS CLI)

Per importare utenti con hash di password, è necessario specificare l'algoritmo di hashing delle password quando si crea il processo di importazione. Tutti gli utenti del processo di importazione devono utilizzare lo stesso algoritmo.

Utilizzate il AWS Command Line Interface comando seguente per creare un processo di importazione con hash di password. Il `--password-hashing-algorithm` parametro specifica l'algoritmo utilizzato per creare gli hash delle password nel file CSV.

```
aws cognito-idp create-user-import-job \
    --job-name "JOB_NAME" \
    --user-pool-id "USER_POOL_ID" \
    --cloud-watch-logs-role-arn "ROLE_ARN" \
    --password-hashing-algorithm BCRYPT
```

Sostituisci *BCRYPT* con uno degli algoritmi supportati:`BCRYPT`,, `SCRYPT` o. `ARGON2ID` `PBKDF2_SHA256`

Example Risposta di esempio

```
{
    "UserImportJob": {
        "Status": "Created",
        "SkippedUsers": 0,
        "UserPoolId": "USER_POOL_ID",
        "ImportedUsers": 0,
        "JobName": "JOB_NAME",
        "JobId": "JOB_ID",
        "PreSignedUrl": "PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn": "ROLE_ARN",
        "PasswordHashingAlgorithm": "BCRYPT",
        "FailedUsers": 0,
        "CreationDate": 1470957431.965
    }
}
```

La risposta include il `PasswordHashingAlgorithm` campo che conferma l'algoritmo specificato. Carica il file CSV nell'URL predefinito e avvia il processo come descritto in. Caricamento del file CSV

Esperienza utente dopo l'importazione dell'hash della password

Gli utenti importati con gli hash delle password hanno la seguente esperienza:

- Accesso immediato: gli utenti possono accedere con le password esistenti subito dopo l'importazione. Gli utenti non ricevono una richiesta di reimpostazione delle password.
- Stato utente: Amazon Cognito crea questi utenti con uno `CONFIRMED` stato invece di. `RESET_REQUIRED`
- Migrazione trasparente: quando gli utenti accedono per la prima volta, Amazon Cognito verifica la password in base all'hash importato. Una volta completata la verifica, Amazon Cognito migra

le credenziali al sistema di autenticazione nativo. Tutti gli accessi successivi utilizzano il sistema nativo.

- Latenza di accesso iniziale: il primo accesso per gli utenti con hash di password importati potrebbe richiedere leggermente più tempo rispetto agli accessi successivi. Amazon Cognito deve verificare la password rispetto all'hash importato e migrare le credenziali al sistema di autenticazione nativo.
- Reimpostazione della password: se un utente reimposta la password prima del primo accesso, l'hash della password importato viene sostituito con una nuova password utilizzando il sistema di autenticazione nativo di Amazon Cognito.

Risoluzione degli errori di importazione dell'hash delle password

Se l'importazione dell'hash della password non riesce per un utente, Amazon Cognito registra l'errore nei log di CloudWatch Amazon Logs. Gli scenari di errore più comuni includono:

Hash malformato

L'hash della password non corrisponde al formato previsto per l'algoritmo specificato. Verifica che il formato hash sia corretto e corrisponda a uno dei formati elencati in. Algoritmi di hashing delle password supportati

Parametro fuori dai limiti

L'hash della password contiene valori di parametro (come il fattore di costo o le iterazioni) che superano i valori massimi consentiti. Per gli utenti con hash incompatibili, scegliete una delle seguenti opzioni:

- Importa quegli utenti senza l'hash della password lasciando il `password_hash` campo vuoto. Amazon Cognito li imposta sullo `RESET_REQUIRED` stato e devono reimpostare le password al primo accesso.
- Utilizza un trigger Lambda per la migrazione degli utenti per migrare tali utenti al momento dell'accesso anziché tramite l'importazione in formato CSV.
- Prima di disattivare il sistema di origine, esegui nuovamente il hash delle password con parametri compatibili quando gli utenti accedono al sistema esistente.

Mancata corrispondenza dell'algoritmo

L'hash della password è stato creato con un algoritmo diverso da quello specificato nel processo di importazione. Tutti gli hash di un singolo processo di importazione devono utilizzare lo stesso algoritmo. Se hai utenti con algoritmi diversi, crea processi di importazione separati per ogni algoritmo.

Colonna password_hash mancante

Hai specificato un algoritmo di hashing delle password durante la creazione del lavoro, ma il file CSV non include la colonna. `password_hash` Aggiungi la colonna al tuo file CSV o crea un nuovo lavoro senza specificare un algoritmo.

Quando l'importazione dell'hash di una password non riesce per un utente, Amazon Cognito non crea l'utente. Gli altri utenti coinvolti nel processo di importazione continuano a essere elaborati. Controlla i log di Amazon CloudWatch Logs per identificare gli utenti che hanno fallito e il motivo di ogni errore.

Considerazioni sulla sicurezza per l'importazione di hash delle password

Amazon Cognito applica misure di sicurezza aggiuntive agli hash delle password importati:

- Doppio hashing: Amazon Cognito esegue nuovamente l'hash di tutte le password importate con un ulteriore livello di protezione crittografica prima dell'archiviazione, indipendentemente dalla forza dell'algoritmo originale.
- Migrazione automatica: dopo il successo della prima autenticazione, Amazon Cognito migra le credenziali dell'utente al protocollo Secure Remote Password (SRP) nativo utilizzato da Amazon Cognito. Ciò garantisce che tutti gli utenti utilizzino alla fine il meccanismo di autenticazione nativo ad alta sicurezza.
- Pulizia dei dati: una volta completato il processo di importazione, Amazon Cognito rimuove il file CSV caricato contenente gli hash delle password dallo spazio di archiviazione temporaneo.
- Politica sulle password non applicata: Amazon Cognito non applica la politica di password che hai configurato per il pool di utenti quando importi utenti con hash di password. Se una password non soddisfa i criteri in materia di password che hai configurato, Amazon Cognito importa comunque l'utente. Per rafforzare la sicurezza delle password e rifiutare le password che non soddisfano la politica, convalida la sicurezza delle password nel sistema di origine prima di generare hash per l'importazione.

> **Important**
>
> Tratta il file CSV contenente gli hash delle password come dati sensibili. Proteggilo in transito ed eliminalo in modo sicuro dopo il completamento dell'importazione.

Utilizzo degli attributi utente

Gli attributi sono informazioni che ti aiutano a identificare i singoli utenti, ad esempio nome, indirizzo e-mail e numero di telefono. Un nuovo bacino d'utenza include un set di attributi standard di default. È inoltre possibile aggiungere attributi personalizzati alla definizione del pool di utenti in Console di gestione AWS. Questo argomento descrive questi attributi nel dettaglio e fornisce suggerimenti su come configurare il bacino d'utenza.

Non tutte le informazioni sugli utenti devono essere archiviate negli attributi. Ad esempio, i dati utente che cambiano spesso, come le statistiche di utilizzo o i punteggi dei giochi, devono essere conservati in un archivio dati separato, come Amazon Cognito Sync o Amazon DynamoDB.

Disinfetta gli input per i valori delle stringhe degli attributi utente prima di inviarli al tuo pool di utenti. Un metodo per analizzare i valori degli attributi utente proposti consiste nell'utilizzare un trigger Lambda come la pre-registrazione.

> **Note**
>
> In alcuni documenti e standard si fa riferimento agli attributi come membri.

Argomenti

- Standard attributes (Attributi standard)
- Nomi utente e nomi utente preferiti
- Personalizzazione degli attributi di accesso
- Attributi personalizzati
- Autorizzazioni attributo e ambiti

Standard attributes (Attributi standard)

Amazon Cognito assegna a tutti gli utenti un set di attributi standard in base alla specifica OpenID Connect. Per impostazione predefinita, i valori degli attributi standard e personalizzati possono essere costituiti da qualsiasi stringa composta da un massimo di 2.048 caratteri, ma alcuni valori di attributo hanno restrizioni di formato.

Gli attributi standard sono:

- name
- family_name
- given_name
- middle_name
- nickname
- preferred_username
- profile
- picture
- website
- gender
- birthdate
- zoneinfo
- locale
- updated_at
- address
- email
- phone_number
- sub

Ad eccezione di sub, per impostazione predefinita, gli attributi standard sono opzionali per tutti gli utenti. Per rendere obbligatorio un attributo, durante il processo di creazione del pool di utenti, seleziona la casella di controllo Richiesto accanto all'attributo. Amazon Cognito assegna un valore identificativo utente unico all'attributo sub di ciascun utente. È possibile verificare solo gli attributi email e phone_number.

Gli attributi standard hanno proprietà predefinite che è possibile visualizzare nel SchemaAttributes parametro di una risposta DescribeUserPool API. È possibile impostare valori personalizzati per queste proprietà degli attributi, come il tipo di dati, la mutabilità e i vincoli di lunghezza. Per modificare le proprietà degli attributi standard, impostate i relativi valori personalizzati nel parametro Schema. CreateUserPool Lo schema è anche il luogo in cui si impostano gli attributi richiesti. Non puoi modificare le proprietà degli attributi standard quando crei pool di utenti nella console Amazon Cognito.

> **Note**
>
> Quando contrassegni un attributo standard come Required (Obbligatorio), un utente non può effettuare la registrazione, a meno che non fornisca un valore per l'attributo. Per creare utenti e non fornire valori per gli attributi obbligatori, gli amministratori possono utilizzare l'AdminCreateUserAPI. Dopo aver creato il bacino d'utenza, non è più possibile cambiare un attributo da obbligatorio a non obbligatorio e viceversa.

Dettagli degli attributi standard e restrizioni di formato

birthdate

Il valore deve essere una data valida di 10 caratteri nel formato YYYY-MM-DD.

e-mail

Utenti e amministratori possono verificare i valori dell'indirizzo e-mail.

Un amministratore con Account AWS le autorizzazioni appropriate può modificare l'indirizzo e-mail dell'utente e contrassegnarlo come verificato. Contrassegna un indirizzo email come verificato con l'AdminUpdateUserAttributesAPI o il comando AWS Command Line Interface admin-update-user-attributes ().AWS CLI Con questo comando, l'amministratore può impostare l'attributo `email_verified` su `true`. Puoi anche modificare un utente nel menu Utenti della console Amazon Cognito per contrassegnare un indirizzo e-mail come verificato.

Il valore deve essere una stringa di indirizzo e-mail valida conforme al formato e-mail standard con il simbolo @ e il dominio, lunga fino a 2048 caratteri.

phone_number

Un utente deve fornire un numero di telefono se è attiva l'autenticazione a più fattori (MFA) con SMS. Per ulteriori informazioni, consulta Aggiunta dell'autenticazione MFA a un bacino d'utenza.

Utenti e amministratori possono verificare i valori del numero di telefono.

Un amministratore con Account AWS le autorizzazioni appropriate può modificare il numero di telefono dell'utente e contrassegnarlo come verificato. Contrassegna un numero di telefono come verificato con l'AdminUpdateUserAttributesAPI o il comando AWS CLI admin-update-user-attributes. Con questo comando, l'amministratore può impostare l'attributo `phone_number_verified` su `true`. Puoi anche modificare un utente nel menu Utenti della console Amazon Cognito per contrassegnare un numero di telefono come verificato.

> **Important**
>
> I numeri di telefono devono attenersi alle seguenti regole di formattazione: devono iniziare con un segno più (**+**), seguito immediatamente dal codice del paese. Un numero di telefono può contenere solo il segno **+** e le cifre. Prima di inviare il valore al servizio, rimuovi dal numero di telefono qualsiasi altro carattere, come parentesi, spazi o trattini (**-**). Un numero di telefono degli Stati Uniti ad esempio, deve seguire questo formato: **+14325551212**.

preferred_username

È possibile selezionare `preferred_username` come obbligatorio o come alias, ma non entrambi. Se si `preferred_username` tratta di un alias, puoi effettuare una richiesta all'operazione UpdateUserAttributesAPI e aggiungere il valore dell'attributo dopo aver confermato l'utente.

sub

Indicizza e cerca i tuoi utenti in base all'attributo `sub`. L'attributo `sub` è un identificativo utente unico all'interno di ciascun pool di utenti. Gli utenti possono modificare attributi come `phone_number` e `email`. L'attributo `sub` ha un valore fisso. Per ulteriori informazioni sull'esito di utenti, consultare Gestione e ricerca degli account utente.

> **Important**
>
> Amazon Cognito genera `sub` in un Cognito-specific formato Amazon non conforme a un formato UUID specifico, incluso l'UUID RFC. Non dovresti convalidare rigorosamente il formato di. `sub`

Visualizza gli attributi obbligatori

Utilizza la procedura seguente per visualizzare gli attributi obbligatori per un determinato bacino d'utenza.

> **Note**
>
> Non è possibile modificare gli attributi obbligatori dopo che il bacino d'utenza è stato creato.

Visualizzare gli attributi obbligatori

1. Vai ad Amazon Cognito in. Console di gestione AWS Se la console te lo richiede, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu. Sign-up
5. Visualizza gli attributi obbligatori del bacino d'utenza nella sezione Required attributes (Attributi obbligatori).

Nomi utente e nomi utente preferiti

Il valore `username` è un attributo separato ed è diverso dall'attributo `name`. Ogni utente ha un attributo `username`. Amazon Cognito genera automaticamente un nome utente per gli utenti federati. È necessario fornire un attributo `username` per creare un utente locale nella directory Amazon Cognito. Dopo aver creato un utente, non puoi più modificare il valore dell'attributo `username`.

Gli sviluppatori possono utilizzare l'attributo `preferred_username` per fornire agli utenti un nome utente che possano modificare. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

Se l'applicazione non richiede un nome utente, non è necessario chiedere agli utenti di fornirne uno. L'app è in grado di creare un nome utente univoco per gli utenti in background. Questa funzione è utile se, ad esempio, desideri che gli utenti si registrino e accedano con un indirizzo e-mail e una password. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

Il `username` deve essere univoco all'interno di un bacino d'utenza. Un `username` può essere riutilizzato, ma solo dopo che è stato eliminato e non è più in uso. Per informazioni sui vincoli di stringa agli `username` attributi, consulta la proprietà username di una richiesta SignUpAPI.

Personalizzazione degli attributi di accesso

Quando crei un pool di utenti, puoi configurare attributi del nome utente se gli utenti devono essere in grado di effettuare la registrazione e l'accesso utilizzando un indirizzo e-mail o un numero di telefono come il rispettivo nome utente. In alternativa, è possibile configurare attributi alias per fornire agli utenti la possibilità di includere più attributi quando effettuano la registrazione e quindi l'accesso con un nome utente, un nome utente preferito, un indirizzo e-mail o un numero di telefono.

> **Important**
>
> Dopo aver creato il bacino d'utenza, non puoi più cambiare questa impostazione.

Come scegliere tra attributi alias e attributi nome utente

Il tuo requisito	Attributi alias	Attributi nome utente
Gli utenti dispongono di più attributi di accesso	Sì[1]	No[2]
Gli utenti devono verificare l'indirizzo e-mail o il numero di telefono prima di poter accedere con esso	Sì	No
Registra gli utenti con indirizzi e-mail o numeri di telefono duplicati e previeni `UsernameExistsException` gli errori[3]	Sì	No
Può assegnare lo stesso valore di attributo dell'indirizzo e-mail o del numero di telefono a più di un utente	Sì[4]	No

[1] Gli attributi di accesso disponibili sono nome utente, indirizzo e-mail, numero di telefono e nome utente preferito.

[2] Può accedere utilizzando l'indirizzo e-mail o il numero di telefono.

[3] Il tuo pool di utenti non genera `UsernameExistsException` errori quando gli utenti si registrano con indirizzi e-mail o numeri di telefono potenzialmente duplicati, ma senza nome utente. Questo comportamento è indipendente da Previeni errori di presenza degli utenti, che si applica alle operazioni di accesso, ma non di registrazione.

[4] Solo l'ultimo utente che ha verificato l'attributo può utilizzarlo per l'accesso.

Opzione 1: attributi di accesso multipli (attributi alias)

Un attributo è un alias quando gli utenti hanno un nome utente ma possono anche accedere con quell'attributo. Configura gli alias quando desideri consentire agli utenti di scegliere tra il loro nome utente e altri valori di attributo nel campo nome utente del modulo di accesso. L'`username`attributo è un valore fisso che gli utenti non possono modificare. Se contrassegni un attributo come alias, gli utenti possono effettuare l'accesso usando l'attributo al posto del nome utente. Puoi contrassegnare come alias gli attributi dell'indirizzo e-mail, del numero di telefono e del nome utente preferito. Ad esempio, se contrassegni l'e-mail e il numero telefono come alias per un pool di utenti, gli utenti di quel pool di utenti possono effettuare l'accesso utilizzando il proprio nome utente, indirizzo e-mail o numero di telefono, insieme alla password.

Per scegliere gli attributi alias, seleziona User name (Nome utente) e almeno un'opzione di accesso aggiuntiva quando crei il pool di utenti.

> **Note**
>
> Se il bacino d'utenza è configurato per non applicare la distinzione tra maiuscole e minuscole, un utente può utilizzare lettere minuscole o maiuscole per registrarsi o accedere con l'alias. Per ulteriori informazioni, consulta il riferimento CreateUserPoolall'API dei pool di utenti di Amazon Cognito.

Se selezioni l'indirizzo e-mail come alias, Amazon Cognito non accetta un nome utente corrispondente a un formato di indirizzo e-mail valido. Analogamente, se selezioni un numero di telefono come alias, Amazon Cognito non accetta per il pool di utenti un nome utente che corrisponde a un formato di numero di telefono valido.

> **Note**
>
> I valori alias devono essere univoci in un bacino d'utenza. Se configuri un alias per un indirizzo e-mail o un numero di telefono, il valore che fornisci può essere nello stato verificato in un solo account. Se durante la registrazione l'utente fornisce un indirizzo e-mail o un numero di telefono come valore alias e un altro utente ha già utilizzato quel valore alias, la registrazione viene completata. Tuttavia, quando un utente cerca di confermare l'account con l'indirizzo e-mail (o il numero di telefono) e immette il codice valido, Amazon Cognito restituisce un errore `AliasExistsException`. L'errore indica all'utente che un account con l'indirizzo e-mail (o il numero di telefono) specificato è già esistente. A questo punto,

l'utente può abbandonare la creazione del nuovo account e tentare di ripristinare la password dell'account precedente. Se l'utente continua a creare il nuovo account, l'app deve chiamare l'API `ConfirmSignUp` con l'opzione `forceAliasCreation`. `ConfirmSignUp` con `forceAliasCreation` sposta l'alias dall'account precedente all'account appena creato e contrassegna l'attributo come non verificato nell'account precedente.
Gli alias possono anche essere trasferiti tra account quando un utente verifica un attributo con. `VerifyUserAttribute` Se un utente verifica un indirizzo e-mail o un numero di telefono che è già un alias verificato su un altro account, Amazon Cognito trasferisce l'alias sul suo account e contrassegna l'attributo come non verificato sull'account originale. Questo trasferimento non genera un. `AliasExistsException`

I numeri di telefono e gli indirizzi e-mail diventano alias attivi per un utente solo dopo che vengono verificati dall'utente. Ti consigliamo di scegliere la verifica automatica degli indirizzi e-mail e dei numeri di telefono se li utilizzi come alias.

Scegli gli attributi alias per evitare errori `UsernameExistsException` negli attributi dell'indirizzo e-mail e del numero di telefono quando gli utenti effettuano la registrazione.

Attiva l'attributo `preferred_username` in modo che l'utente possa modificare il nome utente che utilizza per accedere quando il valore dell'attributo `username` non cambia. Se desideri configurare questa esperienza utente, invia il nuovo valore `username` come `preferred_username` e scegli il `preferred_username` come alias. Gli utenti possono in questo modo effettuare l'accesso inserendo il nuovo valore. Se selezioni `preferred_username` come alias, gli utenti possono fornire il valore solo quando confermano l'account e non durante la registrazione.

Quando l'utente effettua la registrazione con un nome utente, puoi scegliere se può accedere con uno o più dei seguenti alias.

- Un indirizzo e-mail verificato
- Un numero di telefono verificato
- Il nome utente preferito

Questi alias possono essere modificati dall'utente dopo che ha effettuato la registrazione.

> **Important**
>
> Se il tuo pool di utenti supporta l'accesso con alias e desideri autorizzare o cercare un utente, non identificarlo in base a nessuno dei suoi attributi di accesso. L'identificatore utente a valore fisso `sub` è l'unico indicatore coerente dell'identità dell'utente.

Includi i passaggi seguenti quando crei il bacino d'utenza in modo che gli utenti possano accedere con un alias.

Phone number or email address (console)

È necessario impostare l'indirizzo e-mail e il numero di telefono come attributi di alias quando si crea un pool di utenti.

Per creare un pool di utenti con alias di nome utente nella console Amazon Cognito

1. Passa ad Amazon Cognito nella Console di gestione AWS. Se la console te lo richiede, inserisci le tue credenziali. AWS
2. Crea un nuovo pool di utenti con il pulsante Inizia o Crea pool di utenti.
3. Scegli le impostazioni dell'applicazione in Definisci la tua applicazione.
4. In Configura le opzioni in Opzioni per gli identificatori di accesso, seleziona la casella di controllo accanto a Nome utente e almeno una delle altre opzioni, Email e Numero di telefono.
5. Scegli gli attributi dell'alias come Attributi obbligatori per l'iscrizione. Nel modulo di registrazione con accesso gestito, Amazon Cognito richiede ai nuovi utenti di fornire valori per gli attributi obbligatori.
6. In Aggiungi un URL di ritorno, configura un URL di callback dell'applicazione per il reindirizzamento dopo l'accesso gestito.
7. Scegli Crea.

Phone number or email address (API/SDK)

Crea un nuovo pool di utenti con l'CreateUserPooloperazione API. Configura il `AliasAttributes` parametro come mostrato. Puoi rimuovere la `email` voce se desideri solo gli alias dei numeri di telefono o rimuovere la `phone_number` voce se desideri solo gli alias degli indirizzi e-mail.

```
"AliasAttributes": [
   "email",
   "phone_number"
],
```

Preferred username (API/SDK)

La console Amazon Cognito crea pool di utenti senza `preferred_username` alias. Per creare pool di utenti con un `preferred_username` alias, configura pool di utenti con richieste CreateUserPoolAPI in un SDK. AWS Per supportare la creazione di attributi di nome utente preferiti al momento dell'iscrizione, impostalo `preferred_username` come attributo obbligatorio. Nel modulo di registrazione con accesso gestito, Amazon Cognito richiede ai nuovi utenti di fornire valori per gli attributi obbligatori. Puoi impostarlo `preferred_username` come attributo obbligatorio nella console Amazon Cognito, ma ciò non lo rende disponibile come alias.

Configura come alias

Configura `preferred_username` come alias nel `AliasAttributes` parametro di una `CreateUserPool` richiesta come mostrato. Rimuovi dall'elenco tutti i valori che non desideri utilizzare come attributi di alias.

```
"AliasAttributes": [
   "email",
   "phone_number",
   "preferred_username"
],
```

Configura come richiesto

Nel modulo di registrazione con accesso gestito, Amazon Cognito richiede ai nuovi utenti di fornire valori per gli attributi obbligatori. Configura `preferred_username` come richiesto nel `SchemaAttributes` parametro di una richiesta. CreateUserPool

Per impostare il nome utente preferito come attributo obbligatorio, configuralo come mostrato. L'esempio seguente modifica lo schema predefinito di `preferred_username` per impostarlo come richiesto. Altri parametri dello schema come `AttributeDataType` (il valore predefinito è`string`) e `StringAttributeConstraints` (la lunghezza predefinita è compresa tra 1 e 99 caratteri) assumono valori predefiniti.

```
"Schema": [
```

```
   {
      "Name": "preferred_username",
      "Required": true
   }
]
```

Opzione 2: indirizzo e-mail o numero di telefono come attributo di accesso (attributi nome utente)

Quando l'utente si registra con un indirizzo e-mail o un numero di telefono come nome utente, puoi scegliere se eseguire la registrazione solo con l'indirizzo e-mail, solo con il numero di telefono o con uno dei due a scelta.

Per scegliere gli attributi del nome utente, non selezionare Nome utente come opzione di accesso quando crei il pool di utenti.

L'indirizzo e-mail o il numero di telefono devono essere univoci e non essere già in uso per un altro utente. Non è necessario che siano verificati. Dopo che l'utente ha effettuato la registrazione utilizzando un indirizzo e-mail o un numero di telefono, non può creare un nuovo account con lo stesso indirizzo e-mail o numero di telefono. L'utente può solo riutilizzare l'account esistente e reimpostare la password dell'account, se necessario. Tuttavia, può modificare l'indirizzo e-mail o il numero di telefono con un nuovo indirizzo e-mail o numero di telefono. Se l'indirizzo e-mail o il numero di telefono non sono già in uso diventano il nuovo nome utente.

Quando selezioni sia l'indirizzo email che il numero di telefono come attributi del nome utente, gli utenti possono accedere con l'uno o l'altro, anche se forniscono valori per entrambi gli attributi. Il nome utente di accesso si basa sul valore passato nel `Username` parametro di SignUp.

> **Note**
>
> Quando un utente effettua la registrazione utilizzando un indirizzo e-mail come proprio nome utente, può sostituire il nome utente con un altro indirizzo e-mail, ma non con un numero di telefono. Allo stesso modo, se effettua la registrazione utilizzando un numero di telefono come proprio nome utente, può sostituire il nome utente con un altro numero di telefono, ma non con un indirizzo e-mail.

Utilizza i passaggi seguenti durante il processo di creazione del bacino d'utenza per configurare la registrazione e l'accesso con l'indirizzo e-mail o il numero di telefono.

Username attributes (console)

La procedura seguente crea un pool di utenti con gli attributi del nome utente dell'indirizzo e-mail o del numero di telefono. La differenza nel processo per gli attributi del nome utente nella console Amazon Cognito è che non si imposta anche Username come attributo di accesso.

Per creare un pool di utenti con attributi di nome utente nella console Amazon Cognito

1. Passa ad Amazon Cognito nella Console di gestione AWS. Se la console te lo richiede, inserisci le tue credenziali. AWS
2. Crea un nuovo pool di utenti con il pulsante Inizia o Crea pool di utenti.
3. Scegli le impostazioni dell'applicazione in Definisci la tua applicazione.
4. In Configura le opzioni in Opzioni per gli identificatori di accesso, seleziona gli attributi del nome utente: e-mail, numero di telefono o entrambi. Lascia il nome utente deselezionato.
5. Come procedura consigliata, seleziona gli attributi del nome utente come Attributi obbligatori per l'iscrizione. Nel modulo di registrazione con accesso gestito, Amazon Cognito richiede ai nuovi utenti di fornire valori per gli attributi obbligatori. Se non imposti gli attributi del nome utente come richiesto, Amazon Cognito non richiede ai nuovi utenti di fornire valori per loro. In questo scenario, devi configurare l'applicazione per raccogliere e inviare indirizzi e-mail o numeri di telefono per ogni utente prima che possa accedere.
6. In Aggiungi un URL di ritorno, configura un URL di callback dell'applicazione per il reindirizzamento dopo l'accesso gestito.
7. Scegli Crea.

Username attributes (API/SDK)

In una CreateUserPoolrichiesta, configura il `UsernameAttributes` parametro come mostrato. Per consentire l'accesso solo con nomi utente contenenti indirizzi e-mail, specificate `email` alone in questo elenco. Per consentire l'accesso solo con nomi utente con numeri di telefono, specifica alone. `phone_number` Questo parametro sostituisce il nome utente come opzione di accesso.

```
"UsernameAttributes": [
   "email",
   "phone_number"
],
```

L'esempio seguente configura un pool di utenti in cui ogni utente accede utilizzando solo il proprio indirizzo e-mail come nome utente. Come procedura consigliata, imposta anche `email` come attributo di registrazione obbligatorio in modo che i nuovi utenti forniscano il proprio indirizzo e-mail quando si iscrivono tramite accesso gestito:

```
"UsernameAttributes": [
   "email"
],
```

Quando configuri gli attributi del nome utente, puoi effettuare richieste SignUpAPI che inseriscono un indirizzo e-mail o un numero di telefono nel `username` parametro. Di seguito è riportato il comportamento del funzionamento dell'`SignUp`API con gli attributi del nome utente.

- Se la `username` stringa è in un formato di indirizzo e-mail valido, ad esempio`user@example.com`, il pool di utenti compila automaticamente l'`email`attributo dell'utente con il `username` valore.
- Se la `username` stringa è in un formato numerico di telefono valido, ad esempio`+12065551212`, il pool di utenti compila automaticamente l'`phone_number`attributo dell'utente con il `username` valore.
- Se la stringa `username` non è in un formato di indirizzo e-mail o numero di telefono, l'API `SignUp` restituisce un'eccezione.
- Se la stringa `username` contiene un indirizzo e-mail o un numero di telefono già in uso, l'API `SignUp` restituisce un'eccezione.
- L'`SignUp`API compila l'`username`attributo con un UUID per l'utente. Questo UUID ha lo stesso valore dell'attestazione `sub` nel token di identità dell'utente.

Puoi utilizzare un indirizzo e-mail o un numero di telefono al posto del nome utente in tutte le API tranne l'operazione. ListUsers Ciò include operazioni successive all'iscrizione come e. ConfirmSignUpResendConfirmationCode Sebbene `SignUp` imposti l'`username`attributo sull'UUID, puoi comunque passare l'indirizzo email o il numero di telefono dell'utente come `Username` valore nella richiesta. `SignUp` Nelle richieste `ListUsers` API, puoi specificare un `Filter` di `email` o. `phone_number` Per filtrare in base`username`, fornisci il nome utente UUID. Non puoi filtrare in base all'indirizzo email o al numero di telefono in un `username` filtro.

Attributi personalizzati

È possibile aggiungere fino a 50 attributi personalizzati al bacino d'utenza. Puoi specificare una lunghezza minima, una lunghezza massima o entrambe per gli attributi personalizzati. Tuttavia, la lunghezza massima per qualsiasi attributo personalizzato non può superare i 2048 caratteri. Il nome di un attributo personalizzato deve corrispondere al modello di espressione regolare descritto nel `Name` parametro di SchemaAttributeType.

> **Gli attributi personalizzati sono permanenti**
>
> Non puoi rimuovere o modificare un attributo personalizzato dopo averlo aggiunto al tuo pool di utenti. Non puoi eliminare o rinominare l'attributo. Inoltre, non puoi modificarne il tipo di dati, la mutabilità o i vincoli di lunghezza. Pianificate lo schema di attributi personalizzato prima di aggiungere attributi a un pool di utenti di produzione.

Ogni attributo personalizzato presenta le caratteristiche seguenti:

- Puoi definirlo come stringa, numero, booleano o `DateTime` oggetto. Amazon Cognito scrive i valori degli attributi personalizzati nel token ID solo come stringhe.

  > **Note**
  >
  > Nella console Amazon Cognito, puoi aggiungere attributi personalizzati solo dei tipi di dati stringa e numero. Opzioni aggiuntive come i tipi di dati booleani e di `DateTime` attributi sono disponibili solo nella `SchemaAttributes` proprietà CreateUserPoole UpdateUserPoolnelle richieste API.

  > **Note**
  >
  > Indipendentemente dal tipo di dati assegnato, Amazon Cognito archivia e trasmette ogni valore di attributo personalizzato come stringa. Quando imposti un valore in una richiesta API o SDK AdminUpdateUserAttributes, ad esempio, o UpdateUserAttributesSignUp, devi passare il valore come stringa, anche per un attributo`Number`, `Boolean` o. `DateTime` Passa `"42"` piuttosto `42` che e `"true"` anzichétrue. Un numero o booleano nativo viene rifiutato prima che il valore venga memorizzato; a seconda dell'SDK in uso, si presenta come un errore di convalida del tipo sul lato client o come un servizio.

`InvalidParameterException` L'applicazione è responsabile dell'analisi della stringa riportandola al tipo desiderato quando legge l'attributo.

- Non è possibile richiedere che gli utenti forniscano un valore per l'attributo.
- Non è possibile eliminarlo o modificarlo dopo averlo aggiunto al bacino d'utenza.
- La lunghezza in caratteri del nome dell'attributo rientra nel limite accettato da Amazon Cognito. Per ulteriori informazioni, consulta Quote in Amazon Cognito.
- Può essere mutabile o immutabile. Puoi scrivere un valore in un attributo immutabile solo quando crei un utente. Puoi modificare il valore di un attributo mutabile se il client dell'app dispone dell'autorizzazione di scrittura per l'attributo. Per ulteriori informazioni, consulta Autorizzazioni attributo e ambiti.

> **Note**
>
> Nel codice e nelle impostazioni delle regole per Utilizzo del controllo degli accessi basato su ruoli, gli attributi personalizzati richiedono il prefisso `custom:` per distinguerli dagli attributi standard.

È inoltre possibile aggiungere attributi di sviluppatore quando si creano pool di utenti, nella `SchemaAttributes` proprietà di. CreateUserPool Gli attributi di sviluppatore hanno un prefisso `dev:`. È possibile modificare gli attributi di sviluppatore di un utente solo con AWS credenziali. Gli attributi di sviluppatore sono una funzionalità legacy che è stata sostituita da Amazon Cognito con le autorizzazioni di lettura/scrittura del client dell'app.

Utilizza la procedura seguente per creare un attributo delle chiavi personalizzate.

Aggiunta di un attributo personalizzato utilizzando la console

1. Vai ad Amazon Cognito in. Console di gestione AWS Se la console te lo richiede, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il Sign-upmenu e, nella sezione Attributi personalizzati, scegli Aggiungi attributi personalizzati.

5. Nella pagina Add custom attributes (aggiungi attributi personalizzati) fornisci i seguenti dettagli sul nuovo attributo:

   - Immetti un nome.
   - Seleziona un tipo tra String o Number.
   - Immetti un valore numerico o una lunghezza minima della stringa.
   - Immetti un valore numerico o una lunghezza massima della stringa.
   - Seleziona Mutable (Mutabile) se desideri concedere agli utenti l'autorizzazione per modificare il valore di un attributo personalizzato dopo l'impostazione del valore iniziale.
6. Scegli Save changes (Salva modifiche).

Autorizzazioni attributo e ambiti

Per ogni client di app puoi impostare le autorizzazioni di lettura e scrittura di ogni attributo utente. In questo modo, puoi controllare l'accesso concesso a qualsiasi app per leggere e modificare ogni attributo che archivi per gli utenti. Ad esempio, puoi avere un attributo personalizzato che indica se un utente è un cliente pagante o meno. Le tue app possono essere in grado di vedere questo attributo, ma non di modificarlo direttamente. Al contrario, puoi aggiornare questo attributo utilizzando uno strumento di amministrazione o un processo in background. Puoi impostare le autorizzazioni per gli attributi utente con la console Amazon Cognito, l'API Amazon Cognito o AWS CLI. Di default, tutti i nuovi attributi personalizzati saranno disponibili solo dopo aver impostato le relative autorizzazioni di lettura e scrittura. Per impostazione predefinita, quando crei un nuovo client per l'app, concedi all'app le autorizzazioni di lettura e scrittura per tutti gli attributi standard e personalizzati. Per limitare l'app alla sola quantità di informazioni richiesta, assegna autorizzazioni specifiche agli attributi nella configurazione del client dell'app.

Come procedura consigliata, specifica le autorizzazioni di lettura e scrittura degli attributi quando crei un client per l'app. Concedi al client dell'app l'accesso al set minimo di attributi utente necessari per il funzionamento dell'applicazione.

> **Important**
>
> Se non specifichi un client `WriteAttributes` di app, il client dell'app ha accesso in scrittura a tutti gli attributi standard e personalizzati. Gli utenti possono modificare autonomamente il valore di questi attributi con`UpdateUserAttributes`.
> L'applicazione o il pool di identità potrebbero prendere decisioni di autorizzazione in base a un attributo. In tal caso, rimuovi l'attributo da ogni client `WriteAttributes` di app nel tuo

pool di utenti. Gli attributi utente sono condivisi da tutti i client dell'app in un pool di utenti. Un valore che un utente scrive tramite un client di app appare nei token emessi da tutti i client dell'app.

> **Note**
>
> DescribeUserPoolClientrestituisce valori solo per `ReadAttributes` e `WriteAttributes` quando si configurano autorizzazioni del client dell'app diverse da quelle predefinite.

Per aggiornare le autorizzazioni degli attributi (Console di gestione AWS)

1. Vai ad Amazon Cognito in. Console di gestione AWS Se la console te lo richiede, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu App client e scegli un app client dall'elenco.
5. Nella scheda Autorizzazioni degli attributi, scegli Modifica.
6. Nella pagina Edit attribute read and write permissions (impostazione delle autorizzazioni di lettura e scrittura degli attributi) configura le autorizzazioni di lettura e scrittura, quindi seleziona Save changes (salva modifiche).

Ripeti questi passaggi per ogni client di app che usa l'attributo personalizzato. Per rendere un attributo di sola lettura nelle tue app, rimuovi l'autorizzazione di scrittura per quell'attributo da ogni client di app del tuo pool di utenti.

Per ogni client di app, puoi contrassegnare gli attributi come leggibili o scrivibili. Questo vale sia per gli attributi standard sia per quelli personalizzati. La tua app può recuperare il valore degli attributi contrassegnati come leggibili e può impostare o modificare il valore degli attributi contrassegnati come scrivibili. Se la tua app tenta di impostare un valore per un attributo che non è autorizzata a scrivere, Amazon Cognito restituisce. `NotAuthorizedException` GetUserle richieste includono un token di accesso con una dichiarazione del client dell'app; Amazon Cognito restituisce solo valori per gli attributi che il client dell'app è in grado di leggere. Il token ID dell'utente di un'app contiene solo le attestazioni che corrispondono agli attributi leggibili. Tutti i client dell'app possono scrivere gli attributi richiesti dal pool di utenti. Puoi impostare il valore di un attributo in una richiesta API del pool di utenti

di Amazon Cognito solo quando fornisci anche un valore per gli attributi obbligatori che non hanno ancora un valore.

Gli attributi personalizzati dispongono di funzionalità distinte per le autorizzazioni di lettura e scrittura. È possibile creare questi attributi come mutabili o immutabili per il pool di utenti e impostarli come attributi di lettura o scrittura per qualsiasi client dell'app.

Un attributo personalizzato immutabile può essere aggiornato una sola volta, durante la creazione dell'utente. È possibile popolare un attributo immutabile con i seguenti metodi.

- `SignUp`: un utente esegue la registrazione mediante un client dell'app che dispone dell'accesso in scrittura a un attributo personalizzato immutabile. Specifica un valore per tale attributo.
- Sign-in con un IdP di terze parti: un utente accede a un client di app con accesso in scrittura a un attributo personalizzato immutabile. La configurazione del pool di utenti per il relativo gestore dell'identità digitale prevede una regola per mappare una richiesta specificata a un attributo immutabile. Questo è possibile ma non pratico, perché l'utente potrà accedere una sola volta. In caso di tentativi di accesso successivi al primo, Amazon Cognito rifiuta il tentativo a causa della regola di mappatura su un attributo ora non scrivibile.
- `AdminCreateUser`: viene fornito un valore per un attributo immutabile.

Attribuisci autorizzazioni con ambiti

Nei pool di utenti configurati con un AWS SDK o CDK, l'API REST o il AWS CLI, puoi configurare l'accesso in lettura o scrittura del client dell'app con l'ambito OIDC. `oidc:profile` `oidc:profile`Concede l'accesso in lettura o scrittura ai seguenti attributi standard:

- `name`
- `family_name`
- `given_name`
- `middle_name`
- `nickname`
- `preferred_username`
- `profile`
- `picture`
- `website`

- `gender`
- `birthdate`
- `zoneinfo`
- `locale`

Questo elenco contiene gli attributi standard OIDC meno`email`,, e `phone_number` `subaddress`, come definito nella sezione 2.4 delle specifiche OIDC. Per informazioni sugli ambiti che puoi assegnare ai client dell'app, consulta Ambiti, M2M e server di risorse.

Per configurare il client dell'app in modo che scriva negli attributi inclusi nell'`oidc:profile`ambito, imposta il valore di WriteAttributestoo`idc:profile`, più qualsiasi altro attributo che desideri consentire all'applicazione di modificare, in una richiesta o API. CreateUserPoolClientUpdateUserPoolClient Analogamente, per concedere l'accesso in lettura a questi attributi, aggiungete `oidc:profile` il valore di ReadAttributes.

Puoi modificare le autorizzazioni e gli ambiti attributo dopo aver creato il bacino d'utenza.

Aggiornamento della configurazione del pool di utenti e del client dell'app

Quando desideri modificare un'impostazione in un pool di utenti o in un client di app, puoi applicare l'aggiornamento nella console Amazon Cognito con pochi clic. Navighi tra le schede basate sulle funzionalità nelle impostazioni del pool di utenti e aggiorni i campi come descritto in altre aree di questa guida.

Molte organizzazioni gestiscono le proprie risorse in modo programmatico AWS CloudFormation, in applicazioni basate su AWS SDK o CDK e in altri software di automazione. Se questo è il modello di gestione delle risorse, è necessario prestare particolare attenzione quando si organizzano modifiche alle risorse.

L'API opera UpdateUserPoole UpdateUserPoolClientapporta aggiornamenti a un pool di utenti o a un client di app esistente. Ciascuno include un avviso nell'API Reference: se non fornisci un valore per un attributo, Amazon Cognito lo imposta sul valore predefinito. Quando invii una richiesta di aggiornamento con un solo parametro, Amazon Cognito imposta quel parametro sul valore che preferisci e imposta tutti gli altri su un valore predefinito. Questo può ripristinare le configurazioni tra cui lo schema degli attributi, i trigger Lambda e la configurazione dei messaggi di posta elettronica e SMS.

Inoltre, alcune impostazioni sono bloccate dopo la creazione del pool di utenti o del client dell'app e non è possibile modificarle a meno che non si crei una nuova risorsa.

Argomenti

- Impostazioni che non puoi modificare
- Configurazione SMS
- Aggiornamento di un pool di utenti con un AWS SDK, AWS CDK o API REST

Impostazioni che non puoi modificare

Dopo aver creato un pool di utenti, non puoi più modificare alcune impostazioni. Se desideri modificare le seguenti impostazioni, devi creare un nuovo pool di utenti o un nuovo client dell'app.

> **Note**
>
> In precedenza, non era possibile modificare il nome di un pool di utenti. Questo è cambiato. Ora puoi assegnare nuovi nomi descrittivi ai tuoi pool di utenti.

ID del pool di utenti

Nome del parametro API: Id/UserPoolId

L'ID del pool di utenti, ad esempio`us-east-1_EXAMPLE`, viene generato automaticamente da Amazon Cognito e non può essere modificato.

Opzioni di accesso al pool di utenti di Amazon Cognito

Nomi dei parametri API: AliasAttributese UsernameAttributes

Gli attributi che gli utenti possono passare come nome utente quando accedono. Quando crei un pool di utenti, puoi scegliere di consentire l'accesso con nome utente, indirizzo email, numero di telefono o un nome utente preferito. Per modificare le opzioni di accesso del pool di utenti, devi crearne uno nuovo.

Make user name case sensitive (Il nome utente rispetta la distinzione tra maiuscole e minuscole)

Nome del parametro API: UsernameConfiguration

Quando crei un nome utente che corrisponde a un altro nome utente ad eccezione delle lettere maiuscole, Amazon Cognito può trattarlo come lo stesso utente o come utenti unici. Per ulteriori

informazioni, consulta Distinzione tra maiuscole e minuscole del bacino d'utenza. Per modificare la distinzione tra lettere maiuscole e minuscole, devi creare uno nuovo pool di utenti.

Client secret

Nome del parametro API: GenerateSecret

Quando crei un client dell'app, puoi generare un segreto del client in modo che solo le origini attendibili possano inviare richieste al pool di utenti. Per ulteriori informazioni, consulta Application-specific impostazioni con i client dell'app. Per modificare il segreto di un client, crea un nuovo client dell'app nello stesso pool di utenti.

Attributi richiesti

Nome del parametro API: Schema

Attributi per i quali gli utenti devono fornire valori quando si registrano o quando tali attributi vengono creati. Per ulteriori informazioni, consulta Utilizzo degli attributi utente. Per modificare gli attributi richiesti, devi creare un nuovo pool di utenti.

Attributi personalizzati (eliminazione)

Nome del parametro API: Schema

Attributi con nomi personalizzati. È possibile modificare il valore dell'attributo personalizzato di un utente, ma non è possibile eliminare un attributo personalizzato dal pool di utenti. Per ulteriori informazioni, consulta Utilizzo degli attributi utente. Se si raggiunge il numero massimo di attributi personalizzati e si desidera modificare l'elenco, creare un nuovo pool di utenti.

Configurazione SMS

Dopo aver attivato i messaggi SMS nel tuo pool di utenti, non puoi disattivarli.

- Se scegli di configurare i messaggi SMS quando crei un pool di utenti, non puoi disattivare gli SMS dopo aver completato la configurazione.
- Puoi attivare i messaggi SMS in un pool di utenti che hai creato, ma dopo non puoi disattivare gli SMS.
- Amazon Cognito può utilizzare i messaggi SMS per l'invito e il ripristino degli account utente, la verifica degli attributi e l'autenticazione a più fattori (MFA). Dopo aver attivato i messaggi SMS, puoi attivare o disattivare i messaggi SMS per queste funzioni in qualsiasi momento.

- La configurazione dei messaggi SMS include un ruolo IAM che deleghi ad Amazon Cognito per inviare messaggi con Amazon SNS. Puoi modificare il ruolo assegnato in qualsiasi momento.

Aggiornamento di un pool di utenti con un AWS SDK, AWS CDK o API REST

Nella console Amazon Cognito, puoi modificare le impostazioni del pool di utenti un parametro alla volta. Ad esempio, per aggiungere un trigger Lambda, scegli Aggiungi trigger Lambda e scegli la funzione e il tipo di trigger. L'API dei pool di utenti di Amazon Cognito è strutturata in modo tale che le operazioni di aggiornamento per i pool di utenti e i client delle app richiedano il set completo di parametri per il pool di utenti. Tuttavia, la console automatizza in modo trasparente questa operazione di aggiornamento con le altre impostazioni del pool di utenti.

A volte è possibile che una modifica apportata in un altro punto dell'utente Account AWS possa causare la generazione di un errore negli aggiornamenti quando non sono correlati all'impostazione che si desidera modificare. Un'identità Amazon SES eliminata o una modifica di un'autorizzazione IAM per AWS WAF, ad esempio. Se uno dei parametri correnti non è più valido, non puoi aggiornare le impostazioni finché non lo correggi. Quando riscontri un errore di questo tipo, esamina la risposta all'errore e convalida l'impostazione menzionata.

I pool di utenti di Amazon Cognito AWS Cloud Development Kit (AWS CDK), l'API REST e gli AWS SDK sono strumenti per l'automazione e la configurazione programmatica delle risorse di Amazon Cognito. Le richieste con questi strumenti devono inoltre, come la console Amazon Cognito, aggiornare un'impostazione con una configurazione completa delle risorse nel corpo della richiesta. A un livello elevato, è necessario eseguire la seguente procedura.

1. Acquisisci l'output di un'operazione che descrive la configurazione della risorsa esistente.
2. Modifica l'output modificando le impostazioni.
3. Invia la configurazione modificata con un'operazione che aggiorni la tua risorsa.

La procedura seguente aggiorna la configurazione con l'operazione UpdateUserPoolAPI. Lo stesso approccio, con campi di input diversi, si applica a UpdateUserPoolClient.

> **Important**
>
> Se non fornisci valori per i parametri esistenti, Amazon Cognito li imposta sui valori predefiniti. Ad esempio, se è già presente la proprietà `LambdaConfig` e si invia un'operazione

`UpdateUserPool` con una proprietà `LambdaConfig` vuota, elimina l'assegnazione di tutte le funzioni Lambda ai trigger del pool di utenti. Pianifica di conseguenza quando vuoi automatizzare le modifiche alla configurazione del pool di utenti.

1. Acquisisci lo stato esistente del tuo pool di utenti con DescribeUserPool.
2. Formatta l'output di `DescribeUserPool` per abbinare i parametri della richiesta di`UpdateUserPool`. Rimuovi i seguenti campi di livello superiore e i relativi oggetti figlio dal JSON di output.
   - `Arn`
   - `CreationDate`
   - `CustomDomain`
     - Aggiorna questo campo con l'operazione UpdateUserPoolDomainAPI.
   - `Domain`
     - Aggiorna questo campo con l'operazione UpdateUserPoolDomainAPI.
   - `EmailConfigurationFailure`
   - `EstimatedNumberOfUsers`
   - `Id`
   - `LastModifiedDate`
   - `Name`
   - `SchemaAttributes`
   - `SmsConfigurationFailure`
   - `Status`
3. Conferma che il JSON risultante corrisponda ai parametri della richiesta di `UpdateUserPool`.
4. Modifica tutti i parametri che desideri modificare nel JSON risultante.
5. Invia una richiesta API `UpdateUserPool` con il JSON modificato come input della richiesta.

È possibile utilizzare anche questo output `DescribeUserPool` modificato nel parametro `--cli-input-json` di `update-user-pool` nella AWS CLI.

In alternativa, esegui il AWS CLI comando seguente per generare JSON con valori vuoti per i campi di input accettati per. `update-user-pool` È quindi possibile popolare questi campi con i valori esistenti del pool di utenti.

```
aws cognito-idp update-user-pool --generate-cli-skeleton --output json
```

Esegui il comando riportato di seguito per generare lo stesso oggetto JSON per un client dell'app.

```
aws cognito-idp update-user-pool-client --generate-cli-skeleton --output json
```

Personalizzazione dei pool di utenti

I pool di utenti di Cognito espongono i punti di estensione in modo che tu possa modellare l'esperienza di registrazione e accesso alla tua applicazione. Usa i trigger Lambda per eseguire il tuo codice nei momenti chiave del ciclo di vita dell'utente e configurare i messaggi e le analisi inviati dal tuo pool di utenti.

Argomenti

- Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda
- Messaggistica e analisi

Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda

Amazon Cognito utilizza AWS Lambda funzioni per modificare il comportamento di autenticazione del tuo pool di utenti. Puoi configurare il tuo pool di utenti per richiamare automaticamente le funzioni Lambda prima della prima registrazione, dopo aver completato l'autenticazione e in diverse fasi intermedie. Le tue funzioni possono modificare il comportamento predefinito del flusso di autenticazione, effettuare richieste API per modificare il pool di utenti o altre AWS risorse e comunicare con sistemi esterni. Il codice delle funzioni Lambda è personale. Amazon Cognito invia i dati degli eventi alla tua funzione, attende che la funzione elabori i dati e nella maggior parte dei casi anticipa un evento di risposta che riflette le modifiche che desideri apportare alla sessione.

All'interno del sistema di eventi di richiesta e risposta, puoi introdurre problemi di autenticazione personalizzati, migrare gli utenti tra il tuo pool di utenti e un altro archivio di identità, personalizzare i messaggi e modificare i token web JSON (JWT).

I trigger Lambda possono personalizzare la risposta che Amazon Cognito restituisce all'utente dopo aver avviato un'operazione nel pool di utenti. Ad esempio, puoi impedire l'accesso da parte di un utente che altrimenti verrebbe consentito. Possono anche eseguire operazionidi runtime nell'ambiente AWS , nelle API esterne, nei database o negli archivi di identità. Il trigger di migrazione utente, ad esempio, può combinare un'operazione esterna con una modifica in Amazon Cognito, cercando le informazioni sull'utente in una directory esterna e quindi impostando gli attributi del nuovo utente in base alle informazioni esterne.

Quando un trigger Lambda è assegnato al pool di utenti, Amazon Cognito interrompe il flusso predefinito per richiedere informazioni alla funzione. Amazon Cognito genera pertanto un evento JSON e lo passa alla funzione. L'evento contiene informazioni sulla richiesta dell'utente di creare un account utente, eseguire l'accesso, reimpostare una password o aggiornare un attributo. La funzione ha quindi l'opportunità di eseguire l'operazione o di restituire l'evento senza modifiche. Un evento restituito non modificato notifica al pool di utenti di procedere con l'azione predefinita per l'evento. Ad esempio, il trigger di pre-registrazione può confermare automaticamente gli utenti per la fonte del `PreSignUp_SignUp` trigger, ma restituire l'evento invariato per gli utenti esterni e creati dall'amministratore.

La tabella riportata di seguito riepiloga alcuni dei modi in cui è possibile utilizzare i trigger Lambda per personalizzare le operazioni del bacino d'utenza:

Flusso di bacini d'utenza	Operation	Description
Flusso di autenticazione personalizzato	Definizione di una richiesta di autenticazione	Determina la prossima richiesta in un flusso di autorizzazione personalizzato
	Creazione di una richiesta di autenticazione	Crea una richiesta in un flusso di autorizzazione personalizzato
	Verifica della risposta a una richiesta di autenticazione	Determina se una risposta è corretta in un flusso di autorizzazione personalizzato
Eventi di autenticazione	the section called “Autenticazione preliminare”	Convalida personalizzata per accettare o rifiutare la richiesta di accesso
	the section called “Autenticazione successiva”	Esegue il log di eventi per analisi personalizzate
	the section called “Generazione pre-token”	Aumenta o sopprime le richieste di token
Federazione	the section called “Federazione in entrata”	Trasforma gli attributi utente federati prima della creazione

Flusso di bacini d'utenza	Operation	Description
		o dell'aggiornamento degli utenti nei pool di utenti di Amazon Cognito
Sign-Up	the section called “Pre-iscrizione”	Esegue la convalida personalizzata accettando o negando la richiesta di registrazione
	the section called “Conferma successiva”	Aggiunge messaggi di benvenuto personalizzati o il log di eventi per analisi personalizzate
	the section called “Migrazione utente”	Migra un utente da una directory di utenti esistente ai bacini d'utenza
Messaggi	the section called “Messaggio personalizzato”	Esegue la personalizzazione avanzata e localizzazione di messaggi
Creazione di token	the section called “Generazione pre-token”	Aggiunge o rimuove attributi negli ID e nei token di accesso
Provider di terze parti di messaggi e-mail ed SMS	the section called “Mittenti personalizzati”	Utilizza un provider di terze parti per inviare SMS e messaggi e-mail

Argomenti

- Cose da sapere sui trigger Lambda
- Aggiungere un trigger Lambda per un pool di utenti
- Evento trigger Lambda per il bacino d'utenza
- Parametri comuni del trigger Lambda del bacino d'utenza
- Metadati del client
- Connessione delle operazioni API ai trigger Lambda

- Connessione dei trigger Lambda alle operazioni funzionali del pool di utenti
- Trigger Lambda di pre-registrazione
- Trigger Lambda di post-conferma
- Trigger Lambda di pre-autenticazione
- Trigger Lambda di post-autenticazione
- Trigger Lambda di federazione in entrata
- Trigger Lambda di richieste di autenticazione personalizzate
- Trigger Lambda di pre-generazione del token
- Trigger Lambda di migrazione utenti
- Trigger Lambda di messaggi personalizzati
- Trigger Lambda del mittente personalizzato

Cose da sapere sui trigger Lambda

Quando prepari i tuoi pool di utenti per le funzioni Lambda, considera quanto segue:

- Gli eventi inviati da Amazon Cognito ai trigger Lambda potrebbero cambiare con nuove funzionalità. Le posizioni degli elementi di risposta e richiesta nella gerarchia JSON potrebbero cambiare o potrebbero essere aggiunti nomi di elementi. Nella funzione Lambda, puoi ricevere le coppie chiave-valore dell'elemento di input descritte in questa guida, ma una convalida dell'input più rigorosa può causare il mancato completamento delle funzioni.
- Puoi scegliere una di più versioni degli eventi inviati da Amazon Cognito ad alcuni trigger. Alcune versioni potrebbero richiedere di accettare una modifica ai prezzi di Amazon Cognito. Per ulteriori informazioni sui prezzi, consulta Prezzi di Amazon Cognito. Per personalizzare i token di accesso in aTrigger Lambda di pre-generazione del token, devi configurare il tuo pool di utenti con un piano di funzionalità diverso da Lite e aggiornare la configurazione del trigger Lambda per utilizzare la versione 2 dell'evento.
- Ad eccezione di Trigger Lambda del mittente personalizzato, Amazon Cognito richiama le funzioni Lambda in modo sincrono. Quando Amazon Cognito chiama la tua funzione Lambda, questa deve rispondere entro 5 secondi. In caso contrario e se la chiamata può essere ritentata, Amazon Cognito può riprovare la chiamata. Se tutti i tentativi falliscono, la funzione scade. Non puoi modificare questo valore di timeout di cinque secondi. Per ulteriori informazioni, consulta il modello di programmazione Lambda nella AWS Lambda Developer Guide.

Amazon Cognito non riprova le chiamate di funzione che restituiscono un errore Invoke con un codice di stato HTTP 500-599. Questi codici indicano un problema di configurazione che impedisce a Lambda di avviare la funzione. Per ulteriori informazioni, consulta Gestione degli errori e tentativi automatici in. AWS Lambda

- Non puoi dichiarare una versione della funzione nella configurazione del trigger Lambda. I pool di utenti di Amazon Cognito richiamano la versione più recente della tua funzione per impostazione predefinita. Tuttavia, puoi associare una versione della funzione a un alias e impostare il trigger `LambdaArn` sull'alias ARN in una CreateUserPoolrichiesta o API. UpdateUserPool Questa opzione non è disponibile in Console di gestione AWS. Per ulteriori informazioni sugli alias, consulta Alias delle funzioni Lambda nella Guida per gli sviluppatori di AWS Lambda .
- Se si elimina un trigger , è necessario aggiornare il corrispondente trigger nel bacino d'utenza. Ad esempio, se si elimina il trigger di post autenticazione, è necessario impostare il trigger Post authentication (Post autenticazione) nel bacino d'utenza corrispondente su none (nessuno).
- Se la funzione Lambda non restituisce i parametri di richiesta e risposta ad Amazon Cognito o restituisce un errore, l'evento di autenticazione non va a buon fine. Puoi restituire un errore nella funzione per impedire a un utente di eseguire la registrazione, l'autenticazione, la generazione di token o qualsiasi altra fase del flusso di autenticazione di un utente che richiami il trigger Lambda.

  L'accesso gestito restituisce gli errori generati dai trigger Lambda come testo di errore sopra la richiesta di accesso. L'API dei pool di utenti di Amazon Cognito restituisce errori dei trigger nel formato *`[trigger]`* `failed with error` *`[error text from response]`*. Come best practice, genera nelle funzioni Lambda solo gli errori che desideri mostrare agli utenti. Utilizza metodi di output come `print()` registrare qualsiasi informazione sensibile o di debug in Logs. CloudWatch Per un esempio, consulta Esempio di pre-registrazione: Nega l'iscrizione se il nome utente ha meno di cinque caratteri.
- Puoi aggiungere una funzione Lambda in un'altra Account AWS come trigger per il tuo pool di utenti. È necessario aggiungere trigger tra account con le operazioni CreateUserPoole UpdateUserPoolAPI o i relativi equivalenti in e. CloudFormation AWS CLI Non è possibile aggiungere funzioni tra account in. Console di gestione AWS
- Quando aggiungi un trigger Lambda nella console di Amazon Cognito, Amazon Cognito aggiunge alla funzione una policy basata sulle risorse che consente al pool di utenti di invocare la funzione. Quando crei un trigger Lambda al di fuori della console di Amazon Cognito, inclusa una funzione tra account, devi aggiungere le autorizzazioni alla policy basata sulle risorse della funzione Lambda. Le autorizzazioni aggiunte devono consentire ad Amazon Cognito di invocare la funzione

per conto del pool di utenti. Puoi aggiungere autorizzazioni dalla console Lambda o utilizzare l'operazione API Lambda AddPermission.

Esempio di politica Lambda Resource-Based

La seguente policy basata su risorse Lambda garantisce ad Amazon Cognito la capacità limitata di richiamare una funzione Lambda. Amazon Cognito può richiamare questa funzione solo quando lo fa per conto sia del bacino d'utenza nella condizione `aws:SourceArn` sia dell'account nella condizione `aws:SourceAccount`.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "default",
    "Statement": [
        {
            "Sid": "LambdaCognitoIdpTrust",
            "Effect": "Allow",
            "Principal": {
                "Service": "cognito-idp.amazonaws.com"
            },
            "Action": "lambda:InvokeFunction",
            "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:MyFunction",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "AWS:SourceArn": "arn:aws:cognito-idp:us-
east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        }
    ]
}
```

Aggiungere un trigger Lambda per un pool di utenti

Come aggiungere un trigger Lambda al bacino d'utenza con la console

1. Usa la console Lambda per creare una funzione Lambda. Per ulteriori informazioni sulle funzioni Lambda, consulta la Guida per gli sviluppatori di AWS Lambda.
2. Passa alla console Amazon Cognito e scegli User Pools (Bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Estensioni e individua i trigger Lambda.
5. Scegli Add a Lambda trigger (Aggiungi un trigger Lambda).
6. Seleziona una categoria di trigger Lambda in base alla fase di autenticazione che desideri personalizzare.
7. Seleziona Assegna funzione Lambda e seleziona una funzione nello Regione AWS stesso gruppo di utenti.

   > **Note**
   >
   > Se le tue credenziali AWS Identity and Access Management (IAM) sono autorizzate ad aggiornare la funzione Lambda, Amazon Cognito aggiunge una policy Lambda basata sulle risorse. Con questa policy, Amazon Cognito può richiamare la funzione selezionata. Se le credenziali di accesso non dispongono di autorizzazioni IAM sufficienti, è necessario aggiornare separatamente la policy basata sulle risorse. Per ulteriori informazioni, consulta the section called “Da sapere”.

8. Scegli Salva modifiche.
9. Puoi usarla CloudWatch nella console Lambda per registrare la tua funzione Lambda. Per ulteriori informazioni, consulta Accesso ai CloudWatch registri per Lambda.

Evento trigger Lambda per il bacino d'utenza

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione Lambda restituisce ad Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Se la funzione restituisce l'evento di input senza modifiche, Amazon Cognito procede con il comportamento predefinito. Di seguito vengono mostrati i parametri comuni a tutti gli eventi di input del trigger Lambda. Per la sintassi degli eventi specifici del trigger, consulta lo schema degli eventi nella sezione di questa guida per ogni trigger.

JSON

```
{
    "version": "string",
    "triggerSource": "string",
    "region": AWSRegion,
    "userPoolId": "string",
    "userName": "string",
    "callerContext":
        {
            "awsSdkVersion": "string",
            "clientId": "string"
        },
    "request":
        {
            "userAttributes": {
                "string": "string",
                ....
            }
        },
    "response": {}
}
```

Parametri comuni del trigger Lambda del bacino d'utenza

version

Il numero di versione della tua funzione Lambda.

triggerSource

Il nome dell'evento che ha attivato la funzione Lambda. Per una descrizione di ogni triggerSource, consulta Connessione dei trigger Lambda alle operazioni funzionali del pool di utenti.

region

Il Regione AWS come esempio. `AWSRegion`

utente PoolId

L'ID del pool di utenti.

userName

Il nome dell'utente corrente.

callerContext

I metadati sulla richiesta e sull'ambiente del codice. Contiene i campi aws SdkVersion e clientId.

aws SdkVersion

La versione dell' AWS SDK che ha generato la richiesta.

clientId

L'ID del client dell'app del pool di utenti.

richiesta

I dettagli della richiesta API dell'utente. Include i seguenti campi e i parametri di richiesta specifici del trigger, Ad esempio, un evento che Amazon Cognito invia a un trigger di preautenticazione contiene anche un parametro. `userNotFound` Puoi elaborare il valore di questo parametro per eseguire un'operazione personalizzata quando l'utente tenta di accedere con un nome utente non registrato.

userAttributes

Una o più coppie chiave-valore di nomi e valori di attributi dell'utente, ad esempio `"email": "john@example.com"`.

risposta

Questo parametro non contiene alcuna informazione nella richiesta originale. La funzione Lambda deve restituire l'intero evento ad Amazon Cognito e aggiungere eventuali parametri di riposta a `response`. Per visualizzare quali parametri di riposta può includere la funzione, fai riferimento alla documentazione relativa al trigger che vuoi usare.

Metadati del client

Puoi inviare parametri personalizzati alle funzioni di attivazione Lambda nelle operazioni e Endpoint Token nelle richieste API. Con i metadati del client, l'applicazione può raccogliere informazioni aggiuntive sull'ambiente in cui hanno origine le richieste. Quando passi i metadati del client alle tue funzioni Lambda, queste possono elaborare i dati aggiuntivi e utilizzarli per la registrazione o la personalizzazione dei flussi di autenticazione. I metadati del client sono coppie di stringhe di tua scelta e progettate in un formato chiave-valore JSON.

Esempi di utilizzo dei metadati dei client

- Passa i dati di geolocalizzazione al momento dell'iscrizione al trigger di pre-registrazione e impedisci l'accesso da posizioni indesiderate.
- Trasmetti i dati identificativi degli inquilini ai trigger di sfida personalizzati e invia sfide diverse ai clienti di diverse unità aziendali.
- Passa il token di un utente al trigger precedente alla generazione del token e genera un registro del principale per il quale è stata effettuata una richiesta M2M. Per una richiesta di esempio, consulta Credenziali del client con autorizzazione di base.

Ecco un esempio di trasferimento dei metadati del client al trigger di pre-registrazione.

SignUp request

Di seguito è riportato un esempio di SignUprichiesta con metadati del cliente che Amazon Cognito passa a un trigger di pre-registrazione.

```
POST HTTP/1.1
Host: cognito-idp.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: gzip, deflate, br
X-Amz-Target: AWSCognitoIdentityProviderService.SignUp
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
    "ClientId": "1example23456789",
    "Username": "mary_major",
    "Password": "<Password>",
    "SecretHash": "<Secret hash>",
    "ClientMetadata": {
        "IpAddress" : "192.0.2.252",
        "GeoLocation" : "Netherlands (Kingdom of the) [NL]"
    }
    "UserAttributes": [
        {
            "Name": "name",
            "Value": "Mary"
        },
```

```
        {
            "Name": "email",
            "Value": "mary_major@example.com"
        },
        {
            "Name": "phone_number",
            "Value": "+12065551212"
        }
    ],
}
```

Lambda trigger input event

La richiesta genera il seguente corpo di richiesta alla funzione di pre-registrazione.

```
{
    "callerContext": {
        "awsSdkVersion": "aws-sdk-unknown-unknown",
        "clientId": "1example23456789"
    },
    "region": "us-west-2",
    "request": {
        "clientMetadata": {
            "GeoLocation": "Netherlands (Kingdom of the) [NL]",
            "IpAddress": "192.0.2.252"
        },
        "userAttributes": {
            "email": "mary_major@example.com",
            "name": "Mary",
            "phone_number": "+12065551212"
        },
        "validationData": null
    },
    "response": {
        "autoConfirmUser": false,
        "autoVerifyEmail": false,
        "autoVerifyPhone": false
    },
    "triggerSource": "PreSignUp_SignUp",
    "userName": "mary_major2",
    "userPoolId": "us-west-2_EXAMPLE",
    "version": "1"
}
```

Metadati client per le credenziali del client machine-to-machine (M2M)

È possibile trasmettere i metadati dei client nelle richieste M2M. I metadati del client sono informazioni aggiuntive provenienti da un ambiente utente o applicativo che possono contribuire ai risultati di un. Trigger Lambda di pre-generazione del token Nelle operazioni di autenticazione con un user principal, è possibile passare i metadati del client al trigger precedente alla generazione del token nel corpo delle richieste API AdminRespondToAuthChallenge. RespondToAuthChallenge Poiché le applicazioni gestiscono il flusso per la generazione di token di accesso per M2M con richieste dirette aEndpoint Token, hanno un modello diverso. Nel corpo POST delle richieste di token per le credenziali del client, passate un `aws_client_metadata` parametro con l'oggetto di metadati del client URL-encoded () `x-www-form-urlencoded` a string. Per una richiesta di esempio, consulta Credenziali del client con autorizzazione di base. Di seguito è riportato un parametro di esempio che passa le coppie chiave-valore. `{"environment": "dev", "language": "en-US"}`

```
aws_client_metadata=%7B%22environment%22%3A%20%22dev%22,%20%22language%22%3A%20%22en-US%22%7D
```

Attributi utente temporanei: **`validationData`**

Alcune operazioni di autenticazione hanno anche un `validationData` parametro. Come i metadati dei client, questa è un'opportunità per passare informazioni esterne che Amazon Cognito non raccoglie automaticamente ai trigger Lambda. Il campo dei dati di convalida ha lo scopo di fornire alla funzione Lambda un contesto utente aggiuntivo nelle operazioni di registrazione e accesso. SignUpe AdminCreateUserpassa al trigger di `validationData` pre-registrazione. InitiateAuthe AdminInitiateAuthpassa `ClientMetadata` il corpo della richiesta API come `validationData` nell'evento di input ai trigger di pre-autenticazione e migrazione dell'utente.

Per mappare le operazioni API alle funzioni a cui possono trasmettere i metadati del client, consulta le sezioni sull'origine dei trigger che seguono.

Connessione delle operazioni API ai trigger Lambda

Le sezioni seguenti descrivono i trigger Lambda che Amazon Cognito richiama dall'attività nel pool di utenti.

Quando la tua app accede agli utenti tramite l'API dei pool di utenti di Amazon Cognito, l'accesso gestito o gli endpoint del pool di utenti, Amazon Cognito richiama le tue funzioni Lambda in base al contesto della sessione. Per ulteriori informazioni sull'API dei pool di utenti di Amazon Cognito e gli

endpoint del pool di utenti, consultare Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite. Le tabelle nelle seguenti sezioni descrivono gli eventi che inducono Amazon Cognito a richiamare una funzione e la stringa `triggerSource` che Amazon Cognito include nella richiesta.

Argomenti

- Trigger Lambda nell'API Amazon Cognito
- Trigger Lambda per gli utenti locali di Amazon Cognito nell'accesso gestito
- Trigger Lambda per utenti federati

Trigger Lambda nell'API Amazon Cognito

La tabella seguente descrive le stringhe di origine per i trigger Lambda che possono essere richiamati da Amazon Cognito quando l'app crea, aggiorna o fornisce l'accesso a un utente locale.

Origini dei trigger di utenti locali nell'API Amazon Cognito

Operazione API	Trigger Lambda	Origine del trigger
AdminCreateUser	Pre-iscrizione	`PreSignUp_AdminCreateUser`
	Prima della generazione di token	`TokenGeneration_NewPasswordChallenge`
	Messaggio personalizzato	`CustomMessage_AdminCreateUser`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_AdminCreateUser`
	Mittente di SMS personalizzato	`CustomSMSSender_AdminCreateUser`
SignUp	Pre-iscrizione	`PreSignUp_SignUp`
	Messaggio personalizzato	`CustomMessage_SignUp`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_SignUp`

Operazione API	Trigger Lambda	Origine del trigger
	Mittente di SMS personalizzato	`CustomSMSSender_SignUp`
ConfirmSignUp AdminConfirmSignUp	Conferma postale	`PostConfirmation_ConfirmSignUp`
InitiateAuth AdminInitiateAuth	Pre autenticazione	`PreAuthentication_Authentication`
	Autenticazione successiva	`PostAuthentication_Authentication`
	Definisci la sfida di autenticazione	`DefineAuthChallenge_Authentication`
	Crea una sfida di autenticazione	`CreateAuthChallenge_Authentication`
	Verifica della richiesta di autenticazione	`VerifyAuthChallenge_Authentication`

	Prima della generazione di token	`TokenGeneration_Authentication`
		
		`TokenGeneration_AuthenticateDevice`
		
		`TokenGeneration_RefreshTokens`

	Migrazione utente	`UserMigration_Authentication`
	Messaggio personalizzato	`CustomMessage_Authentication`

Operazione API	Trigger Lambda	Origine del trigger
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_AccountTakeOverNotification` `CustomEmailSender_Authentication`
	Mittente di SMS personalizzato	`CustomSMSSender_Authentication`
RespondToAuthChallenge AdminRespondToAuthChallenge	Autenticazione post	`PostAuthentication_Authentication`
	Definisci la sfida di autenticazione	`DefineAuthChallenge_Authentication`
	Crea una sfida di autenticazione	`CreateAuthChallenge_Authentication`
	Verifica della richiesta di autenticazione	`VerifyAuthChallenge_Authentication`

	Prima della generazione di token	`TokenGeneration_Authentication`
		`TokenGeneration_AuthenticateDevice`
		`TokenGeneration_RefreshTokens`

	Messaggio personalizzato	`CustomMessage_Authentication`

Operazione API	Trigger Lambda	Origine del trigger
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_AccountTakeOverNotification` `CustomEmailSender_Authentication`
	Mittente di SMS personalizzato	`CustomSMSSender_Authentication`
ForgotPassword	Migrazione utente	`UserMigration_ForgotPassword`
	Messaggio personalizzato	`CustomMessage_ForgotPassword`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_ForgotPassword`
	Mittente di SMS personalizzato	`CustomSMSSender_ForgotPassword`
ConfirmForgotPassword	Conferma postale	`PostConfirmation_ConfirmForgotPassword`
UpdateUserAttributes AdminUpdateUserAttributes	Messaggio personalizzato	`CustomMessage_UpdateUserAttribute`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_UpdateUserAttribute`
	Mittente di SMS personalizzato	`CustomSMSSender_UpdateUserAttribute`
VerifyUserAttributes	Messaggio personalizzato	`CustomMessage_VerifyUserAttribute`

Operazione API	Trigger Lambda	Origine del trigger
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_VerifyUserAttribute`
	Mittente di SMS personalizzato	`CustomSMSSender_VerifyUserAttribute`
GetTokensFromRefreshToken	Prima della generazione di token	`TokenGeneration_Authentication`

Trigger Lambda per gli utenti locali di Amazon Cognito nell'accesso gestito

La tabella seguente descrive le stringhe di origine per i trigger Lambda che Amazon Cognito può richiamare quando un utente locale accede al tuo pool di utenti con accesso gestito.

Fonti di attivazione utente locali nell'accesso gestito

URI di accesso gestito	Trigger Lambda	Origine del trigger
`/signup`	Pre-iscrizione	`PreSignUp_SignUp`
	Messaggio personalizzato	`CustomMessage_SignUp`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_SignUp`
	Mittente di SMS personalizzato	`CustomSMSSender_SignUp`
`/confirmuser`	Conferma postale	`PostConfirmation_ConfirmSignUp`
`/login`	Pre autenticazione	`PreAuthentication_Authentication`
	Prima della generazione di token	`TokenGeneration_Authentication`

URI di accesso gestito	Trigger Lambda	Origine del trigger

		`TokenGeneration_AuthenticateDevice`
		
		`TokenGeneration_RefreshTokens`

	Migrazione utente	`UserMigration_Authentication`
	Messaggio personalizzato	`CustomMessage_Authentication`

	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_AccountTakeOverNotification`
		
		`CustomEmailSender_Authentication`

	Mittente di SMS personalizzato	`CustomSMSSender_Authentication`
`/forgotpassword`	Migrazione utente	`UserMigration_ForgotPassword`
	Messaggio personalizzato	`CustomMessage_ForgotPassword`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_ForgotPassword`
	Mittente di SMS personalizzato	`CustomSMSSender_ForgotPassword`
`/confirmforgotpassword`	Conferma postale	`PostConfirmation_ConfirmForgotPassword`

Trigger Lambda per utenti federati

È possibile utilizzare i seguenti trigger Lambda per personalizzare i flussi di lavoro del bacino d'utenza per gli utenti che accedono con un provider federato.

> **Note**
>
> Gli utenti federati possono utilizzare l'accesso gestito per accedere oppure puoi generare una richiesta a loro Endpoint Authorize che li reindirizza silenziosamente alla pagina di accesso del loro provider di identità. Non è possibile fornire l'accesso agli utenti federati con l'API del pool di utenti Amazon Cognito.

Origini dei trigger utente federati

Sign-in evento	Trigger Lambda	Origine del trigger
Primo accesso	Pre-iscrizione	`PreSignUp_ExternalProvider`
	Dopo la conferma	`PostConfirmation_ConfirmSignUp`
	Prima della generazione di token	`TokenGeneration_HostedAuth`
Accessi successivi	Pre-autenticazione	`PreAuthentication_Authentication`
	Autenticazione successiva	`PostAuthentication_Authentication`
	Prima della generazione di token	`TokenGeneration_HostedAuth`

L'accesso federato non richiama Trigger Lambda di richieste di autenticazione personalizzate, Trigger Lambda di migrazione utenti, Trigger Lambda di messaggi personalizzati o Trigger Lambda del mittente personalizzato nel bacino d'utenza.

Connessione dei trigger Lambda alle operazioni funzionali del pool di utenti

Ogni trigger Lambda svolge un ruolo funzionale nel pool di utenti, ad esempio un trigger può modificare il flusso di registrazione o aggiungere una richiesta di autenticazione personalizzata. L'evento inviato da Amazon Cognito a una funzione Lambda può riflettere una delle tante operazioni che costituiscono il ruolo funzionale, ad esempio Amazon Cognito richiama un trigger di pre-registrazione quando un utente si registra e quando viene creato. Ognuno di questi casi diversi per lo stesso ruolo funzionale ha un proprio valore `triggerSource`. La funzione Lambda può elaborare gli eventi in entrata in modo diverso in base all'operazione che l'ha richiamata.

Amazon Cognito richiama tutte le funzioni assegnate anche quando un evento corrisponde a un'origine del trigger, ad esempio quando un utente accede a un pool di utenti a cui hai assegnato i trigger di migrazione utenti e di pre-autenticazione, vengono attivati entrambi i trigger.

Sign-up, trigger di conferma e accesso (autenticazione)

Trigger	Valore triggerSource	Event
Prima della registrazione	`PreSignUp_SignUp`	Pre registrazione.
Pre-iscrizione	`PreSignUp_AdminCreateUser`	Pre registrazione quando un amministratore crea un nuovo utente.
Pre-iscrizione	`PreSignUp_ExternalProvider`	Pre-registrazione per provider di identità esterni.
Dopo la conferma	`PostConfirmation_ConfirmSignUp`	Conferma post registrazione.
Conferma successiva	`PostConfirmation_ConfirmForgotPassword`	Conferma successiva alla password dimenticata.
Pre autenticazione	`PreAuthentication_Authentication`	Preautenticazione.
Autenticazione successiva	`PostAuthentication_Authentication`	Post autenticazione.

Trigger di richieste di autenticazione personalizzate

Trigger	Valore triggerSource	Event
Definisci la sfida di autentica zione	`DefineAuthChalleng e_Authentication`	Definizione di una richiesta di autenticazione.
Crea una sfida di autentica zione	`CreateAuthChalleng e_Authentication`	Creazione di una richiesta di autenticazione.
Verifica della richiesta di autenticazione	`VerifyAuthChalleng eResponse_Authenti cation`	Verifica di risposta di una richiesta di autenticazione.

Trigger della federazione

Trigger	Valore triggerSource	Event
Federazione in entrata	`InboundFederation_ ExternalProvider`	Federazione in entrata.

Trigger della pre-generazione di token

Trigger	Valore triggerSource	Event
Generazione precedente al token	`TokenGeneration_Ho stedAuth`	Amazon Cognito autentica l'utente dalla tua pagina di accesso gestita.
Prima della generazione di token	`TokenGeneration_Au thentication`	Autenticazione utente o aggiornamento del token completati.
Generazione precedente al token	`TokenGeneration_Ne wPasswordChallenge`	L'amministratore crea l'utente. Amazon Cognito richiama questa operazione quando l'utente deve modificare una password temporanea.

Trigger	Valore triggerSource	Event
Prima della generazione di token	`TokenGeneration_AuthenticateDevice`	Passaggio finale dell'autenticazione di un dispositivo dell'utente.
Prima della generazione di token	`TokenGeneration_RefreshTokens`	L'utente cerca di aggiornare l'identità e i token di accesso.

Trigger di migrazione utenti

Trigger	Valore triggerSource	Event
Migrazione degli utenti	`UserMigration_Authentication`	Migrazione degli utenti al momento dell'accesso.
Migrazione degli utenti	`UserMigration_ForgotPassword`	Migrazione degli utenti durante il flusso di password dimenticata.

Trigger di messaggi personalizzati

Trigger	Valore triggerSource	Event
Messaggio personalizzato	`CustomMessage_SignUp`	Messaggio personalizzato quando un utente effettua la registrazione nel bacino d'utenza.
Messaggio personalizzato	`CustomMessage_AdminCreateUser`	Messaggio personalizzato quando crei un utente come amministratore e Amazon Cognito gli invia una password temporanea.
Messaggio personalizzato	`CustomMessage_ResendCode`	Messaggio personalizzato quando un utente esistente

Trigger	Valore triggerSource	Event
		richiede un nuovo codice di conferma.
Messaggio personalizzato	`CustomMessage_ForgotPassword`	Messaggio personalizzato quando l'utente richiede la reimpostazione della password.
Messaggio personalizzato	`CustomMessage_UpdateUserAttribute`	Messaggio personalizzato quando un utente modifica il proprio indirizzo e-mail o numero di telefono e Amazon Cognito gli invia un codice di verifica.
Messaggio personalizzato	`CustomMessage_VerifyUserAttribute`	Messaggio personalizzato quando un utente aggiunge un indirizzo e-mail o un numero di telefono e Amazon Cognito gli invia un codice di verifica.
Messaggio personalizzato	`CustomMessage_Authentication`	Messaggio personalizzato quando un utente che ha configurato SMS MFA esegue l'accesso.

Trigger personalizzati per i mittenti

Trigger	Valore triggerSource	Event

Mittente personalizzato	`CustomEmailSender_SignUp`
	
	`CustomSmsSender_SignUp`
		Quando un utente si iscrive al tuo pool di utenti.

Trigger	Valore triggerSource	Event

Mittente personalizzato	`CustomEmailSender_AdminCreateUser`
	
	`CustomSmsSender_AdminCreateUser`
		Quando crei un utente come amministratore e Amazon Cognito gli invia una password temporanea.

Mittente personalizzato	`CustomEmailSender_ForgotPassword`
	
	`CustomSmsSender_ForgotPassword`
		Quando l'utente richiede la reimpostazione della password.

Mittente personalizzato	`CustomEmailSender_UpdateUserAttribute`
	
	`CustomSmsSender_UpdateUserAttribute`
		Quando un utente modifica l'indirizzo e-mail o il numero di telefono e Amazon Cognito invia un codice di verifica.

Mittente personalizzato	`CustomEmailSender_VerifyUserAttribute`
	
	`CustomSmsSender_VerifyUserAttribute`
		Quando un utente aggiunge un indirizzo e-mail o un numero di telefono e Amazon Cognito invia un codice di verifica.

Mittente personalizzato	`CustomEmailSender_Authentication`
	
	`CustomSmsSender_Authentication`
		Quando un utente che ha configurato SMS o e-mail MFA o OTP accede.

Mittente personalizzato	`CustomEmailSender_AccountTakeOverNotification`	Quando le impostazioni di protezione dalle minacce intraprendono un'azione automatica contro il tentativo di accesso di un utente e l'azione relativa al livello di rischio include una notifica.

Trigger Lambda di pre-registrazione

Potresti voler personalizzare la procedura di registrazione nei pool di utenti che dispongono di opzioni di iscrizione self-service. Alcuni usi comuni del trigger di pre-iscrizione consistono nell'eseguire analisi e registrazioni personalizzate di nuovi utenti, applicare standard di sicurezza e governance o collegare gli utenti di un IdP di terze parti a un profilo utente consolidato. Potresti anche avere utenti fidati che non sono tenuti a sottoporsi a verifiche e conferme.

Immediatamente prima che Amazon Cognito completi la creazione di un nuovo utente locale o federato, attiva la funzione Lambda di pre-registrazione. L'`userAttributes`oggetto di richiesta inviato a questa funzione contiene attributi che sono stati forniti durante la registrazione dell'utente locale o che sono stati mappati correttamente dagli attributi del provider per un utente federato. Il tuo pool di utenti richiama questo trigger all'iscrizione in modalità self-service SignUpo al primo accesso con un provider di identità affidabile e alla creazione di utenti con. AdminCreateUser Come parte del processo di registrazione, puoi utilizzare questa funzione per analizzare l'evento di accesso con una logica personalizzata e modificare o rifiutare il nuovo utente.

Argomenti

- Parametri del trigger Lambda di pre-registrazione
- Esempio di pre-registrazione: Auto-confirm utenti di un dominio registrato
- Esempio di pre-iscrizione: Auto-confirm e verifica automatica di tutti gli utenti
- Esempio di pre-registrazione: Nega l'iscrizione se il nome utente ha meno di cinque caratteri

Parametri del trigger Lambda di pre-registrazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "validationData": {
            "string": "string",
```

```
            . . .
         },
        "clientMetadata": {
            "string": "string",
            . . .
         }
    },

    "response": {
        "autoConfirmUser": "boolean",
        "autoVerifyPhone": "boolean",
        "autoVerifyEmail": "boolean"
    }
}
```

Parametri di richiesta di pre registrazione

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente. I nomi di attributo sono le chiavi.

validationData

Una o più coppie chiave-valore con i dati degli attributi utente che la tua app ha passato ad Amazon Cognito nella richiesta di creazione di un nuovo utente. Invia queste informazioni alla tua funzione Lambda nel ValidationData parametro della tua richiesta AdminCreateUsero SignUpAPI.

Amazon Cognito non imposta ValidationData i tuoi dati come attributi dell'utente che crei. ValidationData sono informazioni temporanee sull'utente fornite ai fini del trigger Lambda prima della registrazione.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di pre-registrazione. Puoi passare questi dati alla tua funzione Lambda utilizzando il ClientMetadata parametro nelle seguenti azioni API: AdminCreateUser, AdminRespondToAuthChallengeForgotPassword, e. SignUp

Parametri di risposta di pre-registrazione

Nella risposta, puoi impostare `autoConfirmUser` su `true` se desideri auto confermare l'utente. Puoi impostare `autoVerifyEmail` su `true` per auto verificare l'e-mail dell'utente. Puoi impostare `autoVerifyPhone` su `true` per auto verificare il numero di telefono dell'utente.

> **Note**
>
> I parametri di risposta `autoVerifyPhone`, `autoVerifyEmail` e `autoConfirmUser` vengono ignorati da Amazon Cognito quando la funzione Lambda di pre-registrazione viene attivata dall'API `AdminCreateUser`.

auto ConfirmUser

Impostare su `true` per auto confermare l'utente o altrimenti su `false`.

auto VerifyEmail

Imposta su `true` per definire come verificato il messaggio e-mail di un utente che sta effettuando la registrazione o altrimenti su `false`. Se `autoVerifyEmail` è impostato su `true`, l'attributo della `email` deve avere un valore valido e non nullo. In caso contrario, si verificherà un errore e l'utente non sarà in grado di completare la registrazione.

Se l'`email`attributo è selezionato come alias, viene creato un alias per l'indirizzo e-mail dell'utente quando `autoVerifyEmail` viene impostato. Se esiste già un alias con quell'indirizzo e-mail, l'alias viene spostato nel nuovo utente e l'indirizzo e-mail dell'utente precedente viene contrassegnato come non verificato. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

auto VerifyPhone

Imposta su `true` per impostare come verificato il numero di telefono di un utente che sta effettuando la registrazione o altrimenti su `false`. Se `autoVerifyPhone` è impostato su `true`, l'attributo della `phone_number` deve avere un valore valido e non nullo. In caso contrario, si verificherà un errore e l'utente non sarà in grado di completare la registrazione.

Se l'`phone_number`attributo è selezionato come alias, viene creato un alias per il numero di telefono dell'utente quando `autoVerifyPhone` viene impostato. Se esiste già un alias con quel numero di telefono, l'alias viene spostato sul nuovo utente e il numero di telefono dell'utente

precedente viene contrassegnato come non verificato. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

Esempio di pre-registrazione: Auto-confirm utenti di un dominio registrato

Questo è un esempio di codice trigger Lambda. Il trigger di pre-registrazione viene richiamato immediatamente prima che Amazon Cognito elabori la richiesta di registrazione. Utilizza un attributo personalizzato custom:domain per confermare automaticamente i nuovi utenti di un determinato dominio e-mail. Tutti i nuovi utenti non appartenenti al dominio personalizzato vengono aggiunti al pool di utenti, ma non vengono confermati automaticamente.

Node.js

```
export const handler = async (event, context, callback) => {
  // Set the user pool autoConfirmUser flag after validating the email domain
  event.response.autoConfirmUser = false;

  // Split the email address so we can compare domains
  var address = event.request.userAttributes.email.split("@");

  // This example uses a custom attribute "custom:domain"
  if (event.request.userAttributes.hasOwnProperty("custom:domain")) {
    if (event.request.userAttributes["custom:domain"] === address[1]) {
      event.response.autoConfirmUser = true;
    }
  }

  // Return to Amazon Cognito
  callback(null, event);
};
```

Python

```
def lambda_handler(event, context):
    # It sets the user pool autoConfirmUser flag after validating the email domain
    event['response']['autoConfirmUser'] = False

    # Split the email address so we can compare domains
    address = event['request']['userAttributes']['email'].split('@')

    # This example uses a custom attribute 'custom:domain'
```

```
    if 'custom:domain' in event['request']['userAttributes']:
        if event['request']['userAttributes']['custom:domain'] == address[1]:
            event['response']['autoConfirmUser'] = True

    # Return to Amazon Cognito
    return event
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "request": {
        "userAttributes": {
            "email": "testuser@example.com",
            "custom:domain": "example.com"
        }
    },
    "response": {}
}
```

Esempio di pre-iscrizione: Auto-confirm e verifica automatica di tutti gli utenti

Questo esempio conferma tutti gli utenti e imposta lo stato di avvenuta verifica per gli attributi `email` e `phone_number` dell'utente se presenti. Inoltre, se l'aliasing è abilitato, gli alias vengono creati per `phone_number` e `email` quando è impostata la verifica automatica.

> **Note**
>
> Se esiste già un alias con lo stesso numero di telefono, l'alias viene spostato sul nuovo utente e quello dell'utente precedente viene contrassegnato come non verificato. `phone_number` Lo stesso vale per gli indirizzi e-mail. Per evitare che ciò accada, puoi utilizzare l'ListUsers API dei pool di utenti per verificare se esiste un utente esistente che sta già utilizzando il numero di telefono o l'indirizzo e-mail del nuovo utente come alias.

Node.js

```
exports.handler = (event, context, callback) => {
  // Confirm the user
  event.response.autoConfirmUser = true;

  // Set the email as verified if it is in the request
  if (event.request.userAttributes.hasOwnProperty("email")) {
    event.response.autoVerifyEmail = true;
  }

  // Set the phone number as verified if it is in the request
  if (event.request.userAttributes.hasOwnProperty("phone_number")) {
    event.response.autoVerifyPhone = true;
  }

  // Return to Amazon Cognito
  callback(null, event);
};
```

Python

```
def lambda_handler(event, context):
    # Confirm the user
    event['response']['autoConfirmUser'] = True

    # Set the email as verified if it is in the request
    if 'email' in event['request']['userAttributes']:
        event['response']['autoVerifyEmail'] = True

    # Set the phone number as verified if it is in the request
    if 'phone_number' in event['request']['userAttributes']:
        event['response']['autoVerifyPhone'] = True

    # Return to Amazon Cognito
    return event
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
  "request": {
    "userAttributes": {
      "email": "user@example.com",
      "phone_number": "+12065550100"
    }
  },
  "response": {}
}
```

Esempio di pre-registrazione: Nega l'iscrizione se il nome utente ha meno di cinque caratteri

Questo esempio verifica la lunghezza del nome utente in una richiesta di registrazione. L'esempio restituisce un errore se l'utente immette un nome con meno di cinque caratteri.

Node.js

```
export const handler = (event, context, callback) => {
    // Impose a condition that the minimum length of the username is 5 is imposed on
 all user pools.
    if (event.userName.length < 5) {
        var error = new Error("Cannot register users with username less than the
 minimum length of 5");
        // Return error to Amazon Cognito
        callback(error, event);
    }
    // Return to Amazon Cognito
    callback(null, event);
};
```

Python

```
def lambda_handler(event, context):
    if len(event['userName']) < 5:
        raise Exception("Cannot register users with username less than the minimum
 length of 5")
    # Return to Amazon Cognito
    return event
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
  "userName": "rroe",
  "response": {}
}
```

Trigger Lambda di post-conferma

Amazon Cognito richiama questo trigger dopo che un utente registrato conferma il proprio account utente. Nella funzione Lambda di post conferma, puoi inviare messaggi personalizzati o aggiungere richieste API personalizzate. Ad esempio, puoi eseguire query su un sistema esterno e compilare attributi aggiuntivi per l'utente. Amazon Cognito richiama questo trigger solo per gli utenti che effettuano la registrazione al pool di utenti, non per gli account utente creati con le tue credenziali dell'amministratore.

La richiesta contiene gli attributi attuali per l'utente confermato. Il tuo pool di utenti richiama la funzione di conferma successiva su ConfirmSignUpAdminConfirmSignUp, e. ConfirmForgotPassword Questo trigger si attiva anche quando gli utenti confermano l'iscrizione o la reimpostazione della password nell'accesso gestito.

Argomenti

- Parametri del trigger Lambda di post-conferma
- Esempio di post conferma

Parametri del trigger Lambda di post-conferma

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
```

```
    "request": {
            "userAttributes": {
                "string": "string",
                . . .
            },
            "clientMetadata": {
             "string": "string",
             . . .
            }
        },
    "response": {}
}
```

Parametri di richiesta di post-conferma

userAttributes

Una o più coppie chiave-valore che rappresentano gli attributi utente.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di post-conferma. Puoi passare questi dati alla tua funzione Lambda utilizzando il ClientMetadata parametro nelle seguenti azioni API: AdminConfirmSignUp, ConfirmForgotPasswordConfirmSignUp, e. SignUp

Parametri di risposta post-conferma

Nella risposta non è prevista la restituzione di alcuna informazione aggiuntiva.

Esempio di post conferma

Questa funzione Lambda di esempio invia un messaggio e-mail di conferma all'utente utilizzando Amazon SES. Per ulteriori informazioni, consulta la Guida per gli sviluppatori di Amazon Simple Storage Service.

Node.js

```
// Import required AWS SDK clients and commands for Node.js. Note that this requires
// the `@aws-sdk/client-ses` module to be either bundled with this code or included
// as a Lambda layer.
import { SES, SendEmailCommand } from "@aws-sdk/client-ses";
```

```
const ses = new SES();

const handler = async (event) => {
  if (event.request.userAttributes.email) {
    await sendTheEmail(
      event.request.userAttributes.email,
      `Congratulations ${event.userName}, you have been confirmed.`,
    );
  }
  return event;
};

const sendTheEmail = async (to, body) => {
  const eParams = {
    Destination: {
      ToAddresses: [to],
    },
    Message: {
      Body: {
        Text: {
          Data: body,
        },
      },
      Subject: {
        Data: "Cognito Identity Provider registration completed",
      },
    },
    // Replace source_email with your SES validated email address
    Source: "<source_email>",
  };
  try {
    await ses.send(new SendEmailCommand(eParams));
  } catch (err) {
    console.log(err);
  }
};

export { handler };
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella

console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "request": {
        "userAttributes": {
            "email": "user@example.com",
            "email_verified": true
        }
    },
    "response": {}
}
```

Trigger Lambda di pre-autenticazione

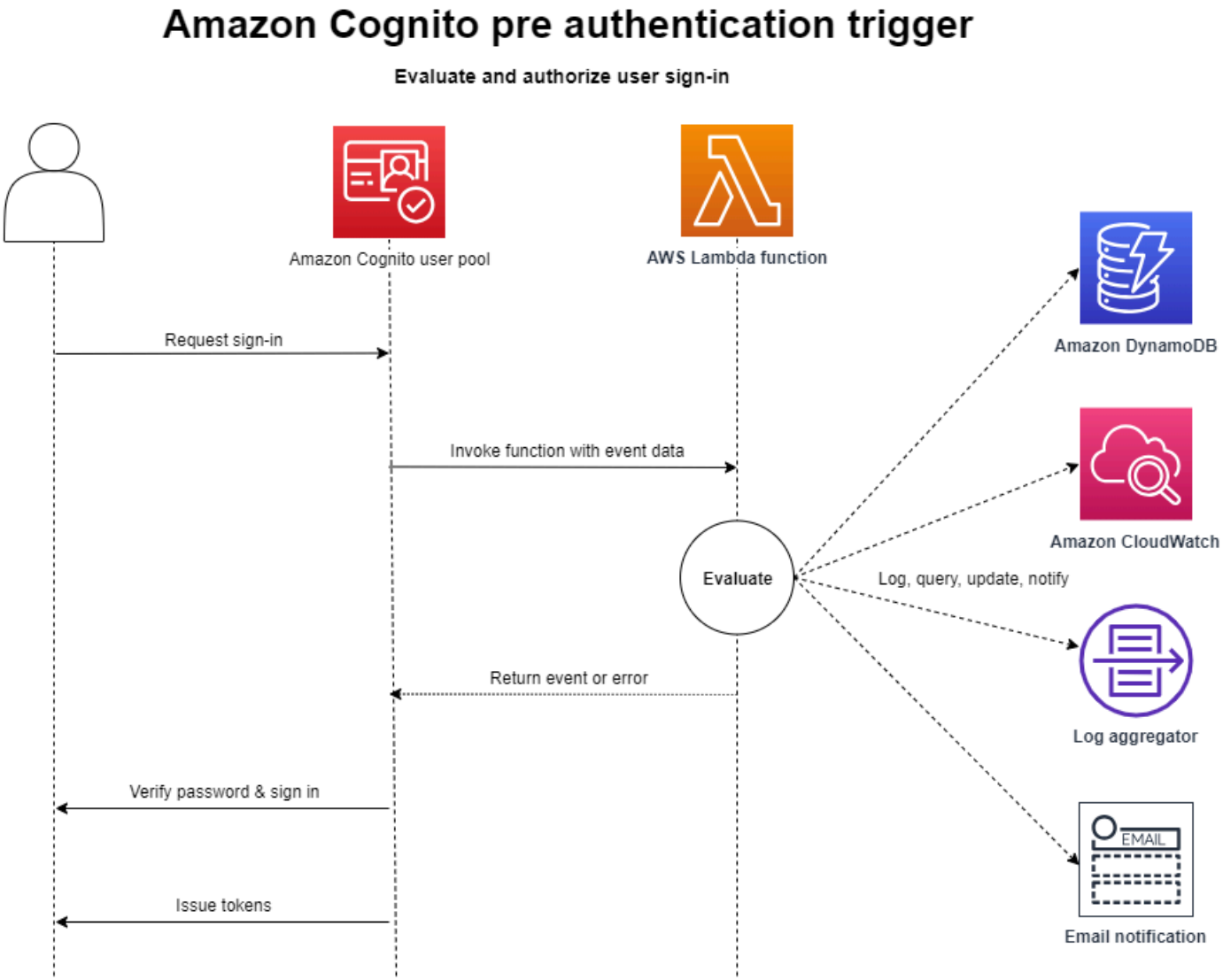
Amazon Cognito richiama questo trigger quando un utente prova a effettuare l'accesso, in modo da poter creare la convalida personalizzata che esegue azioni preparatorie. Ad esempio, puoi rifiutare la richiesta di autenticazione o registrare i dati di sessione in un sistema esterno.

> **Note**
>
> Questo trigger Lambda non si attiva quando un utente non esiste, a meno che l'`PreventUserExistenceErrors`impostazione di un client di app per pool di utenti non sia impostata su. `ENABLED` Inoltre, il rinnovo di una sessione di autenticazione esistente non attiva questo trigger.

Argomenti

- Panoramica del flusso
- Parametri di attivazione Lambda di pre-autenticazione
- Esempio di pre autenticazione

Panoramica del flusso



La richiesta include i dati di convalida dal client che provengono dai valori di `ClientMetadata` trasmessi dall'app alle operazioni API `InitiateAuth` e `AdminInitiateAuth` del pool di utenti.

Per ulteriori informazioni, consulta Un esempio di sessione di autenticazione.

Parametri di attivazione Lambda di pre-autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "validationData": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {}
}
```

Parametri di richiesta di pre-autenticazione

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

utente NotFound

Quando imposti `PreventUserExistenceErrors` su `ENABLED` per il tuo client del bacino d'utenza, Amazon Cognito popola questo valore booleano.

validationData

Una o più coppie chiave-valore contenenti i dati di convalida nella richiesta di registrazione dell'utente. Per passare questi dati alla funzione Lambda, usa il ClientMetadata parametro nelle azioni InitiateAuthe AdminInitiateAuthAPI.

Parametri di risposta di pre-autenticazione

Amazon Cognito non elabora le informazioni aggiuntive che la funzione restituisce nella risposta. La funzione può restituire un errore per rifiutare il tentativo di accesso o utilizzare le operazioni API per eseguire query e modificare le risorse.

Esempio di pre autenticazione

Questa funzione di esempio impedisce agli utenti di effettuare l'accesso al pool di utenti con un client dell'app specifico. Poiché la funzione Lambda di preautenticazione non viene richiamata quando l'utente dispone di una sessione esistente, questa funzione impedisce solo nuove sessioni con l'ID del client dell'app che desideri bloccare.

Node.js

```
const handler = async (event) => {
  if (
    event.callerContext.clientId === "user-pool-app-client-id-to-be-blocked"
  ) {
    throw new Error("Cannot authenticate users from this user pool app client");
  }

  return event;
};

export { handler };
```

Python

```
def lambda_handler(event, context):
    if event['callerContext']['clientId'] == "<user pool app client id to be
 blocked>":
        raise Exception("Cannot authenticate users from this user pool app client")

    # Return to Amazon Cognito
    return event
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "callerContext": {
```

```
        "clientId": "<user pool app client id to be blocked>"
    },
    "response": {}
}
```

Trigger Lambda di post-autenticazione

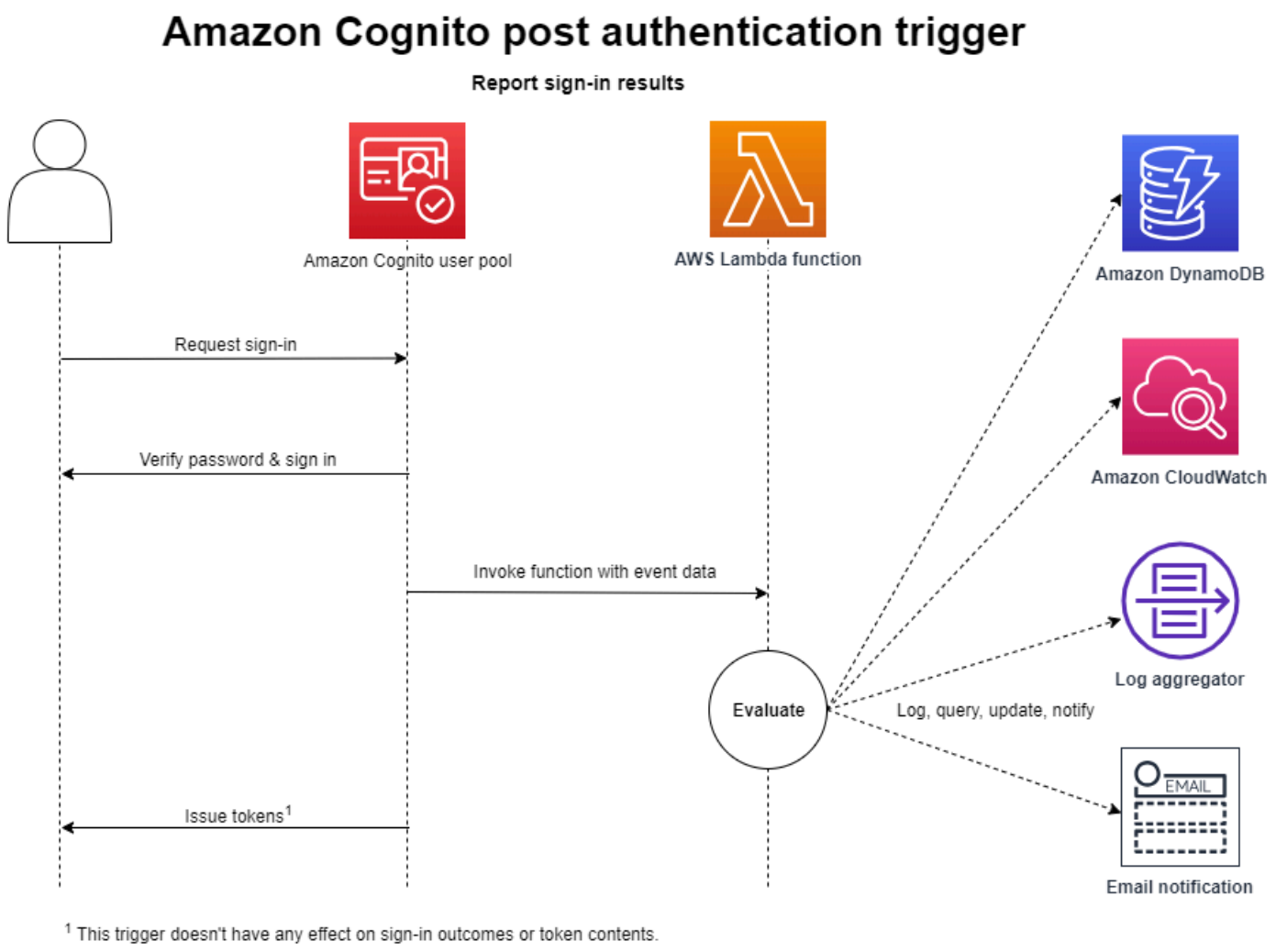
Il trigger di post-autenticazione non modifica il flusso di autenticazione per un utente. Amazon Cognito richiama questa Lambda dopo il completamento dell'autenticazione, prima che un utente abbia ricevuto i token. Aggiungi un trigger di post-autenticazione quando desideri aggiungere una post-elaborazione personalizzata degli eventi di autenticazione, ad esempio la registrazione o le modifiche del profilo utente che si riflettono all'accesso successivo.

Una Lambda dopo l'autenticazione che non restituisce il corpo della richiesta ad Amazon Cognito può comunque causare il mancato completamento dell'autenticazione. Per ulteriori informazioni, consulta Cose da sapere sui trigger Lambda.

Argomenti

- Panoramica sul flusso dell'autenticazione
- Parametri del trigger Lambda di post autenticazione
- Esempio di post autenticazione

Panoramica sul flusso dell'autenticazione



Per ulteriori informazioni, consulta Un esempio di sessione di autenticazione.

Parametri del trigger Lambda di post autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
```

```
            . . .
        },
        "newDeviceUsed": boolean,
        "clientMetadata": {
            "string": "string",
            . . .
           }
       },
    "response": {}
}
```

Parametri di richiesta di post autenticazione

nuovo DeviceUsed

Questo contrassegno indica se l'utente ha effettuato l'accesso da un nuovo dispositivo. Amazon Cognito imposta questo contrassegno solo se il valore dei dispositivi memorizzati del bacino d'utenza è impostato su `Always` o `User Opt-In`.

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di post-autenticazione. Per passare questi dati alla funzione Lambda, puoi utilizzare il ClientMetadata parametro nelle azioni AdminRespondToAuthChallengee RespondToAuthChallengeAPI. Amazon Cognito non include i dati del ClientMetadata parametro AdminInitiateAuthe le operazioni InitiateAuthAPI nella richiesta che passa alla funzione di post-autenticazione.

Parametri di risposta di post autenticazione

Amazon Cognito non prevede di restituire ulteriori informazioni nella risposta. La funzione può utilizzare le operazioni API per interrogare e modificare le risorse o registrare i metadati degli eventi in un sistema esterno.

Esempio di post autenticazione

Questo esempio di funzione Lambda di post-autenticazione invia i dati di un accesso riuscito a Logs. CloudWatch

Node.js

```
const handler = async (event) => {
  // Send post authentication data to Amazon CloudWatch logs
  console.log("Authentication successful");
  console.log("Trigger function =", event.triggerSource);
  console.log("User pool = ", event.userPoolId);
  console.log("App client ID = ", event.callerContext.clientId);
  console.log("User ID = ", event.userName);

  return event;
};

export { handler };
```

Python

```
import os
def lambda_handler(event, context):

    # Send post authentication data to Cloudwatch logs
    print ("Authentication successful")
    print ("Trigger function =", event['triggerSource'])
    print ("User pool = ", event['userPoolId'])
    print ("App client ID = ", event['callerContext']['clientId'])
    print ("User ID = ", event['userName'])

    # Return to Amazon Cognito
    return event
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
  "triggerSource": "testTrigger",
  "userPoolId": "testPool",
  "userName": "testName",
```

```
  "callerContext": {
      "clientId": "12345"
  },
  "response": {}
}
```

Trigger Lambda di federazione in entrata

Il trigger di federazione in entrata trasforma gli attributi utente federati durante il processo di autenticazione con provider di identità esterni. Quando gli utenti si autenticano tramite provider di identità configurati, questo trigger consente di modificare le risposte dei provider SAML e OIDC esterni intercettando e trasformando i dati nel processo di autenticazione, fornendo il controllo programmatico su come i pool di utenti di Amazon Cognito gestiscono gli utenti federati e i relativi attributi.

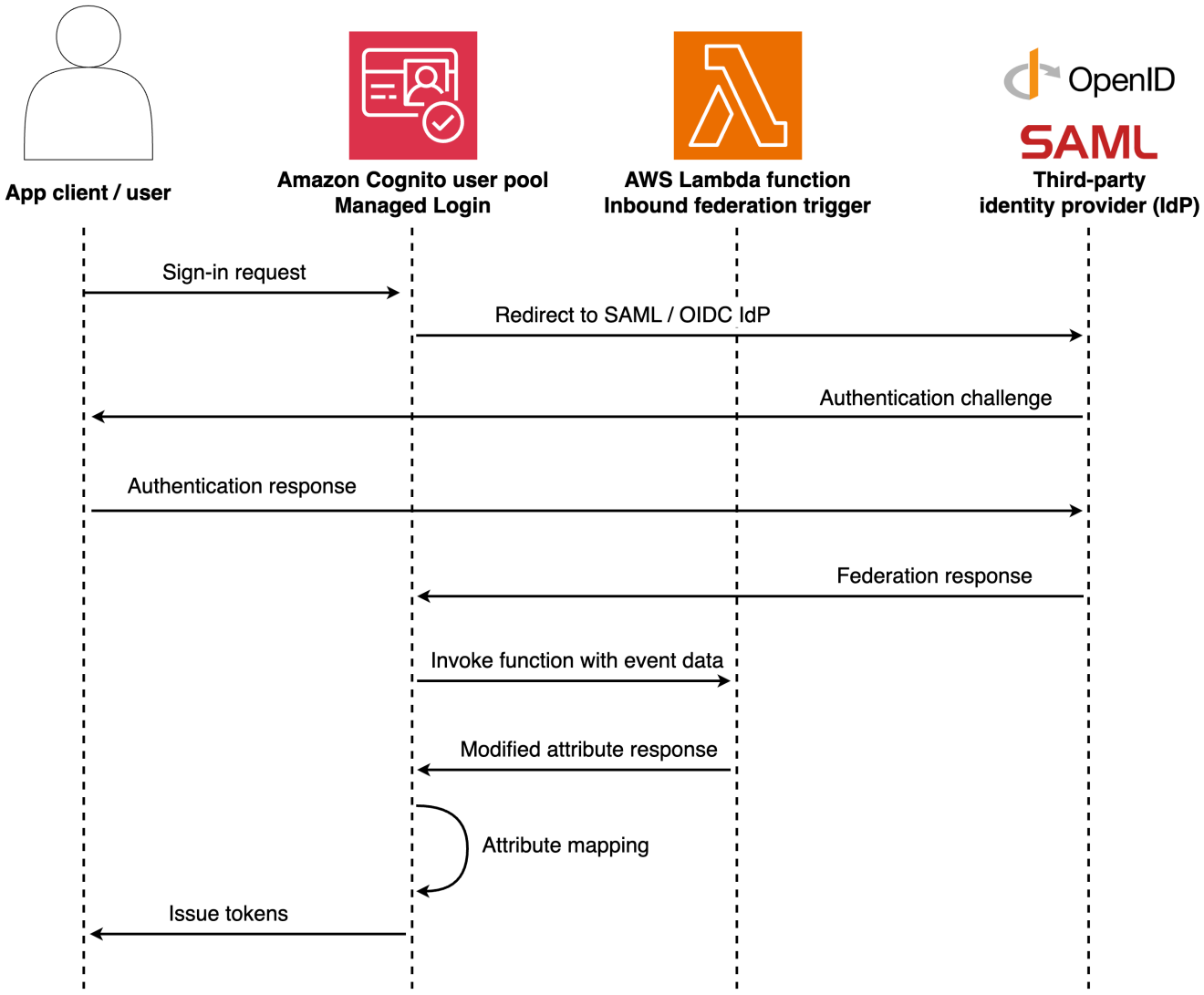
Usa questo trigger per aggiungere, sovrascrivere o sopprimere gli attributi prima di creare nuovi utenti o aggiornare i profili utente federati esistenti. Questo trigger riceve gli attributi non elaborati del provider di identità come input e restituisce gli attributi modificati che Amazon Cognito applica al profilo utente.

Argomenti

- Panoramica del flusso
- Parametri di attivazione Lambda della federazione in entrata
- Esempio di federazione in entrata: gestione dell'appartenenza a un gruppo
- Esempio di federazione in entrata: tronca gli attributi di grandi dimensioni
- Esempio di federazione in entrata: registrazione degli eventi di federazione

Panoramica del flusso

Quando un utente si autentica con un provider di identità esterno, Amazon Cognito richiama il trigger di federazione in entrata prima di creare o aggiornare il profilo utente. Il trigger riceve gli attributi non elaborati dal provider di identità e può trasformarli prima che Amazon Cognito li memorizzi. Questo flusso si verifica sia per i nuovi utenti federati che per gli utenti esistenti che accedono nuovamente tramite la federazione.



Parametri di attivazione Lambda della federazione in entrata

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "version": "string",
    "triggerSource": "InboundFederation_ExternalProvider",
    "region": AWSRegion,
    "userPoolId": "string",
    "userName": "string",
```

```
    "callerContext": {
        "awsSdkVersion": "string",
        "clientId": "string"
    },
    "request": {
        "providerName": "string",
        "providerType": "string",
        "attributes": {
            "tokenResponse": {
                "access_token": "string",
                "token_type": "string",
                "expires_in": "string"
            },
            "idToken": {
                "sub": "string",
                "email": "string",
                "email_verified": "string"
            },
            "userInfo": {
                "email": "string",
                "given_name": "string",
                "family_name": "string"
            },
            "samlResponse": {
                "string": "string"
            }
        }
    },
    "response": {
        "userAttributesToMap": {
            "string": "string"
        }
    }
}
```

Parametri della richiesta di federazione in entrata

Nome del provider

Il nome del provider di identità esterno.

ProviderType

Il tipo di provider di identità esterno. Valori validi:`OIDC`,`SAML`,`Facebook`,`Google`,`SignInWithApple`,`LoginWithAmazon`.

attributes

Gli attributi non elaborati ricevuti dal provider di identità prima dell'elaborazione. La struttura varia in base al tipo di provider.

Attributes.tokenResponse

Dati di risposta del token OAuth dall'endpoint. `/token` Disponibile solo per OIDC e fornitori di servizi sociali. Contiene`access_token`,`id_token`,, `refresh_token` `token_type`e`expires_in`, e. `scope`

Attributes.idToken

Token ID decodificato e convalidato dichiarato da JWT. Disponibile solo per OIDC e fornitori di servizi sociali. Contiene informazioni verificate sull'identità dell'utente, tra cui `sub` (identificatore utente univoco) e`mailname`, `iss` (emittente), `aud` (pubblico), `exp` (scadenza) e `iat` (ora di emissione).

Attributes.userInfo

Informazioni estese sul profilo utente dall'endpoint. UserInfo Disponibile solo per OIDC e fornitori di servizi sociali. Contiene attributi dettagliati del profilo come`given_name`,, `family_name` `pictureaddress`, e altri campi specifici del provider. Può essere vuoto se l'IdP non supporta l' UserInfo endpoint o se la chiamata all'endpoint fallisce.

Attributi.samlResponse

Attributi di asserzione SAML. Disponibile solo per i provider SAML. Contiene gli attributi della risposta SAML.

Parametri di risposta della federazione in entrata

utente AttributesToMap

Gli attributi utente da applicare al profilo utente.

Important

È necessario includere TUTTI gli attributi utente che si desidera conservare nella risposta, inclusi gli attributi che non si desidera modificare. Tutti gli attributi non inclusi nella `userAttributesToMap` risposta vengono eliminati e non memorizzati nel profilo utente. Questo vale sia per gli attributi modificati che per quelli non modificati.

Comportamento di risposta vuoto

Se si restituisce un oggetto vuoto `{}` per`userAttributesToMap`, tutti gli attributi originali del provider di identità vengono mantenuti invariati. Funziona come un no-op, come se la funzione Lambda non fosse mai stata eseguita. Ciò è diverso dall'omettere gli attributi, che li elimina.

Provider-specific attributi

La struttura di `request.attributes` varia in base a`providerType`. IDC e i fornitori di servizi sociali includono `tokenResponseidToken`, e `userInfo` oggetti. I provider SAML includono solo l'oggetto. `samlResponse`

Esempio di federazione in entrata: gestione dell'appartenenza a un gruppo

Questo esempio mostra come mappare i gruppi di provider di identità federati ai gruppi di pool di utenti di Amazon Cognito. Questa funzione estrae l'appartenenza al gruppo dalla risposta federata e aggiunge automaticamente gli utenti ai gruppi Amazon Cognito corrispondenti, eliminando la necessità di trigger post-autenticazione.

Node.js

```
exports.handler = async (event) => {
    const { providerType, attributes } = event.request;

    // Extract user attributes based on provider type
    let userAttributesFromIdp = {};
    if (providerType === 'SAML') {
        userAttributesFromIdp = attributes.samlResponse || {};
```

```
    } else {
        // For OIDC and Social providers, merge userInfo and idToken
        userAttributesFromIdp = {
            ...(attributes.userInfo || {}),
            ...(attributes.idToken || {})
        };
    }

    // Extract groups from federated response
    const federatedGroups = userAttributesFromIdp.groups?.split(',') || [];

    // Map federated groups to Cognito groups
    const groupMapping = {
        'Domain Admins': 'Administrators',
        'Engineering': 'Developers',
        'Sales': 'SalesTeam'
    };

    // Filter to only in-scope groups
    const mappedGroups = federatedGroups
        .map(group => groupMapping[group.trim()])
        .filter(group => group); // Remove undefined values

    // Pass through attributes with mapped groups as custom attribute
    const attributesToMap = {
        ...userAttributesFromIdp,
        'custom:user_groups': mappedGroups.join(',')
    };

    // Remove original groups attribute
    delete attributesToMap.groups;

    event.response.userAttributesToMap = attributesToMap;
    return event;
};
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "userPoolId": "us-east-1_XXXXXXXXX",
    "request": {
        "providerName": "CorporateAD",
        "providerType": "SAML",
        "attributes": {
            "samlResponse": {
                "email": "jane.smith@company.com",
                "given_name": "Jane",
                "family_name": "Smith",
                "groups": "Engineering,Domain Admins",
                "department": "Engineering"
            }
        }
    },
    "response": {
        "userAttributesToMap": {}
    }
}
```

Esempio di federazione in entrata: tronca gli attributi di grandi dimensioni

Questo esempio mostra come troncare i valori degli attributi che superano i limiti di archiviazione di Amazon Cognito. Questa funzione controlla ogni attributo del provider di identità. Se il valore di un attributo supera i 2048 caratteri, tronca il valore e aggiunge i puntini di sospensione per indicare il troncamento. Tutti gli altri attributi rimangono invariati.

Node.js

```
exports.handler = async (event) => {
    const MAX_ATTRIBUTE_LENGTH = 2048;

    // Get the identity provider attributes based on provider type
    const { providerType, attributes } = event.request;
    let idpAttributes = {};

    if (providerType === 'SAML') {
        idpAttributes = attributes.samlResponse || {};
    } else {
        // For OIDC and Social providers, merge userInfo and idToken
```

```
        idpAttributes = {
            ...(attributes.userInfo || {}),
            ...(attributes.idToken || {})
        };
    }

    const userAttributes = {};

    // Process each attribute
    for (const [key, value] of Object.entries(idpAttributes)) {
        if (typeof value === 'string' && value.length > MAX_ATTRIBUTE_LENGTH) {
            // Truncate the value and add ellipsis
            userAttributes[key] = value.substring(0, MAX_ATTRIBUTE_LENGTH - 3) +
 '...';
            console.log(`Truncated attribute ${key} from ${value.length} to
 ${userAttributes[key].length} characters`);
        } else {
            // Keep the original value
            userAttributes[key] = value;
        }
    }

    // Return the modified attributes
    event.response.userAttributesToMap = userAttributes;
    return event;
};
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "version": "string",
    "triggerSource": "InboundFederation_ExternalProvider",
    "region": "us-east-1",
    "userPoolId": "us-east-1_XXXXXXXXX",
    "userName": "ExampleProvider_12345",
    "callerContext": {
        "awsSdkVersion": "string",
```

```
        "clientId": "string"
    },
    "request": {
        "providerName": "ExampleProvider",
        "providerType": "OIDC",
        "attributes": {
            "tokenResponse": {
                "access_token": "abcDE...",
                "token_type": "Bearer",
                "expires_in": "3600"
            },
            "idToken": {
                "sub": "12345",
                "email": "user@example.com"
            },
            "userInfo": {
                "email": "user@example.com",
                "given_name": "Example",
                "family_name": "User",
                "bio": "This is a very long biography that contains more than 2048
 characters..."
            }
        }
    },
    "response": {
        "userAttributesToMap": {}
    }
}
```

Esempio di federazione in entrata: registrazione degli eventi di federazione

Questo esempio mostra come registrare gli eventi di autenticazione federata per il monitoraggio e il debug. Questa funzione di esempio acquisisce informazioni dettagliate sugli utenti federati e sui relativi attributi, fornendo visibilità sul processo di autenticazione.

Node.js

```
exports.handler = async (event) => {
    const { providerName, providerType, attributes } = event.request;

    // Extract user attributes based on provider type
    let userAttributesFromIdp = {};
```

```
    if (providerType === 'SAML') {
        userAttributesFromIdp = attributes.samlResponse || {};
    } else {
        // For OIDC and Social providers, merge userInfo and idToken
        userAttributesFromIdp = {
            ...(attributes.userInfo || {}),
            ...(attributes.idToken || {})
        };
    }

    // Log federated authentication details
    console.log(JSON.stringify({
        timestamp: new Date().toISOString(),
        providerName,
        providerType,
        userEmail: userAttributesFromIdp.email,
        attributeCount: Object.keys(userAttributesFromIdp).length,
        attributes: userAttributesFromIdp
    }));

    // Pass through all attributes unchanged
    event.response.userAttributesToMap = userAttributesFromIdp;
    return event;
};
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "version": "string",
    "triggerSource": "InboundFederation_ExternalProvider",
    "region": "us-east-1",
    "userPoolId": "us-east-1_XXXXXXXXX",
    "userName": "CorporateAD_john.doe",
    "callerContext": {
        "awsSdkVersion": "string",
        "clientId": "string"
    },
```

```
    "request": {
        "providerName": "CorporateAD",
        "providerType": "SAML",
        "attributes": {
            "samlResponse": {
                "email": "john.doe@company.com",
                "given_name": "John",
                "family_name": "Doe",
                "department": "Engineering",
                "employee_id": "EMP12345"
            }
        }
    },
    "response": {
        "userAttributesToMap": {}
    }
}
```

Output previsto dei CloudWatch log:

JSON

```
{
    "timestamp": "2025-01-14T21:17:40.153Z",
    "providerName": "CorporateAD",
    "providerType": "SAML",
    "userEmail": "john.doe@company.com",
    "attributeCount": 5,
    "attributes": {
        "email": "john.doe@company.com",
        "given_name": "John",
        "family_name": "Doe",
        "department": "Engineering",
        "employee_id": "EMP12345"
    }
}
```

Trigger Lambda di richieste di autenticazione personalizzate

Man mano che crei i flussi di autenticazione per il tuo pool di utenti Amazon Cognito, potresti scoprire di voler estendere il tuo modello di autenticazione oltre i flussi integrati. Un caso d'uso comune per

i trigger di sfida personalizzati consiste nell'implementare controlli di sicurezza aggiuntivi oltre a nome utente, password e autenticazione a più fattori (MFA). Una sfida personalizzata è qualsiasi domanda e risposta che puoi generare in un Lambda-supported linguaggio di programmazione. Ad esempio, potresti voler richiedere agli utenti di risolvere un CAPTCHA o rispondere a una domanda di sicurezza prima di poter effettuare l'autenticazione. Un'altra potenziale esigenza è l'integrazione con fattori o dispositivi di autenticazione specializzati. Oppure potresti aver già sviluppato un software che autentica gli utenti con una chiave di sicurezza hardware o un dispositivo biometrico. La definizione di successo dell'autenticazione per una sfida personalizzata è la risposta che la funzione Lambda accetta come corretta: una stringa fissa, ad esempio, o una risposta soddisfacente da un'API esterna.

Puoi avviare l'autenticazione con la tua sfida personalizzata e controllare completamente il processo di autenticazione, oppure puoi eseguire l'autenticazione nome utente e password prima che l'applicazione riceva la sfida personalizzata.

Il trigger Lambda della sfida di autenticazione personalizzata:

Definisce

Avvia una sequenza di sfida. Determina se si desidera avviare una nuova sfida, contrassegnare l'autenticazione come completata o interrompere il tentativo di autenticazione.

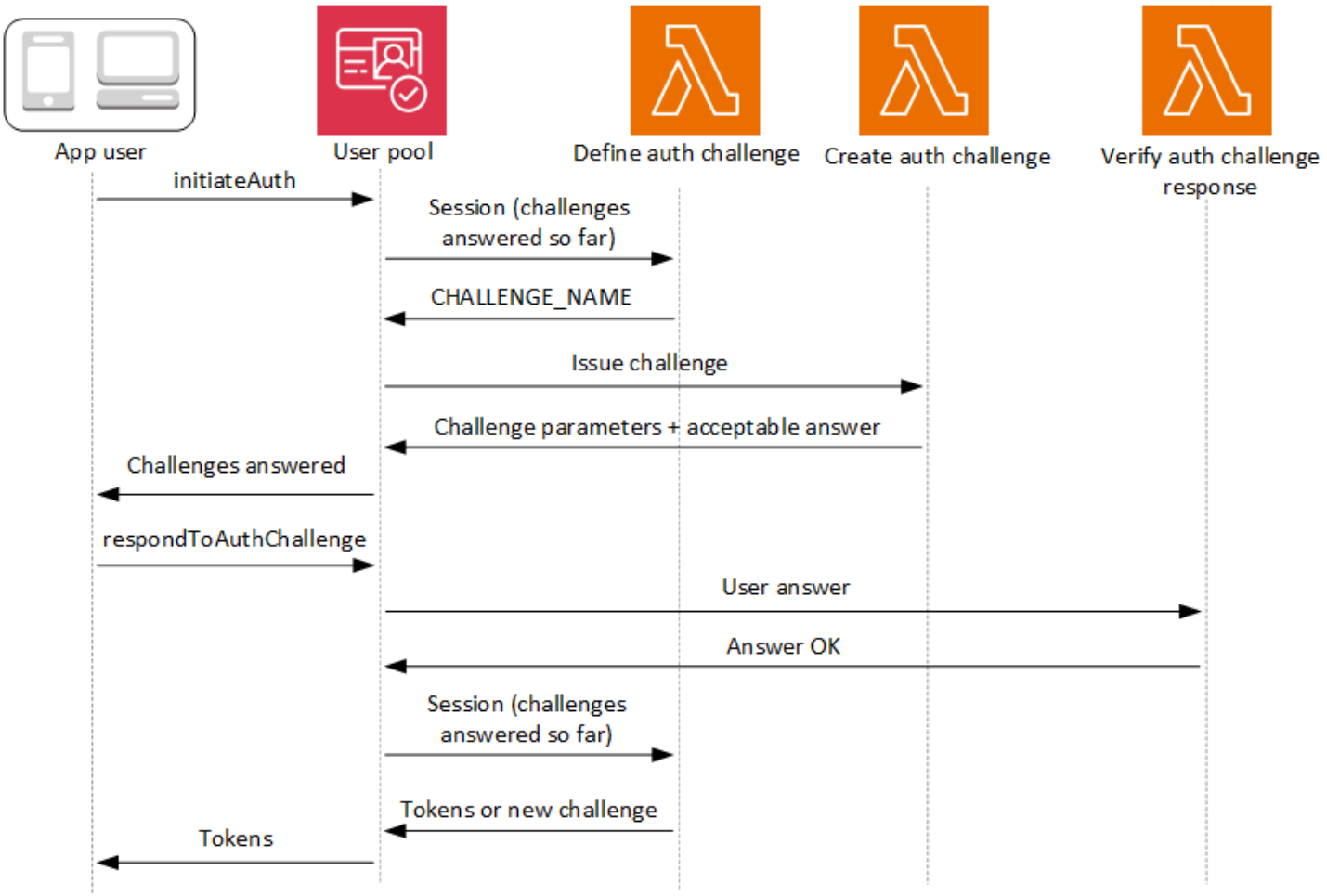
Crea

Invia all'applicazione la domanda a cui l'utente deve rispondere. Questa funzione potrebbe presentare una domanda di sicurezza o un collegamento a un CAPTCHA che l'applicazione dovrebbe mostrare all'utente.

Verifica

Conosce la risposta prevista e la confronta con la risposta fornita dall'applicazione nella risposta alla sfida. La funzione potrebbe richiamare l'API del servizio CAPTCHA per recuperare i risultati attesi del tentativo di soluzione dell'utente.

Queste tre funzioni Lambda si concatenano per presentare un meccanismo di autenticazione completamente sotto il tuo controllo e progettato da te. Poiché l'autenticazione personalizzata richiede la logica dell'applicazione nel client e nelle funzioni Lambda, non è possibile elaborare l'autenticazione personalizzata all'interno dell'accesso gestito. Questo sistema di autenticazione richiede un impegno aggiuntivo da parte degli sviluppatori. L'applicazione deve eseguire il flusso di autenticazione con l'API dei pool di utenti e gestire la sfida risultante con un'interfaccia di accesso personalizzata che ponga la domanda al centro della sfida di autenticazione personalizzata.



Per ulteriori informazioni sull'implementazione dell'autenticazione personalizzata, consulta Flusso di autenticazione personalizzato e sfide

Autenticazione tra le operazioni API InitiateAutho AdminInitiateAuth, e RespondToAuthChallengeo AdminRespondToAuthChallenge. In questo flusso, l'autenticazione di un utente viene effettuata rispondendo a richieste successive di autenticazione finché l'utente non viene bloccato in caso di esito negativo oppure riceve i relativi token. Una risposta alla sfida potrebbe essere una nuova sfida. In questo caso, l'applicazione risponde tutte le volte necessarie alle nuove sfide. L'autenticazione ha successo quando la funzione define auth challenge analizza i risultati finora ottenuti, determina che tutte le sfide hanno avuto risposta e restituisce i risultati. `IssueTokens`

> **Perché il trigger di autenticazione personalizzato non si è attivato**
>
> Se la funzione di risposta Create auth challenge o Verify auth challenge non viene richiamata, la causa è in genere una delle seguenti:

- Il client dell'app deve consentire un flusso di autenticazione personalizzato. Attiva `ALLOW_CUSTOM_AUTH` (o, per l'accesso basato sulla scelta,`ALLOW_USER_AUTH`) sul client dell'app e inizia ad accedere con il flusso corrispondente. Un pool che consente solo l'autenticazione SRP o tramite password non entra mai nella sequenza di sfida personalizzata.
- Amazon Cognito richiama le tre funzioni nell'ordine. Chiama sempre per primo Define auth challenge e chiama Create auth challenge e Verify auth challenge response solo quando ritorna la funzione Define auth challenge. `challengeName: CUSTOM_CHALLENGE` Se Create non è in esecuzione, conferma che Define ritorni `challengeName: CUSTOM_CHALLENGE` con `issueTokens: false` `efailAuthentication: false`, anziché impostare o. `issueTokens: true` `failAuthentication: true`
- Tutti e tre i trigger devono essere assegnati al pool di utenti. Un incarico mancante per uno di essi interrompe la catena.

Argomenti

- Autenticazione SRP nei flussi di sfida personalizzati
- Definizione del trigger Lambda di una richiesta di autenticazione
- Creazione del trigger Lambda di una richiesta di autenticazione
- Trigger Lambda di una verifica di risposta di una richiesta di autenticazione

Autenticazione SRP nei flussi di sfida personalizzati

Puoi fare in modo che Amazon Cognito verifichi le password degli utenti prima di emettere le tue richieste personalizzate. Tutti i trigger Lambda associati alla categoria Authentication delle quote di frequenza di richiesta vengono eseguiti quando si esegue l'autenticazione SRP in un challenge flow personalizzato. Ecco una panoramica del processo:

1. La tua app avvia la procedura di accesso chiamando `InitiateAuth` o `AdminInitiateAuth` con la mappatura `AuthParameters`. I parametri devono includere `CHALLENGE_NAME: SRP_A,` e i valori per `SRP_A` e `USERNAME`.
2. Amazon Cognito invoca il tuo trigger Lambda define auth challenge con una sessione iniziale contenente `challengeName: SRP_A` e `challengeResult: true`.

3. Dopo aver ricevuto questi input, la funzione Lambda risponde con `challengeName: PASSWORD_VERIFIER`, `issueTokens: false`, `failAuthentication: false`.
4. Se la verifica della password ha esito positivo, Amazon Cognito richiama nuovamente la funzione Lambda con una nuova sessione contenente `challengeName: PASSWORD_VERIFIER` e `challengeResult: true`.
5. La funzione Lambda avvia le tue sifde personalizzate rispondendo con `challengeName: CUSTOM_CHALLENGE`, `issueTokens: false` e `failAuthentication: false`. Se non desideri avviare il flusso di autenticazione personalizzato con la verifica della password, è possibile avviare l'accesso con la mappa `AuthParameters` che include `CHALLENGE_NAME: CUSTOM_CHALLENGE`.
6. Il loop di sfide si ripete finché non viene data risposta a tutte le sfide.

Di seguito è riportato un esempio di `InitiateAuth` richiesta iniziale che precede l'autenticazione personalizzata con un flusso SRP.

```
{
    "AuthFlow": "CUSTOM_AUTH",
    "ClientId": "1example23456789",
    "AuthParameters": {
        "CHALLENGE_NAME": "SRP_A",
        "USERNAME": "testuser",
        "SRP_A": "[SRP_A]",
        "SECRET_HASH": "[secret hash]"
    }
}
```

Reimpostazione della password nel flusso SRP di autenticazione personalizzato

Quando `FORCE_CHANGE_PASSWORD` lo status degli utenti è impostato, il flusso di autenticazione personalizzato deve integrare la fase di modifica della password, mantenendo al contempo l'integrità dei problemi di autenticazione. Amazon Cognito richiama il trigger Lambda della tua sfida di autenticazione definita durante la sfida. `NEW_PASSWORD_REQUIRED` In questo scenario, un utente che accede con un flusso di verifica personalizzato e l'autenticazione SRP può impostare una nuova password se si trova in uno stato di reimpostazione della password.

Quando gli utenti sono nello `FORCE_CHANGE_PASSWORD` stato `RESET_REQUIRED` o, devono rispondere a una sfida con un`NEW_PASSWORD_REQUIRED`. `NEW_PASSWORD` Nell'autenticazione personalizzata con SRP, Amazon Cognito restituisce `NEW_PASSWORD_REQUIRED` una sfida

dopo che gli utenti hanno completato la PASSWORD_VERIFIER sfida SRP. Il trigger define auth challenge riceve entrambi i risultati della sfida nell'`session`array e può procedere con ulteriori sfide personalizzate dopo che l'utente ha modificato con successo la propria password.

Il trigger Lambda define auth challenge deve gestire la sequenza di sfida tramite l'autenticazione SRP, la reimpostazione della password e le successive sfide personalizzate. Il trigger riceve una serie di sfide completate nel `session` parametro, incluse entrambe PASSWORD_VERIFIER e i risultati. NEW_PASSWORD_REQUIRED Per un esempio di implementazione, vedereDefinizione di un esempio di una richiesta di autenticazione.

Fasi del flusso di autenticazione

Per gli utenti che devono verificare la propria password prima delle sfide personalizzate, la procedura segue questi passaggi:

1. La tua app avvia la procedura di accesso chiamando `InitiateAuth` o `AdminInitiateAuth` con la mappatura `AuthParameters`. I parametri devono includere `CHALLENGE_NAME: SRP_A` e valori per SRP_A e`USERNAME`.
2. Amazon Cognito invoca il tuo trigger Lambda define auth challenge con una sessione iniziale contenente `challengeName: SRP_A` e `challengeResult: true`.
3. Dopo aver ricevuto questi input, la funzione Lambda risponde con `challengeName: PASSWORD_VERIFIER`, `issueTokens: false`, `failAuthentication: false`.
4. Se la verifica della password ha esito positivo, si verifica una delle due seguenti situazioni:

   Per gli utenti in stato normale:

   Amazon Cognito richiama nuovamente la funzione Lambda con una nuova sessione contenente and. `challengeName: PASSWORD_VERIFIER` `challengeResult: true`

   La funzione Lambda avvia le tue sifde personalizzate rispondendo con `challengeName: CUSTOM_CHALLENGE`, `issueTokens: false` e `failAuthentication: false`.

   Per gli utenti che rientrano nel nostro status: **`RESET_REQUIRED` `FORCE_CHANGE_PASSWORD`**

   Amazon Cognito richiama la funzione Lambda con una sessione contenente e. `challengeName: PASSWORD_VERIFIER` `challengeResult: true`

   La tua funzione Lambda dovrebbe rispondere con `challengeName: NEW_PASSWORD_REQUIRED`, `issueTokens: false` e `failAuthentication: false`.

Dopo aver modificato con successo la password, Amazon Cognito richiama la funzione Lambda con una sessione contenente sia i risultati che. `PASSWORD_VERIFIER` `NEW_PASSWORD_REQUIRED`

La funzione Lambda avvia le tue sifde personalizzate rispondendo con `challengeName: CUSTOM_CHALLENGE`, `issueTokens: false` e `failAuthentication: false`.

5. Il loop di sfide si ripete finché non viene data risposta a tutte le sfide.

Se non desideri avviare il flusso di autenticazione personalizzato con la verifica della password, è possibile avviare l'accesso con la mappa `AuthParameters` che include `CHALLENGE_NAME: CUSTOM_CHALLENGE`.

Gestione della sessione

Il flusso di autenticazione mantiene la continuità della sessione attraverso una serie di ID di sessione e risultati delle sfide. Ogni risposta alla sfida genera un nuovo ID di sessione per prevenire errori di riutilizzo della sessione, il che è particolarmente importante per i flussi di autenticazione a più fattori.

I risultati della sfida vengono archiviati in ordine cronologico nell'array di sessioni ricevuto dai trigger Lambda. Per gli utenti con `FORCE_CHANGE_PASSWORD` status, l'array di sessione contiene:

1. `session[0]`- SRP_A Sfida iniziale
2. `session[1]`- `PASSWORD_VERIFIER` risultato
3. `session[2]`- `NEW_PASSWORD_REQUIRED` risultato
4. Elementi successivi: risultati di ulteriori sfide personalizzate

Esempio di flusso di autenticazione

L'esempio seguente mostra un flusso di autenticazione personalizzato completo per un utente con `FORCE_CHANGE_PASSWORD` status che deve completare sia la modifica della password che una sfida CAPTCHA personalizzata.

1. InitiateAuth richiesta

```
{
    "AuthFlow": "CUSTOM_AUTH",
    "ClientId": "1example23456789",
    "AuthParameters": {
```

```
        "CHALLENGE_NAME": "SRP_A",
        "USERNAME": "testuser",
        "SRP_A": "[SRP_A]"
    }
}
```

2. InitiateAuth risposta

```
{
    "ChallengeName": "PASSWORD_VERIFIER",
    "ChallengeParameters": {
        "USER_ID_FOR_SRP": "testuser"
    },
    "Session": "[session_id_1]"
}
```

3. RespondToAuthChallenge richiesta con **PASSWORD_VERIFIER**

```
{
    "ChallengeName": "PASSWORD_VERIFIER",
    "ClientId": "1example23456789",
    "ChallengeResponses": {
        "PASSWORD_CLAIM_SIGNATURE": "[claim_signature]",
        "PASSWORD_CLAIM_SECRET_BLOCK": "[secret_block]",
        "TIMESTAMP": "[timestamp]",
        "USERNAME": "testuser"
    },
    "Session": "[session_id_1]"
}
```

4. RespondToAuthChallenge risposta con **NEW_PASSWORD_REQUIRED** sfida

```
{
    "ChallengeName": "NEW_PASSWORD_REQUIRED",
    "ChallengeParameters": {},
    "Session": "[session_id_2]"
}
```

5. RespondToAuthChallenge richiesta con **NEW_PASSWORD_REQUIRED**

```
{
    "ChallengeName": "NEW_PASSWORD_REQUIRED",
    "ClientId": "1example23456789",
```

```
    "ChallengeResponses": {
        "NEW_PASSWORD": "[password]",
        "USERNAME": "testuser"
    },
    "Session": "[session_id_2]"
}
```

6. RespondToAuthChallenge risposta con sfida personalizzata CAPTCHA

```
{
    "ChallengeName": "CUSTOM_CHALLENGE",
    "ChallengeParameters": {
        "captchaUrl": "url/123.jpg"
    },
    "Session": "[session_id_3]"
}
```

7. RespondToAuthChallenge richiesta con risposta alla sfida personalizzata CAPTCHA

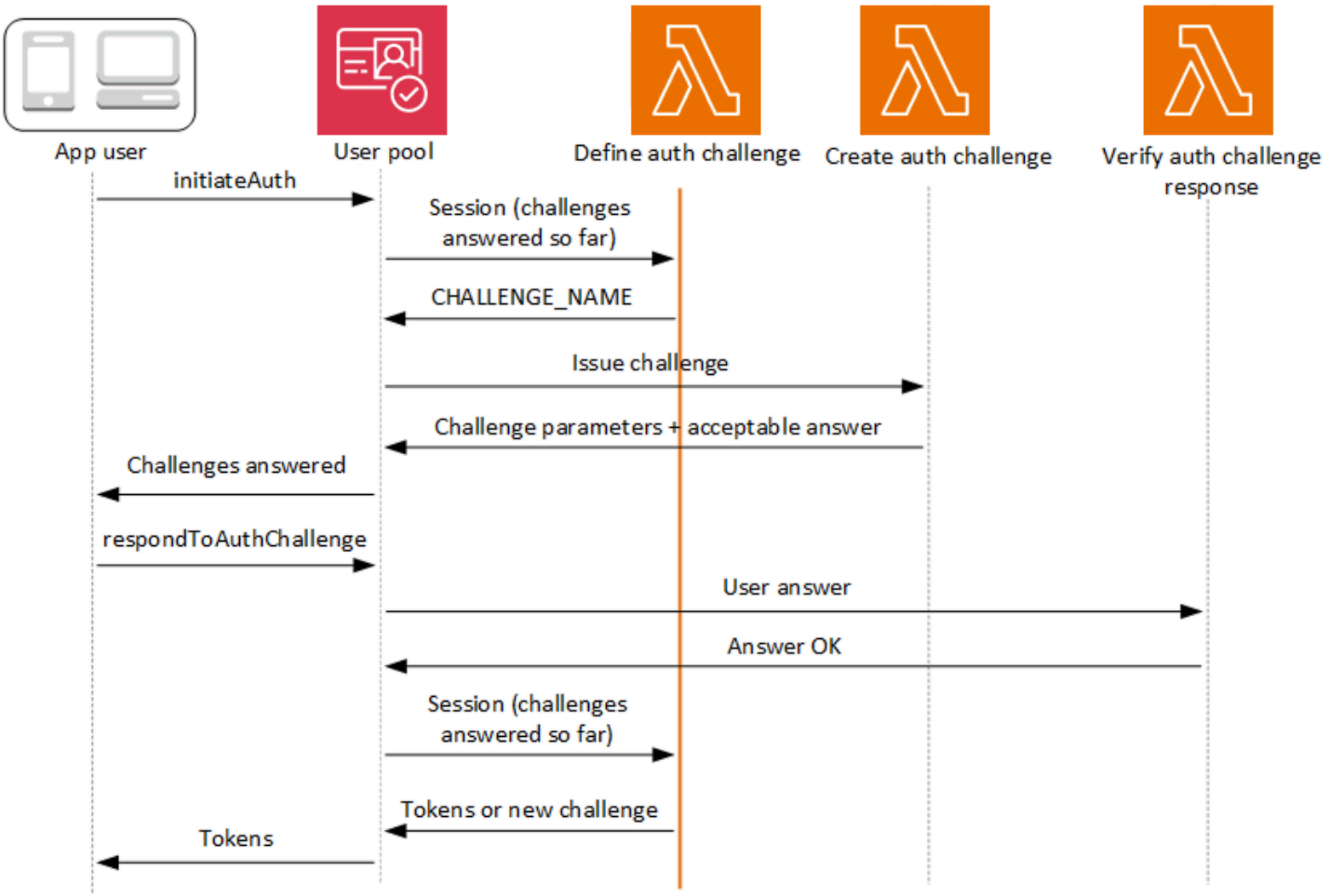
```
{
    "ChallengeName": "CUSTOM_CHALLENGE",
    "ClientId": "1example23456789",
    "ChallengeResponses": {
        "ANSWER": "123",
        "USERNAME": "testuser"
    },
    "Session": "[session_id_3]"
}
```

6. Risposta finale di successo

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "IdToken": "eyJra789ghiEXAMPLE",
        "RefreshToken": "eyJjd123abcEXAMPLE",
        "TokenType": "Bearer"
    },
    "ChallengeParameters": {}
}
```

Definizione del trigger Lambda di una richiesta di autenticazione

Il trigger define auth challenge è una funzione Lambda che mantiene la sequenza di sfida in un flusso di autenticazione personalizzato. Dichiara il successo o il fallimento della sequenza di sfida e imposta la sfida successiva se la sequenza non è ancora completa.



Definisci la sfida di autenticazione

Amazon Cognito richiama questo trigger per avviare il flusso di autenticazione personalizzato.

La richiesta per questo trigger Lambda include `session`. Il parametro `session` è una matrice contenente tutte le sfide a cui l'utente viene sottoposto nel processo di autenticazione in corso. La richiesta include anche il risultato corrispondente. La matrice `session` archivia i dettagli della sfida (`ChallengeResult`) in ordine cronologico. La richiesta `session[0]` rappresenta la prima sfida che l'utente riceve.

Argomenti

- Parametri del trigger Lambda di definizione delle richieste di autenticazione
- Definizione di un esempio di una richiesta di autenticazione

Parametri del trigger Lambda di definizione delle richieste di autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
                . . .
        },
        "session": [
            ChallengeResult,
            . . .
        ],
        "clientMetadata": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {
        "challengeName": "string",
        "issueTokens": boolean,
        "failAuthentication": boolean
    }
}
```

Parametri di richiesta di definizione sfida di autenticazione

Quando Amazon Cognito richiama la tua funzione Lambda, fornisce i seguenti parametri:

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

utente NotFound

Valore booleano che viene popolato da Amazon Cognito quando `PreventUserExistenceErrors` è impostato su `ENABLED` per il client del bacino d'utenza. Un valore `true` indica che l'ID utente (nome utente, indirizzo e-mail e altri dettagli) non corrisponde ad alcun utente esistente. Quando `PreventUserExistenceErrors` è impostato su `ENABLED`, il servizio non informa l'app riguardo a utenti inesistenti. Raccomandiamo che le tue funzioni Lambda mantengano la stessa esperienza utente e tengano conto della latenza. In questo modo, l'intermediario non può rilevare comportamenti diversi quando l'utente esiste o non esiste.

sessione

Una matrice di elementi `ChallengeResult`. Ognuna contiene i seguenti elementi:

challengeName

Uno dei seguenti tipi di sfida: `CUSTOM_CHALLENGESRP_A`,`PASSWORD_VERIFIER`,`SMS_MFA`, `EMAIL_OTPSOFTWARE_TOKEN_MFA`,`DEVICE_SRP_AUTH`,`DEVICE_PASSWORD_VERIFIER`, o`ADMIN_NO_SRP_AUTH`.

Quando la funzione define auth challenge genera una `PASSWORD_VERIFIER` sfida per un utente che ha configurato l'autenticazione a più fattori, Amazon Cognito la segue con `SMS_MFA` una sfida`EMAIL_OTP`, o. `SOFTWARE_TOKEN_MFA` Queste sono le istruzioni per un codice di autenticazione a più fattori. Nella tua funzione, includi la gestione degli eventi di input e delle `SMS_MFA` sfide`EMAIL_OTP`. `SOFTWARE_TOKEN_MFA` Non è necessario richiamare alcuna sfida MFA nella funzione define auth challenge.

> **Important**
>
> Quando la funzione determina se un utente ha completato l'autenticazione ed è necessario emettere token, controlla sempre `challengeName` nella funzione Definisci la sfida di autenticazione e verifica che corrisponda al valore previsto.

challengeResult

Imposta su `true` se l'utente ha completato con successo la sfida o altrimenti su `false`.

challengeMetaData

Il tuo nome per la sfida personalizzata. Usato solo se `challengeName` è `CUSTOM_CHALLENGE`.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di definizione di una richiesta di autenticazione. Per passare questi dati alla funzione Lambda, puoi utilizzare il `ClientMetadata` parametro nelle operazioni AdminRespondToAuthChallengee RespondToAuthChallengeAPI. La richiesta che richiama la funzione define auth challenge non include i dati passati nel ClientMetadata parametro e nelle operazioni API. AdminInitiateAuthInitiateAuth

Parametri di risposta di definizione sfida di autenticazione

Nella risposta, puoi restituire la prossima fase del processo di autenticazione.

challengeName

Una stringa che include il nome della sfida successiva. Se desideri presentare una nuova sfida al tuo utente, specifica qui il nome della sfida.

issueTokens

Se stabilisci che l'utente abbia completato abbastanza sfide per essere autenticato, imposta su `true`. Se l'utente non ha soddisfatto i criteri delle sfide, imposta su `false`.

failAuthentication

Se desideri terminare l'attuale processo di autenticazione, imposta su `true`. Per continuare il processo di autenticazione corrente, imposta su `false`.

Definizione di un esempio di una richiesta di autenticazione

Questo esempio definisce una serie di sfide per l'autenticazione e rilascia token solo se l'utente completa correttamente tutte le sfide. Quando gli utenti completano l'autenticazione SRP con le `PASSWORD_VERIFIER` sfide `SRP_A` e, questa funzione passa loro una risposta `CUSTOM_CHALLENGE` che richiama il trigger create auth challenge. In combinazione con il nostro esempio di sfida di creazione di autenticazione, questa sequenza fornisce una sfida CAPTCHA per la sfida tre e una domanda di sicurezza per la sfida quattro.

Dopo che l'utente ha risolto il CAPTCHA e risposto alla domanda di sicurezza, questa funzione conferma che il pool di utenti può emettere token. L'autenticazione SRP non è richiesta; puoi anche impostare il CAPTCHA e la domanda di sicurezza come sfide uno e due. Nel caso in cui la funzione

define auth challenge non dichiari le sfide SRP, il successo degli utenti è determinato interamente dalle loro risposte ai prompt personalizzati.

Node.js

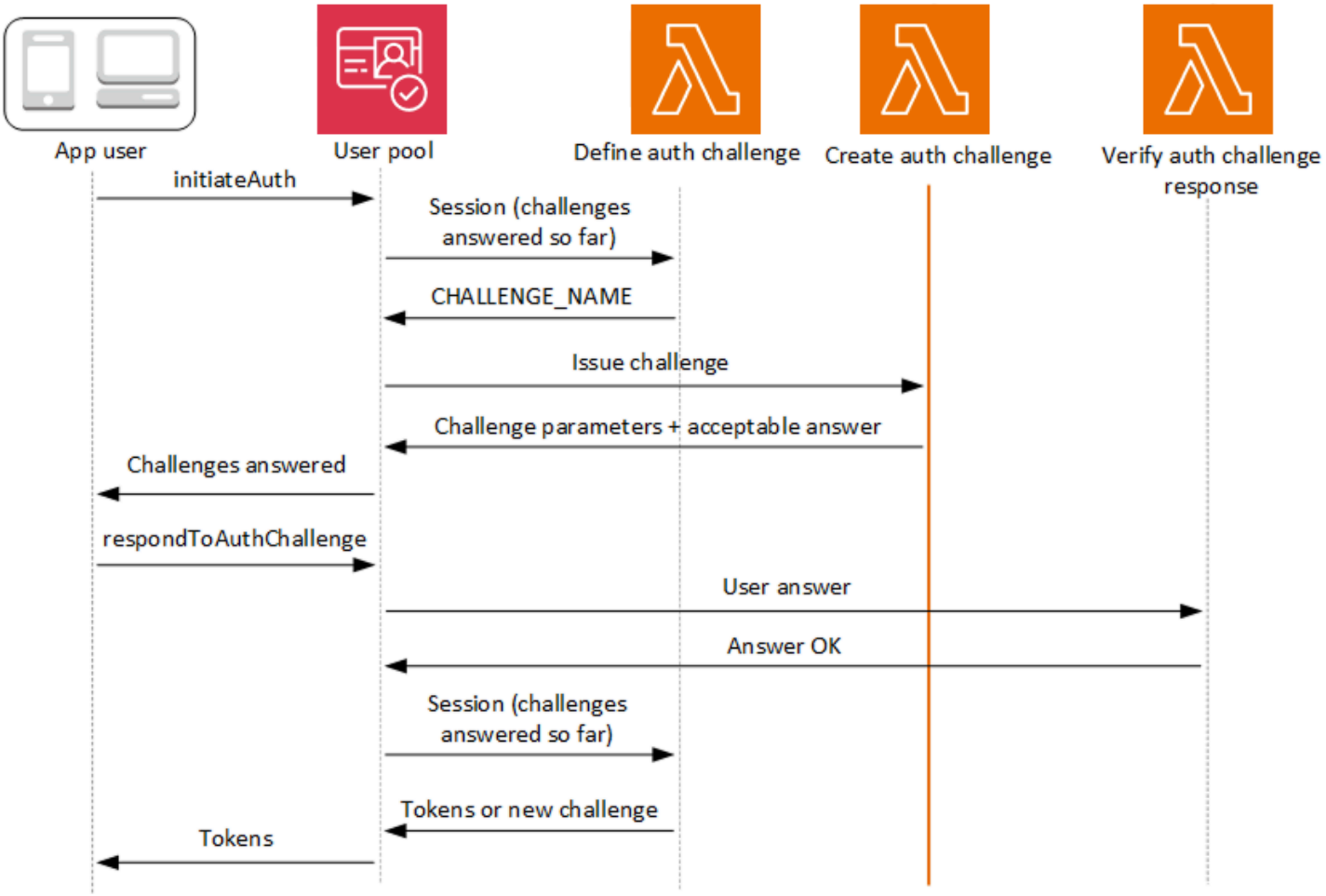
```
const handler = async (event) => {
  if (
    event.request.session.length === 1 &&
    event.request.session[0].challengeName === "SRP_A"
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "PASSWORD_VERIFIER";
  } else if (
    event.request.session.length === 2 &&
    event.request.session[1].challengeName === "PASSWORD_VERIFIER" &&
    event.request.session[1].challengeResult === true
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "CUSTOM_CHALLENGE";
  } else if (
    event.request.session.length === 3 &&
    event.request.session[2].challengeName === "CUSTOM_CHALLENGE" &&
    event.request.session[2].challengeResult === true
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "CUSTOM_CHALLENGE";
  } else if (
    event.request.session.length === 4 &&
    event.request.session[3].challengeName === "CUSTOM_CHALLENGE" &&
    event.request.session[3].challengeResult === true
  ) {
    event.response.issueTokens = true;
    event.response.failAuthentication = false;
  } else {
    event.response.issueTokens = false;
    event.response.failAuthentication = true;
  }

  return event;
};
```

```
export { handler };
```

Creazione del trigger Lambda di una richiesta di autenticazione

Il trigger create auth challenge è una funzione Lambda che contiene i dettagli di ogni sfida dichiarata dal trigger define auth challenge. Elabora il nome della sfida dichiarato dal trigger define auth challenge e restituisce un messaggio `publicChallengeParameters` che l'applicazione deve presentare all'utente. Questa funzione fornisce quindi al pool di utenti la risposta alla sfida`privateChallengeParameters`, che il pool di utenti passa al trigger verify auth challenge. Laddove il trigger define auth challenge gestisce la sequenza della sfida, il trigger create auth challenge gestisce il contenuto della sfida.



Crea una sfida di autenticazione

Amazon Cognito richiama questo trigger dopo la definizione di una richiesta di autenticazione nel caso in cui una richiesta personalizzata sia stata specificata come parte del trigger di definizione di una richiesta di autenticazione. Crea un flusso di autenticazione personalizzato.

Questo trigger Lambda viene richiamato per creare una richiesta da presentare all'utente. La richiesta per questo trigger Lambda include `challengeName` e `session`. Il `challengeName` è una stringa, nonché il nome della prossima sfida per l'utente. Il valore di questo attributo è impostato nel trigger Lambda di definizione di una richiesta di autenticazione.

Il loop di sfide si ripete finché non viene data risposta a tutte le sfide.

Argomenti

- Parametri del trigger Lambda di creazione di una richiesta di autenticazione
- Creazione di un esempio di una richiesta di autenticazione

Parametri del trigger Lambda di creazione di una richiesta di autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "challengeName": "string",
        "session": [
            ChallengeResult,
            . . .
        ],
        "clientMetadata": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
```

```
    },
    "response": {
        "publicChallengeParameters": {
            "string": "string",
            . . .
        },
        "privateChallengeParameters": {
            "string": "string",
            . . .
        },
        "challengeMetadata": "string"
    }
}
```

Parametri di richiesta di creazione sfida di autenticazione

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

utente NotFound

Questo valore booleano viene popolato quando `PreventUserExistenceErrors` è impostato su `ENABLED` per il client del bacino d'utenza.

challengeName

Il nome della nuova sfida.

sessione

L'elemento della sessione è una matrice di elementi `ChallengeResult`, ognuno dei quali contiene i seguenti elementi:

challengeName

Il tipo di sfida. Uno di:`"CUSTOM_CHALLENGE"`, `"PASSWORD_VERIFIER""SMS_MFA","DEVICE_SRP_AUTH","DEVICE_PASSWORD_VERIFIER","NE` o`"ADMIN_NO_SRP_AUTH"`.

challengeResult

Imposta su `true` se l'utente ha completato con successo la sfida o altrimenti su `false`.

challengeMetaData

Il tuo nome per la sfida personalizzata. Usato solo se `challengeName` è `"CUSTOM_CHALLENGE"`.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di creazione di una richiesta di autenticazione. Puoi utilizzare il ClientMetadata parametro nelle azioni AdminRespondToAuthChallengee RespondToAuthChallengeAPI per passare questi dati alla tua funzione Lambda. La richiesta che richiama la funzione create auth challenge non include i dati passati nel ClientMetadata AdminInitiateAuthparametro e le operazioni API. InitiateAuth

Parametri di risposta di creazione sfida di autenticazione

pubblico ChallengeParameters

Una o più coppie chiave-valore per l'app del client da usare nella sfida che deve essere presentata all'utente. Questo parametro deve contenere tutte le informazioni necessarie per presentare, in maniera accurata, la sfida all'utente.

privato ChallengeParameters

Questo parametro viene utilizzato solo dal trigger Lambda di una verifica di risposta di una richiesta di autenticazione. Questo parametro dovrebbe contenere tutte le informazioni necessarie per convalidare la risposta dell'utente alla sfida. In altre parole, il parametro `publicChallengeParameters` contiene la domanda che viene presentata all'utente e `privateChallengeParameters` contiene le risposte valide per la domanda.

challengeMetaData

Il tuo nome per la sfida personalizzata, se questa è una sfida personalizzata.

Creazione di un esempio di una richiesta di autenticazione

Questa funzione ha due sfide personalizzate che corrispondono alla sequenza di sfida nel nostro esempio di challenge define auth. Le prime due sfide sono l'autenticazione SRP. Per la terza sfida, questa funzione restituisce un URL CAPTCHA all'applicazione nella risposta alla sfida. L'applicazione esegue il rendering del CAPTCHA all'URL specificato e restituisce l'input dell'utente. L'URL per

l'immagine CAPTCHA viene aggiunto ai parametri di sfida pubblica come "`captchaUrl`", e la risposta prevista viene aggiunta ai parametri di sfida privata.

Per la quarta sfida, questa funzione restituisce una domanda di sicurezza. L'applicazione esegue il rendering della domanda e richiede all'utente la risposta. Dopo che gli utenti hanno risolto entrambe le sfide personalizzate, il trigger define auth challenge conferma che il pool di utenti può emettere token.

Node.js

```
const handler = async (event) => {
  if (event.request.challengeName !== "CUSTOM_CHALLENGE") {
    return event;
  }

  if (event.request.session.length === 2) {
    event.response.publicChallengeParameters = {};
    event.response.privateChallengeParameters = {};
    event.response.publicChallengeParameters.captchaUrl = "url/123.jpg";
    event.response.privateChallengeParameters.answer = "5";
  }

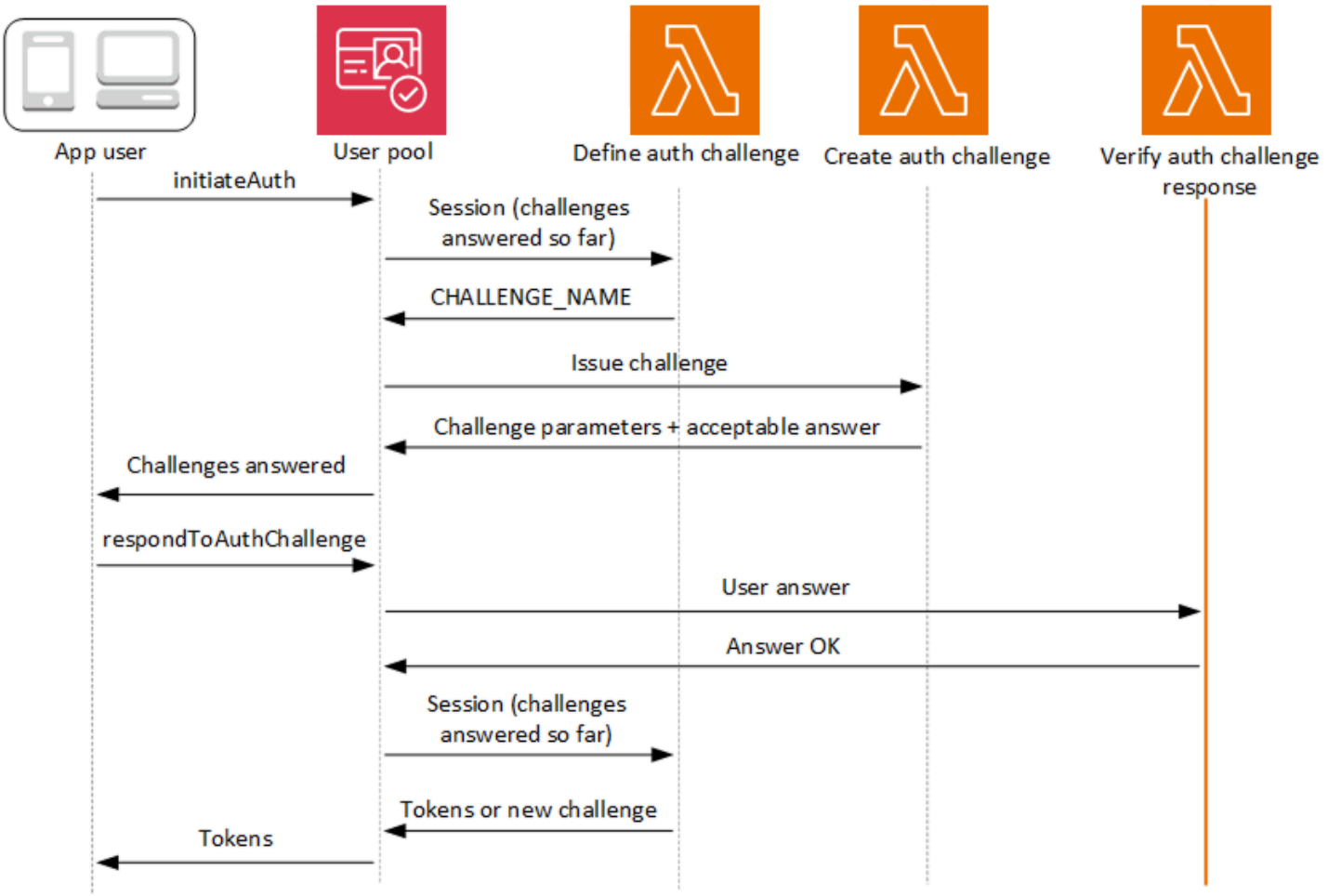
  if (event.request.session.length === 3) {
    event.response.publicChallengeParameters = {};
    event.response.privateChallengeParameters = {};
    event.response.publicChallengeParameters.securityQuestion =
      "Who is your favorite team mascot?";
    event.response.privateChallengeParameters.answer = "Peccy";
  }

  return event;
};

export { handler };
```

Trigger Lambda di una verifica di risposta di una richiesta di autenticazione

Il trigger verify auth challenge è una funzione Lambda che confronta la risposta fornita dall'utente con una risposta nota. Questa funzione indica al gruppo di utenti se l'utente ha risposto correttamente alla sfida. Quando il trigger di verifica dell'autenticazione risponde con un `answerCorrect` of`true`, la sequenza di autenticazione può continuare.



Verifica la risposta alla sfida di autenticazione

> Amazon Cognito richiama questo trigger per verificare se la risposta dell'utente finale per una richiesta di autenticazione personalizzata è valida o meno. Fa parte di un flusso di autenticazione personalizzato di un bacino d'utenza.

La richiesta per questo trigger include il `privateChallengeParameters` e la `challengeAnswer`. Il trigger Lambda di creazione di una richiesta di autenticazione restituisce valori `privateChallengeParameters` e include la risposta prevista fornita dall'utente. Il parametro `challengeAnswer` contiene la risposta dell'utente per la sfida.

La risposta contiene l'attributo `answerCorrect`. Se l'utente completa correttamente la sfida, Amazon Cognito imposta il valore dell'attributo su `true`. Se l'utente non completa correttamente la sfida, Amazon Cognito imposta il valore su `false`.

Il loop di sfide si ripete finché gli utenti non rispondono a tutte le sfide.

Argomenti

- Parametri del trigger Lambda della verifica di una richiesta di autenticazione
- Verifica di un esempio di risposta di una richiesta di autenticazione

Parametri del trigger Lambda della verifica di una richiesta di autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "privateChallengeParameters": {
            "string": "string",
            . . .
        },
        "challengeAnswer": "string",
        "clientMetadata": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {
        "answerCorrect": boolean
    }
}
```

Parametri di richiesta di verifica sfida di autenticazione

userAttributes

Questo parametro contiene una o più coppie nome-valore che rappresentano gli attributi dell'utente.

utente NotFound

Quando Amazon Cognito imposta `PreventUserExistenceErrors` su `ENABLED` per il tuo client del bacino d'utenza, Amazon Cognito popola questo valore booleano.

privato ChallengeParameters

Questo parametro proviene dal trigger Creazione di una richiesta di autenticazione. Per determinare se l'utente ha superato una sfida, Amazon Cognito confronta i parametri con quelli della challengeAnswer di un utente.

Questo parametro contiene tutte le informazioni necessarie per convalidare la risposta dell'utente alla sfida. Tali informazioni includono la domanda che Amazon Cognito pone all'utente (`publicChallengeParameters`) e le risposte valide per la domanda (`privateChallengeParameters`). Questo parametro viene utilizzato solo dal trigger Lambda di verifica della risposta di una richiesta di autenticazione.

challengeAnswer

Il valore di questo parametro è la risposta dell'utente alla sfida.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di verifica di una richiesta di autenticazione. Per passare questi dati alla funzione Lambda, usa il ClientMetadata parametro nelle operazioni AdminRespondToAuthChallengee RespondToAuthChallengeAPI. Amazon Cognito non include i dati del ClientMetadata parametro AdminInitiateAuthe le operazioni InitiateAuthAPI nella richiesta che passa alla funzione verify auth challenge.

Parametri di risposta di verifica sfida di autenticazione

answerCorrect

Se l'utente completa correttamente la sfida, Amazon Cognito imposta questo parametro su `true`. Se l'utente non completa correttamente la sfida, Amazon Cognito imposta il parametro su `false`.

Verifica di un esempio di risposta di una richiesta di autenticazione

Questa funzione di verifica dell'autenticazione verifica se la risposta dell'utente a una sfida corrisponde alla risposta prevista. La risposta dell'utente è definita dall'input dell'applicazione e la risposta preferita è definita dalla `privateChallengeParameters.answer` risposta del trigger di

creazione dell'autenticazione. Sia la risposta corretta che la risposta data fanno parte dell'evento di input di questa funzione.

In questo esempio, se la risposta dell'utente corrisponde alla risposta prevista, Amazon Cognito imposta il `answerCorrect` parametro su. `true`

Node.js

```
const handler = async (event) => {
  if (
    event.request.privateChallengeParameters.answer ===
    event.request.challengeAnswer
  ) {
    event.response.answerCorrect = true;
  } else {
    event.response.answerCorrect = false;
  }

  return event;
};

export { handler };
```

Trigger Lambda di pre-generazione del token

Poiché Amazon Cognito richiama questo trigger prima della generazione di token, puoi personalizzare le richieste di token dei pool di utenti. Con le funzionalità di base dell'evento di attivazione della versione 1 o `V1_0` precedente alla generazione del token, puoi personalizzare il token di identità (ID). Nei pool di utenti con il piano di funzionalità Essentials o Plus, è possibile generare la versione due o l'evento `V2_0` trigger con personalizzazione del token di accesso e la versione tre o l'evento `V3_0` trigger con personalizzazione del token di accesso per l'autorizzazione da macchina a macchina (M2M), comprese le concessioni di credenziali client e il funzionamento dell'API. `GetClientToken`

Amazon Cognito invia un evento `V1_0` come richiesta alla funzione con dati che scriverebbe sul token ID. Un `V3_0` evento `V2_0` or è una singola richiesta con i dati che Amazon Cognito scriverebbe sia sull'identità che sui token di accesso. Per personalizzare entrambi i token, devi aggiornare la funzione per utilizzare la versione due o tre del trigger e inviare i dati per entrambi i token nella stessa risposta.

Amazon Cognito applica le risposte agli eventi della versione due ai token di accesso dall'autenticazione utente, in cui un utente umano ha presentato le credenziali al tuo pool di utenti. Le risposte agli eventi della versione tre si applicano ai token di accesso derivanti dall'autenticazione utente e dall'autenticazione automatica, in cui i sistemi automatici autorizzano le richieste di token di accesso con i segreti del client dell'app. A parte le circostanze dei token di accesso risultanti, gli eventi della versione due e tre sono identici.

Questo trigger Lambda può aggiungere, rimuovere e modificare alcune richieste nei token di identità e accesso prima che Amazon Cognito le invii all'app. Per usare questa funzionalità, puoi associare una funzione Lambda dalla console del pool di utenti di Amazon Cognito o aggiornare il pool di utenti `LambdaConfig` tramite la AWS Command Line Interface (AWS CLI).

Versioni degli eventi

Il tuo pool di utenti può fornire diverse versioni di un evento di attivazione precedente alla generazione di token alla tua funzione Lambda. Un `V1_0` trigger fornisce i parametri per la modifica dei token ID. Un `V3_0` trigger `V2_0` or fornisce i parametri per quanto segue.

1. Le funzioni di un `V1_0` trigger.
2. La possibilità di personalizzare i token di accesso.
3. La possibilità di passare tipi di dati complessi a ID e accedere ai valori delle richieste di token:
   - Stringa
   - Numero
   - Booleano
   - Matrice di stringhe, numeri, valori booleani o una combinazione di questi
   - JSON

> **Note**
>
> Nel token ID, è possibile compilare oggetti complessi in base ai valori delle attestazioni ad eccezione di`phone_number_verified`,, `email_verified` e. `updated_at` `address`

I pool di utenti forniscono `V1_0` eventi per impostazione predefinita. Per configurare il tuo pool di utenti per inviare un `V2_0` evento, scegli una versione evento Trigger delle funzionalità di base + personalizzazione del token di accesso per le identità degli utenti quando configuri il trigger nella

console Amazon Cognito. Per creare `V3_0` eventi, scegli Funzionalità di base + personalizzazione dei token di accesso per le identità di utenti e macchine. Puoi anche impostare il valore di `LambdaVersion` nei LambdaConfigparametri in una richiesta UpdateUserPool o CreateUserPool API. Le versioni degli eventi uno, due e tre sono disponibili nei piani di funzionalità Essentials e Plus. Le operazioni M2M per gli eventi della versione tre hanno una struttura tariffaria separata dalla formula degli utenti attivi mensili (MAU). Per ulteriori informazioni, consultare Prezzi di Amazon Cognito.

> **Note**
>
> I pool di utenti che erano operativi con l'opzione Funzioni di sicurezza avanzate prima del 22 novembre 2024 alle 18:00 GMT e che rimangono nel livello di funzionalità Lite hanno accesso alle versioni 1 e 2 degli eventi del trigger precedente alla generazione di token. I pool di utenti di questo livello legacy senza funzionalità di sicurezza avanzate hanno accesso alla versione 1 dell'evento. La versione tre è disponibile solo in Essentials e Plus.

Riferimento a rivendicazioni e ambiti

Amazon Cognito limita le attestazioni e gli ambiti che puoi aggiungere, modificare o sopprimere nei token di accesso e identità. La tabella seguente descrive le attestazioni che la funzione Lambda può e non può modificare e i parametri degli eventi di attivazione che influiscono sulla presenza o sul valore dell'attestazione.

Richiedi	Tipo di token predefi to	Può aggiun e?	Può modific e?	Può soppri e?	Parametro dell'even to: aggiungi o modifica	Parametro dell'even to: sopprimi	Tipo di identità	Versione dell'evento
Qualsiasi reclamo non incluso nello schema del token	Nessur	Sì	Sì	N/A	`claimsTo dd0r0ver ide`	`claimsTo uppress`	Utente, macchina [1]	Tutto [2]

Richiedi	Tipo di token predefi to	Può aggiun e?	Può modific e?	Può soppri e?	Parametro dell'even to: aggiungi o modifica	Parametro dell'even to: sopprimi	Tipo di identità	Versione dell'evento
del pool di utenti								
`scope`	Access	Sì	Sì	Sì	`scopesTo dd`	`scopesTo uppress`	Utente, macchina [1]	`v2_0`, `v3_0`
`cognito: roups`	ID, access	Sì	Sì	Sì	`groupsTo verride`	`claimsTo uppress`	Utente	Tutti [2]
`cognito: referred_ role`	ID	Sì	Sì	Sì	`preferre Role`	`claimsTo uppress` [3]	Utente	Tutti
`cognito: oles`	ID	Sì	Sì	Sì	`iamRoles oOverrid`	`claimsTo uppress` [3]	Utente	Tutti
`cognito: sername`	ID	No	No	No	N/A	N/A	Utente	N/A
Qualsiasi altra affermazi one con un `cognito:` prefisso	Nessur	No	No	No	N/A	N/A	N/A	N/A
`username`	Access	No	No	No	N/A	N/A	Utente	`v2_0`, `v3_0`

Richiedi	Tipo di token predefi to	Può aggiun e?	Può modific e?	Può soppri e?	Parametro dell'even to: aggiungi o modifica	Parametro dell'even to: sopprimi	Tipo di identità	Versione dell'evento
`sub`	ID, access	No	No	No	N/A	N/A	Utente	N/A
attributo OIDC standard	ID	Sì	Sì	Sì	`claimsTo dd0r0ver ide`	`claimsTo uppress`	Utente	Tutti
Attributo `custom:`	ID	Sì	Sì	Sì	`claimsTo dd0r0ver ide`	`claimsTo uppress`	Utente	Tutti
Attributo `dev:`	ID	No	No	Sì	N/A	`claimsTo uppress`	Utente	Tutti
`identiti s`	ID	No	No	No	N/A	N/A	Utente	N/A
`aud`[4]	ID	No	No	No	N/A	N/A	Utente, macchina	N/A
`client_i`	Access	No	No	No	N/A	N/A	Utente, macchina	N/A
`event_id`	Access	No	No	No	N/A	N/A	Utente, macchina	N/A
`device_k y`	Access	No	No	No	N/A	N/A	Utente	N/A
`version`	Access	No	No	No	N/A	N/A	Utente, macchina	N/A

Richiedi	Tipo di token predefinito	Può aggiunge?	Può modificare?	Può sopprimere?	Parametro dell'evento: aggiungi o modifica	Parametro dell'evento: sopprimi	Tipo di identità	Versione dell'evento
`acr`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`amr`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`at_hash`	ID	No	No	No	N/A	N/A	Utente, macchina	N/A
`auth_time`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`azp`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`exp`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`iat`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`iss`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`jti`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`nbf`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`nonce`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A

Richiedi	Tipo di token predefinito	Può aggiungere?	Può modificare?	Può sopprimere?	Parametro dell'evento: aggiungi o modifica	Parametro dell'evento: sopprimi	Tipo di identità	Versione dell'evento
`origin_ji`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A
`token_use`	ID, access	No	No	No	N/A	N/A	Utente, macchina	N/A

[1] I token di accesso per le identità delle macchine sono disponibili solo con l'evento `v3_0` di input del trigger. La versione 3 dell'evento è disponibile solo nei livelli di funzionalità Essentials e Plus. I pool di utenti del livello Lite possono ricevere `v1_0` eventi. I pool di utenti del livello Lite con funzionalità di sicurezza avanzate possono ricevere `v1_0` `v2_0` eventi.

[2] Configura il trigger precedente alla generazione del token sulla versione evento solo `v1_0` per il token ID, `v2_0` per ID e token di accesso, per ID e token di accesso, `v3_0` per ID e token di accesso con funzionalità per le identità delle macchine.

[3] Per eliminare le `cognito:roles` affermazioni `cognito:preferred_role` and, aggiungi `cognito:groups` a. `claimsToSuppress`

[4] Puoi aggiungere un'`aud`affermazione ai token di accesso, ma il suo valore deve corrispondere all'ID client dell'app della sessione corrente. Puoi derivare l'ID del client nell'evento di richiesta da `event.callerContext.clientId`.

Personalizzazione del token di identità

Con tutte le versioni evento del trigger Lambda precedente alla generazione del token, puoi personalizzare il contenuto di un token di identità (ID) dal tuo pool di utenti. Il token ID fornisce gli attributi utente da un'origine di identità affidabile per l'accesso a un'app web o per dispositivi mobili. Per ulteriori informazioni sui token ID, consulta Comprendere il token di identità (ID).

Gli usi del trigger Lambda di pre-generazione di token con un token ID includono quanto segue.

- Apportare una modifica in fase di runtime al ruolo IAM richiesto dall'utente da un pool di identità.

- Aggiungere gli attributi utente da un'origine esterna.
- Aggiungere o sostituire i valori degli attributi utente esistenti.
- Sopprimere la divulgazione degli attributi utente che, a causa degli ambiti autorizzati dell'utente e dell'accesso in lettura agli attributi concesso al client dell'app, verrebbero altrimenti trasferiti all'app.

Personalizzazione del token di accesso

Con le versioni evento due e tre del trigger Lambda precedente alla generazione del token, puoi personalizzare il contenuto di un token di accesso dal tuo pool di utenti. Il token di accesso autorizza gli utenti a recuperare informazioni da risorse ad accesso protetto come operazioni API autorizzate da token Amazon Cognito e API di terze parti. Per l'autorizzazione da macchina a macchina (M2M), indipendentemente dal fatto che il token di accesso provenga dalla concessione di credenziali del client o dall'operazione API`GetClientToken`, Amazon Cognito richiama il trigger precedente alla generazione del token solo quando il pool di utenti è configurato per un evento versione tre (). `V3_0` Per ulteriori informazioni sui token di accesso, consulta Comprendere il token di accesso.

Gli usi del trigger Lambda di pre-generazione di token con un token di accesso includono i seguenti.

- Aggiungi o sopprimi gli ambiti nel claim. `scope` Ad esempio, puoi aggiungere ambiti a un token di accesso ottenuto dall'autenticazione API dei pool di utenti di Amazon Cognito, che assegna solo l'ambito `aws.cognito.signin.user.admin`.
- Modificare l'appartenenza di un utente ai gruppi di pool di utenti.
- Aggiunta di richieste non ancora presenti in un token di accesso Amazon Cognito.
- Bloccare la divulgazione di attestazioni che altrimenti verrebbero trasmesse all'app.

Per supportare la personalizzazione dell'accesso nel pool di utenti, è necessario configurare il pool di utenti per generare una versione aggiornata della richiesta di trigger. Aggiornare il pool di utenti come illustrato nella procedura seguente.

Console di gestione AWS

Per supportare la personalizzazione del token di accesso in un trigger Lambda di pre-generazione di token

1. Passa alla console Amazon Cognito e scegli User Pools (Bacini d'utenza).
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.

3. Scegli il menu Estensioni e individua i trigger Lambda.
4. Aggiungi o modifica un Trigger di pre-generazione di token.
5. Scegli una funzione Lambda in Assegna la funzione Lambda.
6. Scegliete una versione evento Trigger di Funzionalità di base + personalizzazione del token di accesso per le identità utente o Funzionalità di base + personalizzazione del token di accesso per le identità di utenti e macchine. Questa impostazione aggiorna i parametri di richiesta inviati da Amazon Cognito alla funzione per includere campi per la personalizzazione del token di accesso.

User pools API

Per supportare la personalizzazione del token di accesso in un trigger Lambda di pre-generazione di token

Genera una CreateUserPoolrichiesta o API. UpdateUserPool È necessario specificare un valore per tutti i parametri che non si desidera impostare su un valore predefinito. Per ulteriori informazioni, consulta Aggiornamento della configurazione del pool di utenti e del client dell'app.

Includi il seguente contenuto nel parametro `LambdaVersion` della richiesta. Un `LambdaVersion` valore di `V2_0` fa sì che il pool di utenti aggiunga parametri e applichi modifiche ai token di accesso. Un `LambdaVersion` valore di `V3_0` produce lo stesso evento`V2_0`, ma fa sì che il pool di utenti applichi le modifiche anche ai token di accesso M2M. Per richiamare una versione di funzione specifica, usa un ARN della funzione Lambda on una versione della funzione come valore di `LambdaArn`.

```
"PreTokenGenerationConfig": {
   "LambdaArn": "arn:aws:lambda:us-west-2:123456789012:function:MyFunction",
   "LambdaVersion": "V3_0"
},
```

Metadati client per le credenziali del client machine-to-machine (M2M)

È possibile trasmettere i metadati dei client nelle richieste M2M. I metadati del client sono informazioni aggiuntive provenienti da un ambiente utente o applicativo che possono contribuire ai risultati di un. Trigger Lambda di pre-generazione del token Nelle operazioni di autenticazione con un user principal, è possibile passare i metadati del client al trigger precedente alla generazione del token nel corpo delle richieste API AdminRespondToAuthChallenge. RespondToAuthChallenge

Poiché le applicazioni gestiscono il flusso per la generazione di token di accesso per M2M con richieste dirette aEndpoint Token, hanno un modello diverso. Nel corpo POST delle richieste di token per le credenziali del client, passate un `aws_client_metadata` parametro con l'oggetto di metadati del client URL-encoded () `x-www-form-urlencoded` a string. Per una richiesta di esempio, consulta Credenziali del client con autorizzazione di base. Di seguito è riportato un parametro di esempio che passa le coppie chiave-valore. `{"environment": "dev", "language": "en-US"}`

```
aws_client_metadata=%7B%22environment%22%3A%20%22dev%22,%20%22language%22%3A%20%22en-US%22%7D
```

Altre risorse

- Come personalizzare i token di accesso nei pool di utenti di Amazon Cognito

Argomenti

- Origini del trigger Lambda di pre-generazione del token
- Parametri del trigger Lambda di pre-generazione del token
- Esempio della versione due dell'evento trigger pre-token: aggiunta ed eliminazione di richieste, ambiti e gruppi
- Evento precedente alla generazione di token (versione due): aggiungere attestazioni con oggetti complessi
- Esempio di evento versione 1 di generazione del pre-token: aggiunta di una nuova richiesta ed eliminazione di una richiesta esistente
- Esempio di evento versione 1 di generazione del pre-token: modifica dell'appartenenza al gruppo dell'utente

Origini del trigger Lambda di pre-generazione del token

Valore triggerSource	Event
`TokenGeneration_HostedAuth`	Chiamato durante l'autenticazione dalla pagina di accesso gestito di Amazon Cognito.

Valore triggerSource	Event
`TokenGeneration_Authentication`	Chiamato dopo che i flussi di autenticazione degli utenti sono terminati.
`TokenGeneration_NewPasswordChallenge`	Chiamato dopo che l'utente viene creato da un amministratore. Questo flusso viene chiamato quando l'utente deve cambiare una password temporanea.
`TokenGeneration_ClientCredentials`	Chiamato dopo la concessione delle credenziali del client M2M. Il tuo pool di utenti invia questo evento solo quando la versione dell'evento è. `V3_0`
`TokenGeneration_AuthenticateDevice`	Chiamato alla fine dell'autenticazione di un dispositivo dell'utente.
`TokenGeneration_RefreshTokens`	Chiamato quando un utente cerca di aggiornare l'identità e i token di accesso.

Parametri del trigger Lambda di pre-generazione del token

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste. Quando aggiungi un trigger Lambda di pre-generazione di token al pool di utenti, puoi scegliere una versione del trigger. Questa versione determina se Amazon Cognito passa una richiesta alla funzione Lambda con parametri aggiuntivi per la personalizzazione del token di accesso.

Version one

Il token della versione uno può impostare l'appartenenza al gruppo, i ruoli IAM e nuove attestazioni nei token ID. Le eccezioni relative all'appartenenza al gruppo si applicano anche alla `cognito:groups` dichiarazione contenuta nei token di accesso.

```
{
    "request": {
        "userAttributes": {"string": "string"},
        "groupConfiguration": {
```

```
                "groupsToOverride": [
                    "string",
                    "string"
                ],
                "iamRolesToOverride": [
                    "string",
                    "string"
                ],
                "preferredRole": "string"
            },
            "clientMetadata": {"string": "string"}
        },
        "response": {
            "claimsOverrideDetails": {
                "claimsToAddOrOverride": {"string": "string"},
                "claimsToSuppress": [
                    "string",
                    "string"
                ],
                "groupOverrideDetails": {
                    "groupsToOverride": [
                        "string",
                        "string"
                    ],
                    "iamRolesToOverride": [
                        "string",
                        "string"
                    ],
                    "preferredRole": "string"
                }
            }
        }
}
```

Versions two and three

Le versioni due e tre richiedono eventi che aggiungono campi che personalizzano il token di accesso. I pool di utenti applicano le modifiche dagli eventi della versione tre ai token di accesso per le identità delle macchine. Queste versioni aggiungono anche il supporto per tipi di `claimsToOverride` dati complessi nell'oggetto di risposta. La tua funzione Lambda può restituire i seguenti tipi di dati nel valore di: `claimsToOverride`

- Stringa

- Numero
- Booleano
- Matrice di stringhe, numeri, valori booleani o una combinazione di questi
- JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string"
        },
        "scopes": ["string", "string"],
        "groupConfiguration": {
            "groupsToOverride": ["string", "string"],
            "iamRolesToOverride": ["string", "string"],
            "preferredRole": "string"
        },
        "clientMetadata": {
            "string": "string"
        }
    },
    "response": {
        "claimsAndScopeOverrideDetails": {
            "idTokenGeneration": {
                "claimsToAddOrOverride": {
                    "string": [accepted datatype]
                },
                "claimsToSuppress": ["string", "string"]
            },
            "accessTokenGeneration": {
                "claimsToAddOrOverride": {
                    "string": [accepted datatype]
                },
                "claimsToSuppress": ["string", "string"],
                "scopesToAdd": ["string", "string"],
                "scopesToSuppress": ["string", "string"]
            },
            "groupOverrideDetails": {
                "groupsToOverride": ["string", "string"],
                "iamRolesToOverride": ["string", "string"],
                "preferredRole": "string"
            }
```

```
        }
    }
}
```

Parametri di richiesta di pre-generazione del token

Nome	Description	Versione minima dell'evento di trigger
userAttributes	Attributi del profilo utente nel pool di utenti.	1
groupConfiguration	L'oggetto di input contenente l'attuale configurazione del gruppo. L'oggetto include `groupsToOverride` , `iamRolesToOverride` `epreferredRole` .	1
gruppi ToOverride	I gruppi di pool di utenti di cui l'utente fa parte.	1
sono RolesToOverride	È possibile associare un gruppo di pool di utenti a un ruolo AWS Identity and Access Management (IAM). Questo elemento è un elenco di tutti i ruoli IAM dei gruppi di cui l'utente è membro.	1
preferredRole	Puoi impostare una precedenza per i gruppi di pool di utenti. Questo elemento contiene il nome del ruolo IAM del gruppo con la precedenza più alta nell'elemento `groupsToOverride` .	1
clientMetadata	Una o più coppie chiave-valore che è possibile specificare e fornire come input personalizzato alla funzione Lambda per il trigger di pre-generazione del token. Per passare questi dati alla funzione Lambda, usa il ClientMetadata parametro nelle operazioni AdminRespondToAuthChallengee RespondToAuthChallengeAPI. Amazon Cognito non include i dati del `ClientMetadata` parametro AdminInitiateAuthe le operazioni	1

Nome	Description	Versione minima dell'evento di trigger
	InitiateAuthAPI nella richiesta che passa alla funzione di generazione precedente al token.	
ambiti	Ambiti dei token di accesso. Gli ambiti presenti in un token di accesso sono gli ambiti standard e personalizzati del pool di utenti richiesti dall'utente e che il client dell'app è autorizzato a emettere.	2

Parametri di risposta di pre-generazione del token

Nome	Description	Versione minima dell'evento di trigger
rivendicazioni OverrideDetails	Un container per tutti gli elementi di un evento trigger `V1_0`.	1
reclami AndScopeO verrideDetails	Un contenitore per tutti gli elementi di un evento `V2_0` o `V3_0` trigger.	2
id TokenGene ration	Le attestazioni che desideri sostituire, aggiungere o sopprimere nel token di identità dell'utente. Questi valori di personalizzazione del token da genitore a ID vengono visualizzati solo nella versione 2 e successive dell'even to, ma gli elementi secondari vengono visualizzati negli eventi della versione 1.	2
accesso TokenGeneration	Le attestazioni e gli ambiti che desideri sostituire, aggiungere o sopprimere nel token di accesso dell'uten te. Questo elemento principale per accedere ai valori di personalizzazione dei token appare solo nella versione 2 e successive dell'evento.	2

Nome	Description	Versione minima dell'evento di trigger
rivendicazioni ToAddOrOverride	Una mappa di una o più attestazioni e dei relativi valori che desideri aggiungere o modificare. Per attestazioni relative ai gruppi, utilizza invece `groupOverrideDetails` . Nel caso della versione 2 e successive, questo elemento appare sotto entrambe le versioni `accessTokenGeneration` `eidTokenGeneration` .	1. * -
reclami ToSuppress	Un elenco delle attestazioni che desideri vengano soppresse da Amazon Cognito. Se la tua funzione sopprime e sostituisce un valore di attestazione, Amazon Cognito sopprime l'attestazione. Nel caso della versione 2 e successive, questo elemento appare sotto entrambe le versioni `accessTokenGeneration` `eidTokenGeneration` .	1

Nome	Description	Versione minima dell'evento di trigger
gruppo OverrideDetails	L'oggetto di output contenente l'attuale configurazione di gruppo. L'oggetto include `groupsToOverride` , `iamRolesToOverride` e `preferredRole` . La funzione sostituisce l'oggetto `groupOverrideDetails` con l'oggetto fornito. Se fornisci un oggetto vuoto o nullo nella risposta, Amazon Cognito sopprime i gruppi. Per mantenere la configurazione di un gruppo esistente così com'è, copia il valore dell'oggetto `groupConfiguration` della richiesta nell'oggetto `groupOverrideDetails` della risposta. Quindi ritrasmettilo al servizio. Amazon Cognito ID e i token di accesso contengono entrambi i token reclamo `cognito:groups` . L'oggetto `groupOverrideDetails` sostituirà l'attestazione `cognito:groups` nei token di accesso e nei token ID. Le sostituzioni di gruppo sono le uniche modifiche al token di accesso che gli eventi della versione 1 possono apportare.	1
ambiti ToAdd	Un elenco di ambiti che desideri aggiungere all'`scope`attestazione nel token di accesso dell'utente. Non puoi aggiungere valori di ambito che contengono uno o più caratteri con spazi vuoti.	2
ambiti ToSuppress	Un elenco di ambiti che desideri rimuovere dall'`scope`attestazione nel token di accesso dell'utente.	2

* Gli oggetti di risposta agli eventi della versione uno possono restituire stringhe. Gli oggetti di risposta agli eventi della versione due e tre possono restituire oggetti complessi.

Esempio della versione due dell'evento trigger pre-token: aggiunta ed eliminazione di richieste, ambiti e gruppi

Questo esempio apporta le seguenti modifiche ai token di un utente.

1. Imposta i `family_name` valori come `Doe` nel token ID.
2. Impedisce la visualizzazione delle richieste `email` e `phone_number` nel token ID.
3. Imposta la richiesta `cognito:roles` del token ID su `"arn:aws:iam::123456789012:role\/sns_callerA","arn:aws:iam::123456789012:role\/sns_callerC","arn:aws:iam::123456789012:role\/sns_callerB"`.
4. Imposta la richiesta `cognito:preferred_role` del token ID su `arn:aws:iam::123456789012:role/sns_caller`.
5. Aggiunge gli ambiti `openid`, `email` e `solar-system-data/asteroids.add` al token di accesso.
6. Elimina l'ambito `phone_number` e `aws.cognito.signin.user.admin` dal token di accesso. La rimozione di `phone_number` impedisce il recupero del numero di telefono dell'utente da `userInfo`. La rimozione di `aws.cognito.signin.user.admin` impedisce alle richieste API da parte dell'utente di leggere e modificare il proprio profilo con l'API dei pool di utenti di Amazon Cognito.

   > **Note**
   >
   > La rimozione di `phone_number` dagli ambiti impedisce solo il recupero del numero di telefono di un utente se gli ambiti rimanenti nel token di accesso includono `openid` e almeno un altro ambito standard. Per ulteriori informazioni, consulta Informazioni sugli ambiti.

7. Imposta la richiesta `cognito:groups` del token ID e di accesso su `"new-group-A","new-group-B","new-group-C"`.

JavaScript

```
export const handler = function(event, context) {
  event.response = {
    "claimsAndScopeOverrideDetails": {
      "idTokenGeneration": {
        "claimsToAddOrOverride": {
```

```
          "family_name": "Doe"
        },
        "claimsToSuppress": [
          "email",
          "phone_number"
        ]
      },
      "accessTokenGeneration": {
        "scopesToAdd": [
          "openid",
          "email",
          "solar-system-data/asteroids.add"
        ],
        "scopesToSuppress": [
          "phone_number",
          "aws.cognito.signin.user.admin"
        ]
      },
      "groupOverrideDetails": {
        "groupsToOverride": [
          "new-group-A",
          "new-group-B",
          "new-group-C"
        ],
        "iamRolesToOverride": [
          "arn:aws:iam::123456789012:role/new_roleA",
          "arn:aws:iam::123456789012:role/new_roleB",
          "arn:aws:iam::123456789012:role/new_roleC"
        ],
        "preferredRole": "arn:aws:iam::123456789012:role/new_role"
      }
    }
  };
  // Return to Amazon Cognito
  context.done(null, event);
};
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "version": "2",
    "triggerSource": "TokenGeneration_Authentication",
    "region": "us-east-1",
    "userPoolId": "us-east-1_EXAMPLE",
    "userName": "JaneDoe",
    "callerContext": {
        "awsSdkVersion": "aws-sdk-unknown-unknown",
        "clientId": "1example23456789"
    },
    "request": {
        "userAttributes": {
            "sub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "cognito:user_status": "CONFIRMED",
            "email_verified": "true",
            "phone_number_verified": "true",
            "phone_number": "+12065551212",
            "family_name": "Zoe",
            "email": "Jane.Doe@example.com"
        },
        "groupConfiguration": {
            "groupsToOverride": ["group-1", "group-2", "group-3"],
            "iamRolesToOverride": ["arn:aws:iam::123456789012:role/sns_caller1",
 "arn:aws:iam::123456789012:role/sns_caller2", "arn:aws:iam::123456789012:role/
sns_caller3"],
            "preferredRole": ["arn:aws:iam::123456789012:role/sns_caller"]
        },
        "scopes": [
            "aws.cognito.signin.user.admin", "openid", "email", "phone"
        ]
    },
    "response": {
        "claimsAndScopeOverrideDetails": []
    }
}
```

Evento precedente alla generazione di token (versione due): aggiungere attestazioni con oggetti complessi

Questo esempio apporta le seguenti modifiche ai token di un utente.

1. Aggiunge affermazioni di tipo number, string, booleano e JSON al token ID. Questa è l'unica modifica che gli eventi di attivazione della versione due rendono disponibile al token ID.
2. Aggiunge rivendicazioni di tipo number, string, boolean e JSON al token di accesso.
3. Aggiunge tre ambiti al token di accesso.
4. Elimina l'`email`affermazione nell'ID e nei token di accesso.
5. Elimina l'`aws.cognito.signin.user.admin`ambito nel token di accesso.

JavaScript

```
export const handler = function(event, context) {

    var scopes = ["MyAPI.read", "MyAPI.write", "MyAPI.admin"]
    var claims = {}
    claims["aud"]= event.callerContext.clientId;
    claims["booleanTest"] = false;
    claims["longTest"] = 9223372036854775807;
    claims["exponentTest"] = 1.7976931348623157E308;
    claims["ArrayTest"] = ["test", 9223372036854775807, 1.7976931348623157E308,
 true];
    claims["longStringTest"] = "\{\
        \"first_json_block\": \{\
            \"key_A\": \"value_A\",\
            \"key_B\": \"value_B\"\
        \},\
        \"second_json_block\": \{\
            \"key_C\": \{\
                \"subkey_D\": [\
                    \"value_D\",\
                    \"value_E\"\
                ],\
                \"subkey_F\": \"value_F\"\
            \},\
            \"key_G\": \"value_G\"\
        \}\
    \}";
    claims["jsonTest"] = {
     "first_json_block": {
      "key_A": "value_A",
      "key_B": "value_B"
     },
     "second_json_block": {
```

```
      "key_C": {
       "subkey_D": [
        "value_D",
        "value_E"
       ],
       "subkey_F": "value_F"
      },
      "key_G": "value_G"
     }
    };
    event.response = {
        "claimsAndScopeOverrideDetails": {
            "idTokenGeneration": {
                "claimsToAddOrOverride": claims,
                "claimsToSuppress": ["email"]
            },
            "accessTokenGeneration": {
                "claimsToAddOrOverride": claims,
                "claimsToSuppress": ["email"],
                "scopesToAdd": scopes,
                "scopesToSuppress": ["aws.cognito.signin.user.admin"]
            }
        }
    };
    console.info("EVENT response\n" + JSON.stringify(event, (_, v) => typeof v === 
 'bigint' ? v.toString() : v, 2))
    console.info("EVENT response size\n" + JSON.stringify(event, (_, v) => typeof v
 === 'bigint' ? v.toString() : v).length)
    // Return to Amazon Cognito
    context.done(null, event);
};
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "version": "2",
    "triggerSource": "TokenGeneration_HostedAuth",
```

```
    "region": "us-west-2",
    "userPoolId": "us-west-2_EXAMPLE",
    "userName": "JaneDoe",
    "callerContext": {
        "awsSdkVersion": "aws-sdk-unknown-unknown",
        "clientId": "1example23456789"
    },
    "request": {
        "userAttributes": {
            "sub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "cognito:user_status": "CONFIRMED",
            "email_verified": "true",
            "phone_number_verified": "true",
            "phone_number": "+12065551212",
            "email": "Jane.Doe@example.com"
        },
        "groupConfiguration": {
            "groupsToOverride": ["group-1", "group-2", "group-3"],
            "iamRolesToOverride": ["arn:aws:iam::123456789012:role/sns_caller1"],
            "preferredRole": ["arn:aws:iam::123456789012:role/sns_caller1"]
        },
        "scopes": [
            "aws.cognito.signin.user.admin",
            "phone",
            "openid",
            "profile",
            "email"
        ]
    },
    "response": {
        "claimsAndScopeOverrideDetails": []
    }
}
```

Esempio di evento versione 1 di generazione del pre-token: aggiunta di una nuova richiesta ed eliminazione di una richiesta esistente

Questo esempio usa un evento trigger versione 1 con una funzione Lambda di pre-generazione di token per aggiungere una nuova attestazione ed eliminarne una esistente.

Node.js

```
const handler = async (event) => {
  event.response = {
    claimsOverrideDetails: {
      claimsToAddOrOverride: {
        my_first_attribute: "first_value",
        my_second_attribute: "second_value",
      },
      claimsToSuppress: ["email"],
    },
  };

  return event;
};

export { handler };
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per il codice di esempio. Dal momento che il codice di esempio non elabora i parametri di richiesta, puoi utilizzare un evento di test con una richiesta vuota. Per ulteriori informazioni sui parametri di richiesta comuni, consulta Evento trigger Lambda per il bacino d'utenza.

JSON

```
{
  "request": {},
  "response": {}
}
```

Esempio di evento versione 1 di generazione del pre-token: modifica dell'appartenenza al gruppo dell'utente

Questo esempio usa un evento trigger versione 1 con una funzione Lambda di pre-generazione di token per modificare l'appartenenza ai gruppi dell'utente.

Node.js

```
const handler = async (event) => {
  event.response = {
    claimsOverrideDetails: {
      groupOverrideDetails: {
        groupsToOverride: ["group-A", "group-B", "group-C"],
        iamRolesToOverride: [
          "arn:aws:iam::XXXXXXXXXXX:role/sns_callerA",
          "arn:aws:iam::XXXXXXXXX:role/sns_callerB",
          "arn:aws:iam::XXXXXXXXXX:role/sns_callerC",
        ],
        preferredRole: "arn:aws:iam::XXXXXXXXXXX:role/sns_caller",
      },
    },
  };

  return event;
};

export { handler };
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
  "request": {},
  "response": {}
}
```

Trigger Lambda di migrazione utenti

Amazon Cognito richiama questo trigger quando un utente non esiste nel bacino d'utenza al momento dell'accesso con una password oppure nel flusso di reimpostazione della password. Dopo l'esito positivo della funzione Lambda, Amazon Cognito crea l'utente nel bacino d'utenza. Per i

dettagli sul flusso di autenticazione con il trigger Lambda di migrazione utenti, consulta Importazione di utenti con un trigger Lambda per la migrazione di utenti.

Usa questo trigger Lambda per migrare gli utenti dalla directory degli utenti esistente nel bacino d'utenza ad Amazon Cognito al momento dell'accesso o durante il flusso di reimpostazione della password.

Argomenti

- Origini dei trigger Lambda di migrazione utenti
- Parametri del trigger Lambda di migrazione utenti
- Esempio di migrazione di un utente con una password esistente
- Risoluzione dei problemi relativi al trigger di migrazione utente

Origini dei trigger Lambda di migrazione utenti

Valore triggerSource	Event
`UserMigration_Authentication` [1]	Migrazione degli utenti al momento dell'accesso.
`UserMigration_ForgotPassword`	Migrazione degli utenti durante il flusso di password dimenticata.

[1] Amazon Cognito non richiama questo trigger quando gli utenti si autenticano con l'accesso senza password.

Parametri del trigger Lambda di migrazione utenti

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "userName": "string",
    "request": {
        "password": "string",
        "validationData": {
```

```
            "string": "string",
            . . .
        },
        "clientMetadata": {
            "string": "string",
       . . .
        }
    },
    "response": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "finalUserStatus": "string",
        "messageAction": "string",
        "desiredDeliveryMediums": [ "string", . . .],
        "forceAliasCreation": boolean,
        "enableSMSMFA": boolean
    }
}
```

Parametri di richiesta di migrazione utenti

userName

Il nome utente inserito dall'utente al momento dell'accesso.

password

La password inserita dall'utente al momento dell'accesso. Amazon Cognito non invia questo valore in una richiesta avviata da un flusso di reimpostazione della password.

validationData

Una o più coppie chiave-valore contenenti i dati di convalida nella richiesta di registrazione dell'utente. Per passare questi dati alla funzione Lambda, puoi utilizzare il ClientMetadata parametro nelle azioni InitiateAuthe AdminInitiateAuthAPI.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda per il trigger di migrazione utenti. Per passare questi dati alla funzione Lambda, puoi utilizzare il ClientMetadata parametro nelle azioni AdminRespondToAuthChallengee ForgotPasswordAPI.

Parametri di risposta di migrazione utenti

userAttributes

Questo campo è obbligatorio.

Questo campo deve contenere una o più coppie nome-valore archiviate da Amazon Cognito nel profilo utente del bacino d'utenza e utilizzate come attributi utente. Puoi includere attributi utenti standard e personalizzati. Gli attributi personalizzati richiedono il prefisso `custom:` per distinguerli dagli attributi standard. Per ulteriori informazioni sugli attributi, consulta Attributi personalizzati.

> **Note**
>
> Per poter reimpostare le password nel flusso di reimpostazione della password, gli utenti devono avere un indirizzo e-mail o un numero di telefono verificati. Amazon Cognito invia un messaggio contenente un codice di reimpostazione della password all'indirizzo e-mail o al numero di telefono specificati negli attributi utente.

Attributes	Requisito
Tutti gli attributi contrassegnati come obbligatori al momento della creazione del bacino d'utenza	Se durante la migrazione manca uno qualsiasi degli attributi obbligatori, Amazon Cognito utilizza valori di default.
`username`	Obbligatorio se hai configurato il pool di utenti con attributi alias oltre al nome utente per l'accesso e l'utente ha inserito un valore alias valido come nome utente. Questo valore alias può essere un indirizzo e-mail, un nome utente preferito o un numero di telefono. Se la richiesta e il pool di utenti soddisfano i requisiti relativi agli alias, la risposta della funzione deve assegnare il `username` parametro ricevuto a un attributo alias. Inoltre, la risposta deve assegnare il proprio valore all'attributo. `username` Se il bacino d'utenza non soddisfa le condizioni richieste per mappare lo `username` ricevuto a un alias, il parametro `username` nella

Attributes	Requisito
	risposta deve corrispondere esattamente alla richiesta o essere omesso. **Note** Lo `username` deve essere univoco all'interno del bacino d'utenti.

finale UserStatus

È possibile impostare questo parametro su `CONFIRMED` per confermare automaticamente gli utenti, in modo che possano accedere con le password precedenti. Gli utenti che imposti su `CONFIRMED` non devono intraprendere ulteriori azioni prima di poter effettuare l'accesso. Se non imposti questo attributo su `CONFIRMED`, esso rimane impostato su `RESET_REQUIRED`.

Un `finalUserStatus` di `RESET_REQUIRED` significa che l'utente deve modificare la propria password immediatamente dopo la migrazione al momento dell'accesso e l'app client deve gestire l'eccezione `PasswordResetRequiredException` durante il flusso di autenticazione.

Note

Amazon Cognito non applica la policy relativa alla forza della password che hai configurato per il bacino d'utenza durante la migrazione utilizzando il trigger Lambda. Se la password non soddisfa la tua policy per le password, Amazon Cognito la accetta comunque e continua a migrare l'utente. Per applicare la policy di protezione delle password e rifiutare le password che non soddisfano la policy, convalida la validità della password nel codice. Quindi, se la password non soddisfa i criteri, imposta final UserStatus su`RESET_REQUIRED`.

messageAction

Se desideri impedire l'invio del messaggio di benvenuto che Amazon Cognito invia di solito ai nuovi utenti, imposta questo parametro su `SUPPRESS`. Se la funzione non restituisce questo parametro, Amazon Cognito invia il messaggio di benvenuto.

desiderato DeliveryMediums

Per inviare il messaggio di benvenuto tramite e-mail, imposta il valore su `EMAIL`; per inviarlo tramite SMS, imposta il valore su `SMS`. Se la funzione non restituisce questo parametro, Amazon Cognito invia il messaggio di benvenuto tramite SMS.

forza AliasCreation

Se si imposta questo parametro su `TRUE` e il numero di telefono o l'indirizzo e-mail nel UserAttributes parametro esiste già come alias con un altro utente, la chiamata API migra l'alias dall'utente precedente a quello appena creato. L'utente precedente non potrà più effettuare l'accesso utilizzando tale alias.

Se imposti questo parametro su `FALSE` e l'alias esiste, Amazon Cognito non esegue la migrazione dell'utente e restituisce un errore all'app client.

Se non restituisci questo parametro, Amazon Cognito presuppone che il suo valore sia "false".

enableSMSMFA

Imposta questo parametro su `true` per richiedere all'utente migrato di completare l'autenticazione a più fattori (MFA) tramite SMS per accedere. Il pool di utenti deve avere l'MFA abilitato. Gli attributi dell'utente nei parametri della richiesta devono includere un numero di telefono, altrimenti la migrazione di quell'utente fallirà.

Esempio di migrazione di un utente con una password esistente

Questa funzione Lambda di esempio migra l'utente con una password esistente ed elimina il messaggio di benvenuto da Amazon Cognito.

Node.js

```
exports.handler = (event, context, callback) => {
  var user;

  if (event.triggerSource == "UserMigration_Authentication") {
    // authenticate the user with your existing user directory service
    user = authenticateUser(event.userName, event.request.password);
    if (user) {
      event.response.userAttributes = {
        email: user.emailAddress,
        email_verified: "true",
```

```
      };
      event.response.finalUserStatus = "CONFIRMED";
      event.response.messageAction = "SUPPRESS";
      context.succeed(event);
    } else {
      // Return error to Amazon Cognito
      callback("Bad password");
    }
  } else if (event.triggerSource == "UserMigration_ForgotPassword") {
    // Lookup the user in your existing user directory service
    user = lookupUser(event.userName);
    if (user) {
      event.response.userAttributes = {
        email: user.emailAddress,
        // required to enable password-reset code to be sent to user
        email_verified: "true",
      };
      event.response.messageAction = "SUPPRESS";
      context.succeed(event);
    } else {
      // Return error to Amazon Cognito
      callback("Bad password");
    }
  } else {
    // Return error to Amazon Cognito
    callback("Bad triggerSource " + event.triggerSource);
  }
};
```

Risoluzione dei problemi relativi al trigger di migrazione utente

Se la migrazione non funziona come previsto anche se la funzione restituisce la risposta documentata, verifica le seguenti cause comuni.

Il trigger non viene richiamato

Amazon Cognito richiama il trigger di migrazione utente solo quando l'utente non è già presente nel pool di utenti. Si applica solo ai flussi di accesso `USER_PASSWORD_AUTH`, `ADMIN_USER_PASSWORD_AUTH`, e `USER_AUTH` (con una password) e al flusso di password dimenticata. Non viene richiamato per l'accesso SRP (`USER_SRP_AUTH`), poiché la password dell'utente non raggiunge mai la funzione in quel flusso o per l'accesso senza

password. Se il trigger non si attiva, verifica che il client dell'app consenta uno di questi flussi e che l'utente non sia già presente nel pool.

Amazon Cognito non può richiamare la funzione

Se imposti il trigger con una policy basata sulle risorse sulla funzione anziché tramite la console Amazon Cognito, la policy deve consentire al responsabile del `cognito-idp.amazonaws.com` servizio di effettuare chiamate`lambda:InvokeFunction`, limitatamente all'ARN del tuo pool di utenti. La console aggiunge questa autorizzazione per te; una funzione collegata all'infrastruttura come codice o all'API Lambda deve essere aggiunta in modo esplicito. Senza di essa, la migrazione fallisce perché Amazon Cognito non può richiamare il trigger.

L'utente accede ma non viene creato alcun profilo

Amazon Cognito crea l'utente migrato solo quando la funzione restituisce la risposta nella forma esatta prevista: il `userAttributes` campo deve essere presente e deve includere tutti gli attributi contrassegnati come obbligatori nel pool di utenti. Una funzione che autentica l'utente ma restituisce un `response` oggetto vuoto o non valido o omette un attributo obbligatorio, completa la chiamata di accesso senza rendere persistente un profilo. Restituisce `userAttributes` con tutti gli attributi obbligatori e imposta se necessario. `finalUserStatus` `messageAction`

Trigger Lambda di messaggi personalizzati

Se disponi di uno standard esterno per i messaggi di posta elettronica e SMS che desideri inviare agli utenti o se desideri applicare la tua logica in fase di esecuzione alla formattazione dei messaggi utente, aggiungi un trigger di messaggio personalizzato al tuo pool di utenti. Il messaggio personalizzato Lambda riceve il contenuto di tutti i messaggi e-mail e SMS prima che il pool di utenti li invii. La funzione Lambda ha quindi l'opportunità di modificare il contenuto e l'oggetto del messaggio.

Amazon Cognito richiama questo trigger prima di inviare un messaggio di verifica via e-mail o telefono o un codice di autenticazione a più fattori (MFA). È possibile personalizzare il messaggio in modo dinamico con il trigger di messaggio personalizzato.

La richiesta include `codeParameter`. Questa è una stringa che agisce da segnaposto per il codice che Amazon Cognito fornisce all'utente. Inserisci la stringa `codeParameter` nel corpo del messaggio, nella posizione in cui desideri che appaia il codice di verifica. Alla ricezione di questa risposta, Amazon Cognito sostituisce la stringa `codeParameter` con il codice di verifica effettivo.

> **Note**
>
> L'evento di input per una funzione Lambda per messaggi personalizzati con la fonte del `CustomMessage_AdminCreateUser` trigger include un nome utente e un codice di verifica. Poiché un utente creato dall'amministratore deve ricevere sia il nome utente che il codice, la risposta della funzione deve includere variabili segnaposto per il nome utente e il codice. I segnaposto per il tuo messaggio sono i valori di e. `request.usernameParameter` `request.codeParameter` Questi valori sono in genere `{username}` e`{####}`, come procedura consigliata, fanno riferimento ai valori di input anziché codificare i nomi delle variabili.

> **Note**
>
> I messaggi configurati nel menu Modelli di messaggi della console Amazon Cognito possono sovrascrivere un corpo impostato dalla funzione di messaggio personalizzata. Se il messaggio della tua funzione non viene visualizzato, controlla se un modello di messaggio della console è configurato per lo stesso evento. Per ulteriori informazioni, consulta Configurazione MFA, autenticazione, verifica e messaggi di invito.

Argomenti

- Origini dei trigger Lambda di messaggi personalizzati
- Parametri del trigger Lambda di messaggi personalizzati
- Messaggio personalizzato per un esempio di registrazione
- Messaggio personalizzato per un esempio di amministratore che crea un utente

Origini dei trigger Lambda di messaggi personalizzati

Valore triggerSource	Event
`CustomMessage_SignUp`	Messaggio personalizzato: invio del codice di conferma post registrazione.
`CustomMessage_AdminCreateUser`	Messaggio personalizzato: invio della password temporanea a un nuovo utente.

Valore triggerSource	Event
`CustomMessage_ResendCode`	Messaggio personalizzato: nuovo invio del codice di conferma a un utente esistente.
`CustomMessage_ForgotPassword`	Messaggio personalizzato: invio del codice di conferma per una richiesta di password dimenticata.
`CustomMessage_UpdateUserAttribute`	Messaggio personalizzato: quando l'e-mail o il numero di telefono di un utente viene modificato, questo trigger invia automaticamente un codice di verifica all'utente. Non può essere utilizzato per altri attributi.
`CustomMessage_VerifyUserAttribute`	Messaggio personalizzato: questo trigger invia un codice di verifica all'utente, quando lo richiede manualmente per una nuova e-mail o numero di telefono.
`CustomMessage_Authentication`	Messaggio personalizzato: invio del codice MFA durante l'autenticazione.

Parametri del trigger Lambda di messaggi personalizzati

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        }
        "codeParameter": "####",
        "usernameParameter": "string",
        "clientMetadata": {
```

```
            "string": "string",
            . . .
        }
    },
    "response": {
        "smsMessage": "string",
        "emailMessage": "string",
        "emailSubject": "string"
    }
}
```

Parametri di richiesta di messaggi personalizzati

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

codeParameter

Una stringa per te, da utilizzare come segnaposto per il codice di verifica nel messaggio personalizzato.

Parametro username

Il nome utente. Amazon Cognito include questo parametro nelle richieste generate da utenti creati dall'amministratore.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di messaggi personalizzato. La richiesta che richiama una funzione di messaggio personalizzata non include i dati passati nel ClientMetadata parametro AdminInitiateAuthe le operazioni InitiateAuthAPI. Per passare questi dati alla funzione Lambda, puoi utilizzare il ClientMetadata parametro nelle seguenti azioni API:

- AdminResetUserPassword
- AdminRespondToAuthChallenge
- AdminUpdateUserAttributes
- ForgotPassword
- GetUserAttributeVerificationCode
- ResendConfirmationCode
- SignUp

- UpdateUserAttributes

Parametri di risposta di messaggi personalizzati

Nella risposta, specifica il testo personalizzato da usare nei messaggi destinati agli utenti. Per i vincoli di stringa che Amazon Cognito applica a questi parametri, consulta. MessageTemplateType

smsMessage

L'SMS personalizzato da inviare ai tuoi utenti. Deve includere il valore `codeParameter` che hai ricevuto nella richiesta.

emailMessage

Il messaggio e-mail personalizzato da inviare ai tuoi utenti. È possibile utilizzare la formattazione HTML nel parametro `emailMessage`. Deve includere il valore `codeParameter` ricevuto nella richiesta come variabile `{####}`. Amazon Cognito può utilizzare il parametro `emailMessage` solo se l'attributo `EmailSendingAccount` del bacino d'utenza è `DEVELOPER`. Se l'attributo `EmailSendingAccount` del bacino d'utenza non è `DEVELOPER` e viene restituito un parametro `emailMessage`, Amazon Cognito genera un codice di errore 400 `com.amazonaws.cognito.identity.idp.model.InvalidLambdaResponseException`. Quando scegli di utilizzare Amazon Simple Email Service (Amazon SES) per inviare messaggi e-mail, l'attributo `EmailSendingAccount` di un bacino d'utenza è `DEVELOPER`. In caso contrario, il valore è `COGNITO_DEFAULT`.

emailSubject

L'oggetto per il messaggio personalizzato. Puoi utilizzare il `emailSubject` parametro solo se l' EmailSendingAccountattributo del pool di utenti è. `DEVELOPER` Se l'attributo `EmailSendingAccount` del bacino d'utenza non è `DEVELOPER` e viene restituito un parametro `emailSubject`, Amazon Cognito genera un codice di errore 400 `com.amazonaws.cognito.identity.idp.model.InvalidLambdaResponseException`. Quando scegli di utilizzare Amazon Simple Email Service (Amazon SES) per inviare messaggi e-mail, l'attributo `EmailSendingAccount` di un bacino d'utenza è `DEVELOPER`. In caso contrario, il valore è `COGNITO_DEFAULT`

Messaggio personalizzato per un esempio di registrazione

Questa funzione Lambda di esempio permette di personalizzare un'e-mail o un SMS quando il servizio richiede che un'app invii un codice di verifica all'utente.

Amazon Cognito può richiamare un trigger Lambda durante più eventi: dopo la registrazione, per rinviare un codice di verifica, nel caso di una password dimenticata o per verificare un attributo utente. La risposta include sia SMS sia e-mail. Il messaggio deve includere il parametro di codice "####". Questo parametro è il segnaposto per il codice di verifica che l'utente riceve.

La lunghezza massima di un messaggio di posta elettronica è di 20.000 UTF-8 caratteri. Questa lunghezza include il codice di verifica. In questi messaggi e-mail puoi utilizzare i tag HTML.

La lunghezza massima dei messaggi SMS è di 140 UTF-8 caratteri. Questa lunghezza include il codice di verifica.

> **Note**
>
> Se la tua funzione restituisce un messaggio e-mail più lungo di 20.000 UTF-8 caratteri, Amazon Cognito non restituisce alcun errore. Invece, elimina silenziosamente il messaggio personalizzato e invia il messaggio predefinito per quell'evento. Se il messaggio personalizzato non raggiunge gli utenti, verificane prima la lunghezza rispetto a questo limite.

Node.js

```
const handler = async (event) => {
  if (event.triggerSource === "CustomMessage_SignUp") {
    const message = `Thank you for signing up. Your confirmation code is
 ${event.request.codeParameter}.`;
    event.response.smsMessage = message;
    event.response.emailMessage = message;
    event.response.emailSubject = "Welcome to the service.";
  }
  return event;
};

export { handler };
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
 "version": "1",
 "region": "us-west-2",
 "userPoolId": "us-west-2_EXAMPLE",
 "userName": "test-user",
 "callerContext": {
  "awsSdkVersion": "aws-sdk-unknown-unknown",
  "clientId": "1example23456789"
 },
 "triggerSource": "CustomMessage_SignUp",
 "request": {
  "userAttributes": {
   "sub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
   "cognito:user_status": "CONFIRMED",
   "email_verified": "true",
   "phone_number_verified": "true",
   "phone_number": "+12065551212",
   "email": "test-user@example.com"
  },
  "codeParameter": "{####}",
  "linkParameter": "{##Click Here##}",
  "usernameParameter": "None"
 },
 "response": {
  "smsMessage": "None",
  "emailMessage": "None",
  "emailSubject": "None"
 }
}
```

Messaggio personalizzato per un esempio di amministratore che crea un utente

La richiesta che Amazon Cognito ha inviato a questo esempio di messaggio personalizzato (funzione Lambda) ha un `triggerSource` valore di, un nome utente `CustomMessage_AdminCreateUser` e una password temporanea. La funzione viene compilata `${event.request.codeParameter}` a partire dalla password temporanea nella richiesta e `${event.request.usernameParameter}` dal nome utente nella richiesta.

I messaggi personalizzati devono inserire i valori di `codeParameter` e `usernameParameter` dentro `smsMessage` e `emailMessage` nell'oggetto di risposta. In questo esempio, la funzione scrive lo stesso messaggio nei campi di risposta `event.response.smsMessage` e`event.response.emailMessage`.

La lunghezza massima di un messaggio di posta elettronica è di 20.000 UTF-8 caratteri. Questa lunghezza include il codice di verifica. Puoi usare i tag HTML in queste e-mail. La lunghezza massima dei messaggi SMS è di 140 UTF-8 caratteri. Questa lunghezza include il codice di verifica.

> **Note**
>
> Se la tua funzione restituisce un messaggio e-mail più lungo di 20.000 UTF-8 caratteri, Amazon Cognito non restituisce alcun errore. Invece, elimina silenziosamente il messaggio personalizzato e invia il messaggio predefinito per quell'evento. Se il messaggio personalizzato non raggiunge gli utenti, verificane prima la lunghezza rispetto a questo limite.

La risposta include sia SMS sia e-mail.

Node.js

```
const handler = async (event) => {
  if (event.triggerSource === "CustomMessage_AdminCreateUser") {
    const message = `Welcome to the service. Your user name is
 ${event.request.usernameParameter}. Your temporary password is
 ${event.request.codeParameter}`;
    event.response.smsMessage = message;
    event.response.emailMessage = message;
    event.response.emailSubject = "Welcome to the service";
  }
  return event;
};

export { handler };
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
  "version": 1,
  "triggerSource": "CustomMessage_AdminCreateUser",
  "region": "<region>",
  "userPoolId": "<userPoolId>",
  "userName": "<userName>",
  "callerContext": {
      "awsSdk": "<calling aws sdk with version>",
      "clientId": "<apps client id>",
      ...
  },
  "request": {
      "userAttributes": {
          "phone_number_verified": false,
          "email_verified": true,
           ...
      },
      "codeParameter": "####",
      "usernameParameter": "username"
  },
  "response": {
      "smsMessage": "<custom message to be sent in the message with code parameter
 and username parameter>"
      "emailMessage": "<custom message to be sent in the message with code parameter
 and username parameter>"
      "emailSubject": "<custom email subject>"
  }
}
```

Trigger Lambda del mittente personalizzato

Lambda attiva `CustomEmailSender` e `CustomSMSSender` supporta notifiche e-mail e SMS di terze parti nei pool di utenti. Per inviare notifiche agli utenti dall'interno del codice della funzione Lambda, puoi utilizzare i provider di SMS ed e-mail che desideri. Quando Amazon Cognito invia inviti, codici MFA, codici di conferma, codici di verifica e password temporanee agli utenti, gli eventi attivano le funzioni Lambda configurate. Amazon Cognito invia il codice e le password temporanee (segreti) alle funzioni Lambda attivate. Amazon Cognito crittografa questi segreti con una chiave gestita AWS KMS dal cliente e il. AWS Encryption SDK AWS Encryption SDK È una libreria di crittografia lato client che ti aiuta a crittografare e decrittografare dati generici.

CustomEmailSender

Amazon Cognito richiama questo trigger per inviare notifiche e-mail agli utenti.

PersonalizzaSMSSender

Amazon Cognito richiama questo trigger per inviare notifiche SMS agli utenti.

Concetti di crittografia

Amazon Cognito non invia i codici degli utenti in testo semplice negli eventi che invia a trigger mittenti personalizzati. Le funzioni Lambda devono decrittografare i codici negli eventi. I seguenti concetti sono l'architettura di crittografia che la funzione deve utilizzare per ottenere codici da fornire agli utenti.

AWS KMS

AWS KMS è un servizio gestito per creare e controllare AWS KMS chiavi. Queste chiavi crittografano i tuoi dati. Per ulteriori informazioni, consulta la pagina Che cos'è AWS Key Management Service?

Chiave KMS

Una chiave KMS è la rappresentazione logica di una chiave crittografica. La chiave KMS include metadati, ad esempio l'ID della chiave, la data di creazione, la descrizione e lo stato della chiave. La chiave KMS contiene anche il materiale della chiave utilizzato per crittografare e decrittare i dati. Per ulteriori informazioni, consulta AWS Chiavi KMS.

Chiavi KMS simmetriche

Una chiave KMS simmetrica è una chiave di crittografia a 256 bit che non mantiene crittografato AWS KMS . Per utilizzare una chiave KMS simmetrica, è necessario chiamare. AWS KMS Amazon Cognito utilizza le chiavi simmetriche. La stessa chiave viene usata per la crittografia e la decrittografia. Per ulteriori informazioni, consulta Chiavi KMS simmetriche.

Cose da sapere sui trigger Lambda personalizzati per mittenti

- Per configurare i bacini d'utenza affinché utilizzino questi trigger Lambda, puoi utilizzare la AWS CLI o l'SDK. Queste configurazioni non sono disponibili nella console Amazon Cognito.

  L'`UpdateUserPool`operazione imposta la configurazione Lambda. Le richieste di questa operazione richiedono tutti i parametri del pool di utenti e i parametri che si desidera modificare. Se

non fornisci tutti i parametri rilevanti, Amazon Cognito imposta i valori di tutti i parametri mancanti sui valori predefiniti. Come dimostrato nell'esempio AWS CLI che segue, includi le voci per tutte le funzioni Lambda che desideri aggiungere o conservare nel tuo pool di utenti. Per ulteriori informazioni, consulta Aggiornamento della configurazione del pool di utenti e del client dell'app.

```
   #Send this parameter in an 'aws cognito-idp update-user-pool' CLI command,
 including any existing
   #user pool configurations. This snippet also includes a pre sign-up trigger for
 syntax reference. The pre sign-up trigger
   #doesn't have a role in custom sender triggers.

     --lambda-config "PreSignUp=lambda-arn, \
                      CustomSMSSender={LambdaVersion=V1_0,LambdaArn=lambda-arn}, \
                      CustomEmailSender={LambdaVersion=V1_0,LambdaArn=lambda-arn}, \
                      KMSKeyID=key-id"
```

Per le richieste che utilizzano il corpo JSON del seguente `LambdaConfig` frammento di `UpdateUserPool` codice assegna funzioni personalizzate di invio di SMS e e-mail.

```
"LambdaConfig": {
   "KMSKeyID": "arn:aws:kms:us-
east-1:111122223333:key/a6c4f8e2-0c45-47db-925f-87854bc9e357",
   "CustomEmailSender": {
      "LambdaArn": "arn:aws:lambda:us-east-1:111122223333:function:MyFunction",
      "LambdaVersion": "V1_0"
   },
   "CustomSMSSender": {
      "LambdaArn": "arn:aws:lambda:us-east-1:111122223333:function:MyFunction",
      "LambdaVersion": "V1_0"
   }
```

- Per rimuovere un trigger Lambda del mittente personalizzato con `update-user-pool` AWS CLI un comando, ometti `CustomSMSSender` il parametro `CustomEmailSender` o `--lambda-config` da e includi tutti gli altri trigger che desideri utilizzare con il tuo pool di utenti.

  Per rimuovere un trigger Lambda del mittente personalizzato con una richiesta API `UpdateUserPool`, ometti il parametro `CustomSMSSender` o `CustomEmailSender` dal corpo della richiesta contenente il resto della configurazione del pool di utenti.

- Amazon Cognito crea una sequenza di escape HTML di caratteri riservati come < (`<`) e > (`>`) nella password temporanea dell'utente. Questi caratteri potrebbero essere visualizzati nelle password temporanee inviate da Amazon Cognito alla funzione del mittente e-mail personalizzata, ma non vengono visualizzati nei codici di verifica temporanei. Per inviare password temporanee, la funzione Lambda deve eliminare questi caratteri dopo aver decrittografato la password e prima di inviare il messaggio all'utente.

Attivazione dei trigger Lambda personalizzati del mittente

Per utilizzare una logica personalizzata per inviare SMS o messaggi e-mail per il tuo pool di utenti, configura i trigger mittenti personalizzati. La procedura seguente assegna un trigger SMS personalizzato, un trigger e-mail personalizzato o entrambi al pool di utenti. Dopo aver aggiunto il trigger mittente personalizzato, Amazon Cognito invia sempre gli attributi utente, incluso il numero di telefono e il codice monouso, alla funzione Lambda anziché il comportamento predefinito che prevede l'invio di un SMS o un messaggio e-mail.

1. Crea una chiave di crittografia simmetrica in (). AWS Key Management Service AWS KMS Amazon Cognito genera segreti (password temporanee, codici di verifica, password monouso di autenticazione e codici di conferma), quindi utilizza questa chiave KMS per crittografare i segreti. AWS Encryption SDK È quindi possibile utilizzare la funzione AWS Encryption SDK in Lambda per decrittografare i segreti e inviarli all'utente in formato testo non crittografato.
2. Il principale IAM che crea o aggiorna il tuo pool di utenti crea una concessione una tantum per la chiave KMS utilizzata da Amazon Cognito per crittografare il codice. Concedi queste `CreateGrant` autorizzazioni principali per la tua chiave KMS. Affinché questo esempio di politica delle chiavi KMS sia efficace, l'amministratore che aggiorna il pool di utenti deve accedere con una sessione con un ruolo presunto per il ruolo IAM. `arn:aws:iam::111222333444:role/my-example-administrator-role`

   Applica la seguente politica basata sulle risorse, modificata per il tuo ambiente, alla tua chiave KMS.

   JSON

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

```
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/my-example-administrator-
role"
        },
        "Action": "kms:CreateGrant",
        "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1example-2222-3333-4444-999example",
        "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:userpool-id": "us-west-2_EXAMPLE"
            }
        }
    },
    {
        "Sid": "Allow Lambda to decrypt",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/my-lambda-function-role"
        },
        "Action": "kms:Decrypt",
        "Resource": "*"
    }]
}
```

3. Crea una funzione Lambda per il trigger del mittente personalizzato. Amazon Cognito utilizza l'SDK di crittografia AWS per crittografare i segreti, le password temporanee e i codici di autorizzazione delle richieste API degli utenti.

   a. Assegna un ruolo di esecuzione Lambda con almeno le autorizzazioni per la `kms:Decrypt` tua chiave KMS.

   b. Componi il codice della funzione Lambda per inviare i tuoi messaggi. L'evento di input della tua funzione contiene un segreto. Nella tua funzione, decrittografa il segreto con AWS Encryption SDK ed elabora tutti i metadati pertinenti. Quindi invia il codice, il messaggio personalizzato e il numero di telefono di destinazione all'API personalizzata che distribuisce il messaggio.

   c. Aggiungi il AWS Encryption SDK alla tua funzione Lambda. Per ulteriori informazioni, consulta la sezione relativa ai linguaggi di programmazione di SDK di crittografia AWS. Per aggiornare il pacchetto Lambda, completa la procedura seguente.

      i. Esporta la funzione Lambda come file .zip in Console di gestione AWS.

ii. Apri la tua funzione e aggiungi il AWS Encryption SDK. Per ulteriori informazioni e i link di download, consulta la sezione relative ai linguaggi di programmazione di AWS Encryption SDK nella Guida per gli sviluppatori di AWS Encryption SDK .

iii. Comprimi la funzione con le dipendenze dell'SDK e carica la funzione in Lambda. Per ulteriori informazioni, consulta Distribuzione di funzioni Lambda come archivi di file .zip nella Guida per gli sviluppatori di AWS Lambda .

4. Concedi l'accesso `cognito-idp.amazonaws.com` al principale del servizio Amazon Cognito per richiamare la funzione Lambda.

   Il AWS CLI comando seguente concede ad Amazon Cognito l'autorizzazione a richiamare la funzione Lambda:

   ```
   aws lambda add-permission --function-name lambda_arn --statement-id
    "CognitoLambdaInvokeAccess" --action lambda:InvokeFunction --principal cognito-
   idp.amazonaws.com
   ```

5. Genera una richiesta UpdateUserPoolAPI con un `LambdaConfig` parametro che aggiunge trigger Lambda del mittente personalizzati. Non puoi aggiungere trigger di questo tipo nella console Amazon Cognito. I trigger mittenti personalizzati richiedono `LambdaConfig` i parametri di `KMSKeyID` e `CustomSMSSender` o `CustomEmailSender` (o entrambi).

Trigger Lambda del mittente di e-mail personalizzato

Quando assegni un trigger del mittente dell'e-mail personalizzato al pool di utenti, Amazon Cognito richiama una funzione Lambda anziché il suo comportamento predefinito quando un evento utente richiede l'invio di un messaggio e-mail. Con un trigger mittente personalizzato, la tua AWS Lambda funzione può inviare notifiche e-mail agli utenti tramite un metodo e un provider di tua scelta. Il codice personalizzato della funzione deve elaborare e distribuire tutti i messaggi e-mail del pool di utenti.

Questo trigger è utile per scenari in cui potresti voler avere un maggiore controllo sul modo in cui il tuo pool di utenti invia i messaggi e-mail. La tua funzione Lambda può personalizzare la chiamata alle operazioni dell'API di Amazon SES, ad esempio quando desideri gestire più identità verificate o incrociate. Regioni AWS La tua funzione potrebbe anche reindirizzare i messaggi verso un altro mezzo di consegna o un servizio di terze parti.

Per informazioni su come configurare un trigger personalizzato per il mittente dell'e-mail, consulta. Attivazione dei trigger Lambda personalizzati del mittente

Origini dei trigger Lambda del mittente di e-mail personalizzato

La tabella che segue mostra gli eventi di attivazione per le origini dei trigger delle e-mail personalizzate nel codice Lambda.

`TriggerSource value`	Event
`CustomEmailSender_SignUp`	Un utente si registra e Amazon Cognito invia un messaggio di benvenuto.
`CustomEmailSender_Authentication`	Un utente accede e Amazon Cognito invia un codice OTP o MFA via e-mail.
`CustomEmailSender_ForgotPassword`	Un utente richiede un codice per reimpostare la password.
`CustomEmailSender_ResendCode`	Un utente richiede un codice di conferma dell'account sostitutivo.
`CustomEmailSender_UpdateUserAttribute`	Un utente aggiorna un indirizzo e-mail o un numero di telefono e Amazon Cognito gli invia un codice per verificare l'attributo.
`CustomEmailSender_VerifyUserAttribute`	Un utente crea un nuovo attributo indirizzo e-mail o numero di telefono e Amazon Cognito gli invia un codice per verificare l'attributo.
`CustomEmailSender_AdminCreateUser`	Crei un nuovo utente nel tuo bacino d'utenza e Amazon Cognito gli invia una password temporanea.
`CustomEmailSender_AccountTakeOverNotification`	Amazon Cognito rileva il tentativo di prendere il controllo di un account utente e invia una notifica all'utente.

Parametri del trigger Lambda del mittente di e-mail personalizzato

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "type": "customEmailSenderRequestV1",
        "code": "string",
        "clientMetadata": {
            "string": "string",
             . . .
            },
        "userAttributes": {
            "string": "string",
            . . .
         }
}
```

Parametri della richiesta del mittente di e-mail personalizzato

tipo

La versione della richiesta. Per un evento del mittente di e-mail personalizzato, il valore di questa stringa è sempre `customEmailSenderRequestV1`.

code

Il codice crittografato che la funzione può decrittografare e inviare all'utente.

clientMetadata

Una o più coppie chiave-valore che puoi fornire come input personalizzato al trigger della funzione Lambda del mittente di e-mail personalizzato. Per passare questi dati alla funzione Lambda, puoi utilizzare il ClientMetadata parametro nelle azioni AdminRespondToAuthChallengee RespondToAuthChallengeAPI. Amazon Cognito non include i dati del ClientMetadata parametro AdminInitiateAuthe le operazioni InitiateAuthAPI nella richiesta che passa alla funzione di post-autenticazione.

> **Note**
>
> Amazon Cognito invia `ClientMetadata` a e-mail personalizzate funzioni di attivazione in caso di eventi con le seguenti fonti di attivazione:
>
> - `CustomEmailSender_ForgotPassword`

- `CustomEmailSender_SignUp`
- `CustomEmailSender_Authentication`

Amazon Cognito non invia `ClientMetadata` eventi trigger con l'origine. `CustomEmailSender_AccountTakeOverNotification`

userAttributes

Una o più coppie chiave-valore che rappresentano gli attributi utente.

Parametri di risposta del mittente di e-mail personalizzato

Amazon Cognito non prevede di restituire ulteriori informazioni nella risposta del mittente di e-mail personalizzato. La funzione Lambda deve interpretare l'evento e decrittografare il codice, quindi recapitare il contenuto del messaggio. Una funzione tipica assembla un messaggio e-mail e lo indirizza a un relay SMTP di terze parti.

Esempio di codice

Nel seguente esempio Node.js viene illustrato come elaborare un evento di messaggio e-mail nella funzione Lambda del mittente di e-mail personalizzata. Per questo esempio si presuppone che la funzione abbia due variabili d'ambiente definite.

`KEY_ID`

L'ID della chiave KMS che desideri utilizzare per crittografare e decrittografare i codici degli utenti.

`KEY_ARN`

Il nome della risorsa Amazon (ARN) della chiave KMS che desideri utilizzare per crittografare e decrittografare i codici degli utenti.

Per implementare questa funzione

1. Installa l'ultima versione di NodeJS nel tuo spazio di lavoro per sviluppatori.
2. Crea un nuovo progetto NodeJS nel tuo spazio di lavoro.
3. Inizializza il tuo progetto con. `npm init -y`
4. Crea lo script per la funzione Lambda:. `touch index.mjs`

5. Incolla il contenuto dell'esempio seguente in`index.mjs`.
6. Scarica la dipendenza del progetto, AWS Encryption SDK:`npm install @aws-crypto/client-node`.
7. Comprimi la cartella del progetto in un file:`zip -r my_deployment_package.zip .`.
8. Implementa il file ZIP nella tua funzione.

Questa funzione di esempio decrittografa il codice e, per gli eventi di iscrizione, simula l'invio di un messaggio e-mail all'indirizzo e-mail dell'utente.

```
import { KmsKeyringNode, buildClient, CommitmentPolicy } from '@aws-crypto/client-
node';

// Configure the encryption SDK client with the KMS key from the environment variables
const { encrypt, decrypt } = buildClient(
    CommitmentPolicy.REQUIRE_ENCRYPT_ALLOW_DECRYPT
);

const generatorKeyId = process.env.KEY_ID;
const keyIds = [process.env.KEY_ARN];
const keyring = new KmsKeyringNode({ generatorKeyId, keyIds });

// Example function to simulate sending email.
// This example logs message details to CloudWatch Logs from your Lambda function.
// Update this function with custom logic that sends an email message to 'emailaddress'
 with body 'message'.
const sendEmail = async (emailAddress, message) => {
    // Log the destination with the email address masked.
    console.log(`Simulating email send to ${emailAddress.replace(/[^@.]/g, '*')}`);
    // Log the message with the code masked.
    console.log(`Message content: ${message.replace(/\b\d{6,8}\b/g, '********')}`);
    // Simulate API delay
    await new Promise(resolve => setTimeout(resolve, 100));
    console.log('Email sent successfully');
    return true;
};

export const handler = async (event) => {
    try {
        // Decrypt the secret code using encryption SDK
        let plainTextCode;
        if (event.request.code) {
```

```
            const { plaintext, messageHeader } = await decrypt(keyring,
 Buffer.from(event.request.code, 'base64'));
            plainTextCode = Buffer.from(plaintext).toString('utf-8');
        }

        // Handle different trigger sources
        if (event.triggerSource == 'CustomEmailSender_SignUp') {
            const emailAddress = event.request.userAttributes.email;
            const message = `Welcome! Your verification code is: ${plainTextCode}`;
            await sendEmail(emailAddress, message);
        }
        else if (event.triggerSource == 'CustomEmailSender_ResendCode') {
            // Handle resend code
        }
        else if (event.triggerSource == 'CustomEmailSender_ForgotPassword') {
            // Handle forgot password
        }
        else if (event.triggerSource == 'CustomEmailSender_UpdateUserAttribute') {
            // Handle update attribute
        }
        else if (event.triggerSource == 'CustomEmailSender_VerifyUserAttribute') {
            // Handle verify attribute
        }
        else if (event.triggerSource == 'CustomEmailSender_AdminCreateUser') {
            // Handle admin create user
        }
        else if (event.triggerSource == 'CustomEmailSender_Authentication') {
            // Handle authentication
        }
        else if (event.triggerSource ==
 'CustomEmailSender_AccountTakeOverNotification') {
            // Handle account takeover notification
        }

        return;
    } catch (error) {
        console.error('Error in custom email sender:', error);
        throw error;
    }
};
```

Trigger Lambda del mittente di SMS personalizzato

Quando assegni un trigger del mittente SMS personalizzato al pool di utenti, Amazon Cognito richiama una funzione Lambda anziché il suo comportamento predefinito quando un evento utente richiede l'invio di un messaggio e-mail. Con un trigger mittente personalizzato, la tua AWS Lambda funzione può inviare notifiche SMS agli utenti tramite un metodo e un provider di tua scelta. Il codice personalizzato della funzione deve elaborare e distribuire tutti i messaggi SMS del pool di utenti.

Questo trigger è utile per scenari in cui potresti voler avere un maggiore controllo sul modo in cui il tuo pool di utenti invia i messaggi SMS. La tua funzione Lambda può personalizzare la chiamata alle operazioni dell'API Amazon SNS, ad esempio quando desideri gestire più IDs origini o incroci. Regioni AWS La tua funzione potrebbe anche reindirizzare i messaggi verso un altro mezzo di consegna o un servizio di terze parti.

Per informazioni su come configurare un trigger personalizzato per il mittente dell'e-mail, consulta. Attivazione dei trigger Lambda personalizzati del mittente

Origini dei trigger Lambda del mittente di SMS personalizzato

La tabella che segue mostra l'evento di attivazione per le origini dei trigger di SMS personalizzati nel codice Lambda.

`TriggerSource value`	Event
`CustomSMSSender_SignUp`	Un utente si registra e Amazon Cognito invia un messaggio di benvenuto.
`CustomSMSSender_ForgotPassword`	Un utente richiede un codice per reimpostare la password.
`CustomSMSSender_ResendCode`	Un utente richiede un nuovo codice per confermare la registrazione.
`CustomSMSSender_VerifyUserAttribute`	Un utente crea un nuovo attributo indirizzo e-mail o numero di telefono e Amazon Cognito gli invia un codice per verificare l'attributo.
`CustomSMSSender_UpdateUserAttribute`	Un utente aggiorna un indirizzo e-mail o un numero di telefono e Amazon Cognito gli invia un codice per verificare l'attributo.

TriggerSource value	Event
CustomSMSSender_Authentication	Un utente accede e Amazon Cognito invia un codice OTP o MFA via SMS.
CustomSMSSender_AdminCreateUser	Crei un nuovo utente nel tuo bacino d'utenza e Amazon Cognito gli invia una password temporanea.

Parametri del trigger Lambda del mittente di SMS personalizzato

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "type": "customSMSSenderRequestV1",
        "code": "string",
        "clientMetadata": {
            "string": "string",
             . . .
            },
        "userAttributes": {
            "string": "string",
            . . .
         }
}
```

Parametri di richiesta del mittente di SMS personalizzato

tipo

La versione della richiesta. Per un evento del mittente di SMS personalizzato, il valore di questa stringa è sempre `customSMSSenderRequestV1`.

code

Il codice crittografato che la funzione può decrittografare e inviare all'utente.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato al trigger della funzione Lambda del mittente di SMS personalizzato. Per passare questi dati alla funzione Lambda, puoi utilizzare il ClientMetadata parametro nelle azioni AdminRespondToAuthChallengee RespondToAuthChallengeAPI. Amazon Cognito non include i dati del ClientMetadata parametro AdminInitiateAuthe le operazioni InitiateAuthAPI nella richiesta che passa alla funzione di post-autenticazione.

userAttributes

Una o più coppie chiave-valore che rappresentano gli attributi utente.

Parametri di risposta del mittente di SMS personalizzato

Amazon Cognito non prevede di restituire ulteriori informazioni nella risposta. La funzione può utilizzare le operazioni API per interrogare e modificare le risorse o registrare i metadati degli eventi in un sistema esterno.

Esempio di codice

Nel seguente esempio Node.js viene illustrato come elaborare un evento di messaggio SMS nella funzione Lambda del mittente di SMS personalizzata. Per questo esempio si presuppone che la funzione abbia due variabili d'ambiente definite.

`KEY_ID`

L'ID della chiave KMS che desideri utilizzare per crittografare e decrittografare i codici degli utenti.

`KEY_ARN`

Il nome della risorsa Amazon (ARN) della chiave KMS che desideri utilizzare per crittografare e decrittografare i codici degli utenti.

Per implementare questa funzione

1. Installa l'ultima versione di NodeJS nel tuo spazio di lavoro per sviluppatori.
2. Crea un nuovo progetto NodeJS nel tuo spazio di lavoro.
3. Inizializza il tuo progetto con. `npm init -y`
4. Crea lo script per la funzione Lambda:. `touch index.mjs`

5. Incolla il contenuto dell'esempio seguente in`index.mjs`.
6. Scarica la dipendenza del progetto, AWS Encryption SDK:`npm install @aws-crypto/client-node`.
7. Comprimi la cartella del progetto in un file:`zip -r my_deployment_package.zip .`.
8. Implementa il file ZIP nella tua funzione.

```
import { KmsKeyringNode, buildClient, CommitmentPolicy } from '@aws-crypto/client-
node';

// Configure the encryption SDK client with the KMS key from the environment variables
const { encrypt, decrypt } = buildClient(
    CommitmentPolicy.REQUIRE_ENCRYPT_ALLOW_DECRYPT
);

const generatorKeyId = process.env.KEY_ID;
const keyIds = [process.env.KEY_ARN];
const keyring = new KmsKeyringNode({ generatorKeyId, keyIds });

// Example function to simulate sending SMS.
// This example logs message details to CloudWatch Logs from your Lambda function.
// Update this function with custom logic that sends an SMS message to 'phoneNumber'
 with body 'message'.
const sendSMS = async (phoneNumber, message) => {
    // Log the destination with the phone number masked.
    console.log(`Simulating SMS send to ${phoneNumber.replace(/[^+]/g, '*')}`);
    // Log the message with the code masked.
    console.log(`Message content: ${message.replace(/\b\d{6,8}\b/g, '********')}`);
    // Simulate API delay
    await new Promise(resolve => setTimeout(resolve, 100));
    console.log('SMS sent successfully');
    return true;
};

export const handler = async (event) => {
    try {
        // Decrypt the secret code using encryption SDK
        let plainTextCode;
        if (event.request.code) {
            const { plaintext, messageHeader } = await decrypt(keyring,
 Buffer.from(event.request.code, 'base64'));
            plainTextCode = Buffer.from(plaintext).toString('utf-8');
```

```
        }

        // Handle different trigger sources
        if (event.triggerSource == 'CustomSMSSender_SignUp') {
            const phoneNumber = event.request.userAttributes.phone_number;
            const message = `Welcome! Your verification code is: ${plainTextCode}`;
            await sendSMS(phoneNumber, message);
        }
        else if (event.triggerSource == 'CustomSMSSender_ResendCode') {
            // Handle resend code
        }
        else if (event.triggerSource == 'CustomSMSSender_ForgotPassword') {
            // Handle forgot password
        }
        else if (event.triggerSource == 'CustomSMSSender_UpdateUserAttribute') {
            // Handle update attribute
        }
        else if (event.triggerSource == 'CustomSMSSender_VerifyUserAttribute') {
            // Handle verify attribute
        }
        else if (event.triggerSource == 'CustomSMSSender_AdminCreateUser') {
            // Handle admin create user
        }
        return;
    } catch (error) {
        console.error('Error in custom SMS sender:', error);
        throw error;
    }
};
```

Argomenti

- Valutazione delle funzionalità di messaggio SMS con una funzione del mittente di SMS personalizzato

Valutazione delle funzionalità di messaggio SMS con una funzione del mittente di SMS personalizzato

Una funzione Lambda del mittente di SMS personalizzato accetta i messaggi SMS inviati dal bacino d'utenza e fornisce il contenuto in base alla logica personalizzata. Amazon Cognito invia i Parametri del trigger Lambda del mittente di SMS personalizzato alla tua funzione. La funzione può fare ciò che vuoi con queste informazioni. Ad esempio, puoi inviare il codice a un argomento Amazon Simple

Notification Service (Amazon SNS). Un sottoscrittore dell'argomento Amazon SNS può essere un messaggio SMS, un endpoint HTTPS o un indirizzo e-mail.

Per creare un ambiente di test per la messaggistica SMS di Amazon Cognito con una funzione Lambda del mittente SMS personalizzata, amazon-cognito-user-poolconsulta development-and-testing-with - sms-redirected-to-email - nella libreria aws-samples su. GitHub Il repository contiene AWS CloudFormation modelli che possono creare un nuovo pool di utenti o utilizzare un pool di utenti che già possiedi. Questi modelli creano funzioni Lambda e un argomento Amazon SNS. La funzione Lambda che il modello assegna come trigger del mittente di SMS personalizzato, reindirizza i messaggi SMS ai sottoscrittori dell'argomento Amazon SNS.

Quando distribuisci questa soluzione in un bacino d'utenza, tutti i messaggi che Amazon Cognito invia di solito tramite messaggi SMS vengono inviati dalla funzione Lambda a un indirizzo e-mail centrale. Utilizza questa soluzione per personalizzare e visualizzare in anteprima i messaggi SMS e per testare gli eventi del bacino d'utenza che causano l'invio di un messaggio SMS da parte di Amazon Cognito. Dopo aver completato i test, ripristina lo CloudFormation stack o rimuovi l'assegnazione personalizzata della funzione di mittente SMS dal tuo pool di utenti.

> **Important**
>
> Non utilizzate i modelli in amazon-cognito-user-pool- development-and-testing-with - sms-redirected-to-email per creare un ambiente di produzione. La funzione Lambda del mittente di SMS personalizzato nella soluzione simula i messaggi SMS, ma li invia tutti a un unico indirizzo e-mail centrale. Prima di poter inviare messaggi SMS in un bacino d'utenza di Amazon Cognito di produzione, è necessario completare i requisiti indicati in Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.

Messaggistica e analisi

Configura la messaggistica e-mail e SMS per il tuo pool di utenti e configura l'analisi di Amazon Pinpoint per monitorare il coinvolgimento degli utenti.

Argomenti

- Impostazioni e-mail per i bacini d'utenza di Amazon Cognito
- Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito
- Utilizzo di Amazon Pinpoint per l'analisi dei pool di utenti

Impostazioni e-mail per i bacini d'utenza di Amazon Cognito

Alcuni eventi della tua applicazione possono causare l'invio di e-mail da parte di Amazon Cognito ai tuoi utenti. Ad esempio, se configuri il bacino d'utenza in modo da richiedere la verifica e-mail, Amazon Cognito invia un'e-mail quando un utente accede a un nuovo account nell'app o quando reimposta la password. A seconda dell'azione che attiva l'e-mail, l'e-mail contiene un codice di verifica o una password temporanea.

Per gestire l'invio delle e-mail, è possibile utilizzare le seguenti opzioni:

- La configurazione e-mail predefinita integrata nel servizio Amazon Cognito.
- La configurazione di Amazon Simple Email Service (Amazon SES).

Puoi modificare l'opzione di distribuzione dopo aver creato il pool di utenti.

Amazon Cognito invia messaggi e-mail agli utenti con un codice che possono inserire o un collegamento URL che possono selezionare. Nella tabella seguente vengono mostrati gli eventi che possono generare un messaggio di posta elettronica.

Opzioni relative ai messaggi

Attività	Operazione API	Opzioni di distribuzione	Opzioni di formato	Personalizzabile	Modello di messaggio
Password dimenticata	ForgotPassword, AdminResetUserPassword	E-mail, SMS	code	Sì	Messaggio di verifica
Invito	AdminCreateUser	E-mail, SMS	code	Sì	Messaggio di invito
Self-registration	SignUp, ResendConfirmationCode	E-mail, SMS	codice, link	Sì	messaggio di verifica
Verifica dell'indirizzo e-mail o del	UpdateUserAttributes, AdminUpdateUserAttributes,	E-mail, SMS	code	Sì	Messaggio di verifica

Attività	Operazione API	Opzioni di distribuzione	Opzioni di formato	Personalizzabile	Modello di messaggio
numero di telefono	GetUserAttributeVerificationCode				
Multi-factor autenticazione (MFA)	AdminInitiateAuth, InitiateAuth	Email[1], SMS, app di autenticazione	code	Sì[2]	Messaggio MFA
One-time autenticazione tramite password (OTP)	AdminInitiateAuth, InitiateAuth	E-mail[1], SMS	code	Sì	Messaggio MFA [3]

[1] Richiede un piano di funzionalità Essentials o superiore e la configurazione e-mail di Amazon SES.

[2] Per SMS e messaggi e-mail.

[3] È possibile personalizzare il modello di messaggio MFA solo quando l'MFA è obbligatoria o facoltativa nel pool di utenti. Quando la MFA è inattiva, Amazon Cognito invia password monouso con il modello predefinito.

Amazon SES addebita il costo dei messaggi e-mail. Per ulteriori informazioni, consulta la pagina dei Prezzi di Amazon SES.

Per ulteriori informazioni sull'MFA delle e-mail, consulta. MFA per SMS e messaggi e-mail

Amazon Cognito potrebbe impedire la consegna di messaggi e-mail o SMS aggiuntivi verso un'unica destinazione in un breve periodo di tempo. Se ritieni che il tuo pool di utenti sia interessato, configura e rivedi i log per individuare eventuali errori di recapito dei messaggi, quindi contatta il team del tuo account.

Configurazione e-mail predefinita

Amazon Cognito può utilizzare la sua configurazione e-mail predefinita per gestire le consegne di posta elettronica per te. Quando utilizzi l'opzione di default, Amazon Cognito limita il numero di e-

mail che puoi inviare al giorno al tuo bacino d'utenza. Per ulteriori informazioni sui limiti del servizio, consulta Quote in Amazon Cognito. Per gli ambienti di produzione tipici, il limite di e-mail predefinito è inferiore al volume di distribuzione richiesto. Per consentire un volume di consegna più alto, è possibile utilizzare la configurazione e-mail di Amazon SES.

Quando si utilizza la configurazione predefinita, si utilizzano risorse Amazon SES gestite da AWS per inviare messaggi e-mail. Amazon SES aggiunge indirizzi e-mail che restituiscono un mancato recapito permanente a un elenco di eliminazione a livello di account o a un elenco di eliminazione globale. Se un indirizzo e-mail non recapitabile diventa recapitabile in un secondo momento, non puoi controllarne la rimozione dall'elenco di soppressione mentre il pool di utenti è configurato per utilizzare la configurazione predefinita. Un indirizzo e-mail può rimanere nell'elenco di soppressione gestito a tempo indeterminato. AWS Per gestire gli indirizzi e-mail non recapitabili, utilizza la configurazione e-mail di Amazon SES con un elenco di eliminazione a livello di account, come descritto nella prossima sezione.

Con la configurazione e-mail predefinita, puoi utilizzare uno dei seguenti indirizzi come FROM:

- L'indirizzo e-mail di default, ovvero no-reply@verificationemail.com.
- Un indirizzo e-mail personalizzato. Prima di poter utilizzare il tuo indirizzo e-mail, devi verificarlo con Amazon SES e concedere a Amazon Cognito l'autorizzazione per utilizzarlo.

Configurazione e-mail di Amazon SES

L'applicazione potrebbe richiedere un volume di distribuzione più elevato rispetto a quello disponibile con l'opzione predefinita. Per aumentare il volume di distribuzione, usa le risorse di Amazon SES con il bacino d'utenza per inviare e-mail ai tuoi utenti. Puoi anche monitorare la tua attività di invio e-mail quando invii messaggi e-mail con la tua configurazione Amazon SES.

Prima di poter utilizzare la tua configurazione Amazon SES, devi verificare uno o più indirizzi e-mail, o un dominio, con Amazon SES. Usa un indirizzo e-mail verificato o un indirizzo da un dominio verificato come indirizzo e-mail FROM (DA) da assegnare al bacino d'utenza. Quando Amazon Cognito invia un messaggio e-mail a un utente, utilizza il tuo indirizzo e-mail chiamando automaticamente Amazon SES.

Quando utilizzi la configurazione Amazon SES, si applicano le seguenti condizioni:

- I limiti di invio dei messaggi e-mail per il bacino d'utenza sono gli stessi che valgono per l'indirizzo e-mail verificato Amazon SES nel tuo Account AWS.

- Puoi gestire i tuoi messaggi ad indirizzi e-mail non recapitabili con un elenco di eliminazione a livello di account in Amazon SES che sostituisce l'elenco di eliminazione globale. Quando utilizzi un elenco di eliminazione a livello di account, i mancati recapiti dei messaggi e-mail influiscono sulla reputazione del tuo account come mittente. Per ulteriori informazioni, consulta Utilizzo dell'elenco di eliminazione a livello di account di Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

Regioni di configurazione degli indirizzi e-mail di Amazon SES

Il Regione AWS luogo in cui si crea un pool di utenti ha uno dei tre requisiti per la configurazione dei messaggi e-mail con Amazon SES. Puoi inviare messaggi e-mail da Amazon SES nella stessa regione del tuo pool di utenti, in diverse regioni tra cui la stessa regione o in una o più regioni remote. Per prestazioni ottimali, invia messaggi e-mail con un'identità verificata da Amazon SES nella stessa regione del tuo pool di utenti, se ne hai l'opzione.

Categorie di requisiti regionali per le identità verificate di Amazon SES

In-Region solo

I tuoi pool di utenti possono inviare messaggi e-mail con identità verificate nello Regione AWS stesso gruppo di utenti. Nella configurazione e-mail predefinita senza un indirizzo `FROM` e-mail personalizzato, Amazon Cognito utilizza un'identità `no-reply@verificationemail.com` verificata nella stessa regione.

Compatibile con le versioni precedenti

I tuoi pool di utenti possono inviare messaggi e-mail con identità verificate nella stessa Regione AWS o in una delle seguenti regioni alternative:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (Oregon)
- Europa (Irlanda)

Questa funzionalità supporta la continuità per le risorse del pool di utenti che potresti aver creato per soddisfare i requisiti di Amazon Cognito al momento del lancio del servizio. I pool di utenti di quel periodo potevano inviare messaggi e-mail con identità verificate solo in un numero limitato di. Regioni AWS Nella configurazione e-mail predefinita senza un indirizzo `FROM` e-mail personalizzato, Amazon Cognito utilizza un'identità `no-reply@verificationemail.com` verificata nella stessa regione.

Regione alternativa

I tuoi pool di utenti possono inviare messaggi e-mail con identità verificate in un'alternativa Regione AWS esterna alla regione del pool di utenti. Questa configurazione si verifica quando Amazon SES non è disponibile in una regione in cui è disponibile Amazon Cognito.

La politica di autorizzazione all'invio di Amazon SES per la tua identità verificata nella regione alternativa deve considerare attendibile il responsabile del servizio Amazon Cognito della regione di origine. Per ulteriori informazioni, consulta Per concedere le autorizzazioni per utilizzare la configurazione e-mail predefinita.

In alcune di queste regioni, Amazon Cognito divide i messaggi e-mail tra due regioni alternative per la configurazione e-mail predefinita di. `COGNITO_DEFAULT` In questi casi, per utilizzare un indirizzo `FROM` e-mail personalizzato, la politica di autorizzazione all'invio di Amazon SES per la tua identità verificata in ciascuna regione alternativa deve considerare attendibile il responsabile del servizio Amazon Cognito della regione di origine. Per ulteriori informazioni, consulta Per concedere le autorizzazioni per utilizzare la configurazione e-mail predefinita. Con la configurazione e-mail di Amazon SES `DEVELOPER` in queste regioni, devi utilizzare un'identità verificata nella prima regione elencata e configurarla in modo che si fidi del principale servizio Amazon Cognito nella regione del pool di utenti. Ad esempio, in un pool di utenti in Medio Oriente (Emirati Arabi Uniti), configura un'identità verificata in Europa (Francoforte) da considerare attendibile. `cognito-idp.me-central-1.amazonaws.com` Nella configurazione e-mail predefinita senza un indirizzo `FROM` e-mail personalizzato, Amazon Cognito utilizza un'identità `no-reply@verificationemail.com` verificata in ogni regione.

> **Note**
>
> Nella seguente combinazione di condizioni, devi specificare il `SourceArn` parametro EmailConfigurationcon un carattere jolly nell'elemento Regione, nel formato. `arn:${Partition}:ses:*:${Account}:identity/${IdentityName}` Ciò consente al tuo pool di utenti di inviare messaggi e-mail con identità verificate identiche in entrambi i tuoi Account AWS . Regioni AWS
>
> - Il tuo EmailSendingAccount è. `COGNITO_DEFAULT`
> - Vuoi usare un `FROM` indirizzo personalizzato.
> - Il tuo pool di utenti invia e-mail in una regione alternativa.

- Il tuo pool di utenti ha una seconda regione [1] alternativa specificata nella tabella delle regioni supportate da Amazon SES che segue.

Se crei un pool di utenti in modo programmatico, con un SDK AWS , l'API o la CLI di Amazon Cognito, oppure AWS CDK AWS CloudFormation, il tuo pool di utenti invia messaggi e-mail con l'identità Amazon `SourceArn` SES specificata dal parametro per il tuo pool di utenti. EmailConfiguration L'identità Amazon SES deve occupare un indirizzo supportato Regione AWS. Se il tuo `EmailSendingAccount` è `COGNITO_DEFAULT` e non specific un parametro`SourceArn`, Amazon Cognito invia messaggi e-mail da `no-reply@verificationemail.com` utilizzando risorse nella regione in cui hai creato il bacino d'utenza.

La tabella seguente mostra Regioni AWS dove è possibile utilizzare le identità Amazon SES con Amazon Cognito.

Regione del pool di utenti	Opzione Regione	Regioni supportate da Amazon SES
Stati Uniti orientali (Virginia settentrionale)	Compatibile con le versioni precedenti	Stati Uniti occidentali (Oregon), Stati Uniti orientali (Virginia settentrionale), Europa (Irlanda).
Stati Uniti orientali (Ohio)	Compatibile con le versioni precedenti	Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda).
Stati Uniti occidentali (California settentrionale)	In-Region solo	Stati Uniti occidentali (California settentrionale)
Stati Uniti occidentali (Oregon)	Compatibile con le versioni precedenti	Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda).
Canada (Centrale)	Compatibile con le versioni precedenti	Canada (Centrale), Stati Uniti orientali (Virginia settentri

Regione del pool di utenti	Opzione Regione	Regioni supportate da Amazon SES
		onale), Stati Uniti occidentali (Oregon), Europa (Irlanda).
Canada occidentale (Calgary)	Regione alternativa	Canada (Centrale), Stati Uniti occidentali (California settentrionale) [1]
Messico (centrale)	Regione alternativa	Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon) [1]
Asia Pacifico (Tokyo)	Compatibile con le versioni precedenti	Asia Pacifico (Tokyo), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Asia Pacifico (Hong Kong)	Regione alternativa	Asia Pacifico (Singapore), Asia Pacifico (Tokyo) [1]
Asia Pacifico (Seul)	Compatibile con le versioni precedenti	Asia Pacifico (Seoul), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Asia Pacifico (Malesia)	Regione alternativa	Asia Pacifico (Sydney), Asia Pacifico (Singapore) [1]
Asia Pacifico (Thailandia)	Regione alternativa	Asia Pacifico (Singapore), Asia Pacifico (Mumbai) [1]
Asia Pacifico (Mumbai)	Compatibile con le versioni precedenti	Asia Pacifico (Mumbai), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon)

Regione del pool di utenti	Opzione Regione	Regioni supportate da Amazon SES
Asia Pacifico (Hyderabad)	Regione alternativa	Asia Pacifico (Mumbai), Asia Pacifico (Singapore) [1]
Asia Pacifico (Singapore)	Compatibile con le versioni precedenti	Asia Pacifico (Singapore), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Asia Pacifico (Sydney)	Compatibile con le versioni precedenti	Asia Pacifico (Sydney), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Asia Pacifico (Osaka)	In-Region solo	Asia Pacifico (Osaka)
Asia Pacifico (Giacarta)	In-Region solo	Asia Pacifico (Giacarta)
Asia Pacifico (Melbourne)	Regione alternativa	Asia Pacifico (Sydney), Asia Pacifico (Singapore) [1]
Europa (Irlanda)	Compatibile con le versioni precedenti	Stati Uniti occidentali (Oregon), Stati Uniti orientali (Virginia settentrionale), Europa (Irlanda).
Europa (Londra)	Compatibile con le versioni precedenti	Europa (Londra), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda).
Europa (Parigi)	In-Region solo	Europa (Parigi)

Regione del pool di utenti	Opzione Regione	Regioni supportate da Amazon SES
Europa (Francoforte)	Compatibile con le versioni precedenti	Europa (Francoforte), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Europa (Zurigo)	Regione alternativa	Europa (Francoforte), Europa (Londra) [1]
Europa (Stoccolma)	In-Region solo	Europa (Stoccolma)
Europa (Milano)	In-Region solo	Europa (Milano)
Europa (Spagna)	Regione alternativa	Europa (Parigi), Europa (Stoccolma) [1]
Medio Oriente (Bahrein)	In-Region solo	Medio Oriente (Bahrein)
Medio Oriente (Emirati Arabi Uniti)	Regione alternativa	Europa (Francoforte), Europa (Londra) [1]
Sud America (San Paolo)	In-Region solo	Sud America (San Paolo)
Israele (Tel Aviv)	In-Region solo	Israele (Tel Aviv)
Africa (Città del Capo)	In-Region solo	Africa (Città del Capo)

[1] Utilizzato nei pool di utenti con la configurazione e-mail predefinita. Amazon Cognito distribuisce messaggi e-mail tra identità verificate con lo stesso indirizzo e-mail in ogni regione. Per utilizzare un `FROM` indirizzo personalizzato, configuralo `EmailConfiguration` con un `SourceArn` parametro nel formato. `arn:${Partition}:ses:*:${Account}:identity/${IdentityName}`

Configurazione delle e-mail per il bacino d'utenza

Completa le fasi seguenti per configurare le impostazioni e-mail per il bacino d'utenza. A seconda delle impostazioni che usi, potresti aver bisogno di autorizzazioni IAM in Amazon SES, AWS Identity and Access Management (IAM) e Amazon Cognito.

> **Note**
>
> Le risorse da te create in questi passaggi non possono essere condivise tra Account AWS. Ad esempio, non puoi configurare un bacino d'utenza in un account e poi usarlo con un indirizzo e-mail Amazon SES in un account diverso. Se utilizzi Amazon Cognito in più di un account, ripeti questi passaggi per ognuno di essi.

Fase 1: Verifica dell'indirizzo e-mail o del dominio con Amazon SES

Prima di configurare il bacino d'utenza, se desideri eseguire una delle operazioni seguenti devi verificare uno o più indirizzi e-mail con Amazon SES:

- Utilizzare l'indirizzo e-mail personale come indirizzo FROM
- Utilizzo della configurazione di Amazon SES per gestire la distribuzione delle e-mail

Verificando l'indirizzo e-mail, confermi di esserne il proprietario, il che impedisce l'utilizzo non autorizzato.

Per ulteriori informazioni sulla verifica di un indirizzo e-mail con Amazon SES, consulta Verifica di un indirizzo e-mail nella Guida per gli sviluppatori di Amazon Simple Email Service. Per ulteriori informazioni sulla verifica di un dominio con Amazon SES,, consulta la sezione Verifying domains (verifica dei domini).

Fase 2: Spostamento dell'account all'esterno della sandbox di Amazon SES

Ometti questo passaggio se utilizzi la configurazione e-mail predefinita di Amazon Cognito.

Quando usi Amazon SES per la prima volta in qualsiasi regione Regione AWS, ti colloca Account AWS nella sandbox Amazon SES per quella regione. Amazon SES utilizza la sandbox per prevenire frodi e usi illeciti. Se utilizzi la configurazione di Amazon SES per gestire la consegna delle e-mail, perché Amazon Cognito possa inviare e-mail ai tuoi utenti devi spostare il tuo Account AWS all'esterno della sandbox.

Nella sandbox, Amazon SES impone delle limitazioni al numero di e-mail che si possono inviare e agli indirizzi raggiungibili. Puoi inviare e-mail solo a indirizzi e domini verificati con Amazon SES oppure agli indirizzi del simulatore di mailbox di Amazon SES. Mentre Account AWS resti nella sandbox, non utilizzare la configurazione di Amazon SES per applicazioni in produzione. In questo caso, Amazon Cognito non può inviare messaggi agli indirizzi e-mail degli utenti.

Per rimuovere il tuo Account AWS dalla sandbox, consulta Uscire dalla sandbox di Amazon SES nella Amazon Simple Email Service Developer Guide.

Fase 3: Concessione delle autorizzazioni e-mail ad Amazon Cognito

Prima che Amazon Cognito possa inviare e-mail ai tuoi utenti, è possibile che tu debba concedergli autorizzazioni specifiche. Le autorizzazioni concesse e il processo che utilizzi per concederle dipendono dal fatto che tu stia utilizzando la configurazione e-mail predefinita o la configurazione Amazon SES.

Per concedere le autorizzazioni per utilizzare la configurazione e-mail predefinita

Completa questo passaggio solo se configuri il tuo pool di utenti su Invia email con Cognito o se hai impostato su`EmailSendingAccount`. `COGNITO_DEFAULT`

Con la configurazione e-mail predefinita, il tuo pool di utenti può inviare messaggi e-mail con uno dei seguenti indirizzi.

- L'indirizzo predefinitono-`reply@verificationemail.com`.
- Un indirizzo FROM personalizzato dai tuoi indirizzi e-mail o domini verificati in Amazon SES.

Se utilizzi un indirizzo personalizzato, Amazon Cognito ha bisogno di ulteriori autorizzazioni per inviare e-mail agli utenti da questo indirizzo. Queste autorizzazioni sono concesse da una politica di autorizzazione all'invio per l'indirizzo o il dominio in Amazon SES. Se si utilizza la console Amazon Cognito per aggiungere un indirizzo personalizzato al bacino d'utenza, la policy viene automaticamente collegata all'indirizzo e-mail verificato di Amazon SES. Tuttavia, se configuri il tuo pool di utenti al di fuori della console, ad esempio utilizzando l' AWS CLI API Amazon Cognito, devi allegare la policy utilizzando la console Amazon SES o l'PutIdentityPolicyAPI.

> **Note**
>
> È possibile configurare un indirizzo FROM solo in un dominio verificato utilizzando la AWS CLI o l'API Amazon Cognito.

Una policy di autorizzazione all'invio consente o nega l'accesso in base alle risorse dell'account che utilizzano Amazon Cognito per richiamare Amazon SES. Per ulteriori informazioni sulle policy basate su risorse, consulta Guida per l'utente di IAM. Puoi anche trovare esempi di policy basate su risorse nella Guida per gli sviluppatori di Amazon SES.

Example Policy di autorizzazione di invio

L'esempio seguente di policy di autorizzazione di invio garantisce ad Amazon Cognito la possibilità limitata di utilizzare un'identità verificata di Amazon SES. Amazon Cognito può inviare e-mail solo quando lo fa per conto sia del bacino d'utenza nella condizione `aws:SourceArn` sia dell'account nella condizione `aws:SourceAccount`.

Regions with Amazon SES

La tua politica di autorizzazione all'invio nella regione del pool di utenti o nella regione alternativa deve consentire al responsabile del servizio Amazon Cognito di inviare messaggi e-mail. Per ulteriori informazioni, consulta la tabella Regioni. Se la regione del pool di utenti corrisponde ad almeno un valore nella regione Amazon SES, configura la politica di autorizzazione all'invio con il servizio principale globale nell'esempio seguente.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "stmnt1234567891234",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "email.cognito-idp.amazonaws.com"
                ]
            },
            "Action": [
                "SES:SendEmail",
                "SES:SendRawEmail"
            ],
            "Resource": "arn:aws:ses:us-
east-1:111122223333:identity/support@example.com",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:cognito-idp:us-
east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
```

```
            }
        }
    ]
}
```

Opt-in Regions without Amazon SES

Amazon SES non è disponibile in tutti gli opt-in in Regioni AWS cui è disponibile Amazon Cognito. Il Medio Oriente (Emirati Arabi Uniti) ne è un esempio e può inviare e-mail con identità verificate solo in Europa (Francoforte) (). `eu-central-1` Nei pool di utenti con la configurazione e-mail predefinita, Amazon Cognito invia anche messaggi e-mail con un'identità verificata in ciascuna delle due regioni. Nel caso del Medio Oriente (Emirati Arabi Uniti), la regione aggiuntiva è l'Europa (Londra). È necessario aggiornare la politica di autorizzazione all'invio in entrambe le regioni.

La tua politica di autorizzazione all'invio in ciascuna delle regioni alternative deve consentire al responsabile del servizio Amazon Cognito nella regione di attivazione del pool di utenti di inviare messaggi e-mail. Per ulteriori informazioni, consulta la tabella delle regioni. Se la tua regione è contrassegnata come Regione alternativa, configura le politiche di autorizzazione all'invio con il responsabile del servizio regionale, come nell'esempio seguente. Sostituisci l'identificatore di regione di esempio *`me-central-1`* con l'ID regionale richiesto, se necessario.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cognito-idp.me-central-1.amazonaws.com"
                ]
            },
            "Action": [
                "SES:SendEmail",
                "SES:SendRawEmail"
            ],
            "Resource": "arn:aws:ses:us-
east-1:111122223333:identity/support@example.com",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:cognito-idp:us-
east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        }
    ]
}
```

Per ulteriori informazioni sulla sintassi della policy, consulta la sezione Policy di autorizzazione di invio di Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

Per altri esempi, consulta la sezione Amazon SES sending authorization policy examples (Esempi di policy di autorizzazione di invio di Amazon SES) nella Guida per gli sviluppatori di Amazon Simple Email Service.

Come concedere le autorizzazioni per utilizzare la configurazione Amazon SES

Se configuri il bacino d'utenza in modo da utilizzare la configurazione di Amazon SES, Amazon Cognito avrò bisogno di ulteriori autorizzazioni per richiamare Amazon SES per tuo conto quando invia e-mail agli utenti. Questa autorizzazione è concessa con il servizio IAM.

Quando configuri il bacino d'utenza con questa opzione, Amazon Cognito crea un ruolo collegato al servizio, che è un tipo di ruolo IAM, nel tuo Account AWS. Questo ruolo contiene le autorizzazioni che permettono ad Amazon Cognito di accedere ad Amazon SES e di inviare messaggi e-mail con il tuo indirizzo.

Amazon Cognito crea il tuo ruolo collegato al servizio con AWS le credenziali della sessione utente che imposta la configurazione. Le autorizzazioni IAM di questa sessione devono includere l'azione `iam:CreateServiceLinkedRole`. Per ulteriori informazioni sulle autorizzazioni in IAM, consulta Gestione degli accessi alle AWS risorse nella IAM User Guide.

Per ulteriori informazioni sul ruolo collegato al servizio creato da Amazon Cognito, consulta Utilizzo di ruoli collegati ai servizi per Amazon Cognito.

Fase 4: configurazione del bacino d'utenza

Completa la procedura seguente se vuoi configurare il bacino d'utenza con uno dei seguenti elementi:

- Un indirizzo FROM personalizzato che compare come mittente dell'e-mail
- Un REPLY-TO indirizzo personalizzato che riceve i messaggi che gli utenti inviano al tuo indirizzo FROM
- La tua configurazione di Amazon SES

> **Note**
>
> Se la tua identità verificata è un indirizzo e-mail, Amazon Cognito imposta tale indirizzo e-mail come FROM e indirizzo REPLY-TO e-mail per impostazione predefinita. Tuttavia, se la tua identità verificata è un dominio, devi fornire un valore per l'indirizzo e-mail FROM.

Ometti questa procedura se desideri utilizzare la configurazione e l'indirizzo e-mail predefiniti di Amazon Cognito.

Per configurare il bacino d'utenza per utilizzare un indirizzo e-mail personalizzato

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Metodi di autenticazione, individua la configurazione dell'e-mail, scegli Modifica.
5. Nella pagina Edit email configuration (Modifica della configurazione e-mail) seleziona l'opzione Send email from Amazon SES (Invia e-mail da Amazon SES) o Send email with Amazon Cognito (Invia e-mail con Amazon Cognito). Puoi personalizzare la Regione SES, il Set di configurazione e il nome del mittente FROM (DA) solo selezionando l'opzione Send email from Amazon SES (Invia e-mail da Amazon SES).
6. Per utilizzare un indirizzo FROM (DA) personalizzato, completa la seguente procedura:
   a. In SES Region (Regione SES), scegli la regione che contiene l'indirizzo e-mail verificato.
   b. In FROM email address (Indirizzo e-mail FROM, (DA)), selezionare l'indirizzo e-mail. Usa un indirizzo e-mail che hai verificato con Amazon SES.

c. (Facoltativo) Alla voce Configuration set (Set di configurazione), scegli un set di configurazione da utilizzare con Amazon SES. La creazione e il salvataggio di questa modifica genera un ruolo collegato al servizio.

d. (Facoltativo) Alla voce Indirizzo del mittente FROM (DA), inserisci un indirizzo e-mail. Puoi fornire solo il tuo indirizzo e-mail o il tuo indirizzo e-mail e un nome descrittivo nel formato `Jane Doe <janedoe@example.com>`.

e. (Facoltativo) In Indirizzo REPLY-TO e-mail, inserisci l'indirizzo e-mail a cui desideri ricevere i messaggi che gli utenti inviano al tuo indirizzo FROM.

7. Scegli Save changes (Salva modifiche).

Argomenti correlati

- Personalizzazione dei messaggi di verifica delle e-mail
- Personalizzazione dei messaggi di invito degli utenti

Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito

Alcuni eventi di Amazon Cognito per il tuo bacino d'utenza potrebbero causare l'invio di SMS da parte di Amazon Cognito ai tuoi utenti. Ad esempio, se configuri il bacino d'utenza in modo da richiedere la verifica tramite telefono, Amazon Cognito invia un SMS quando un utente registra un nuovo account nell'app o quando reimposta la password. A seconda dell'azione che attiva il messaggio SMS, il messaggio può contenere un codice di verifica, una password temporanea o un messaggio di benvenuto.

Amazon Cognito può inviare messaggi SMS con una delle seguenti configurazioni:

- Amazon Simple Notification Service (Amazon SNS) — Amazon Cognito chiama Amazon SNS, che indirizza i messaggi SMS verso. AWS End User Messaging SMS
- AWS End User Messaging SMS diretto: Amazon Cognito chiama AWS End User Messaging SMS direttamente, senza Amazon SNS come intermediario. Puoi utilizzare questa opzione nella maggior parte dei paesi in Regioni AWS cui è disponibile Amazon Cognito. Per le eccezioni, consulta. AWS End User Messaging SMS percorso diretto: configurazione della regione

Un pool di utenti utilizza una configurazione SMS alla volta. Amazon SNS e le configurazioni AWS End User Messaging SMS dirette si escludono a vicenda. Quando imposti una

configurazione, Amazon Cognito cancella l'altra. Questo vale per e SetUserPoolMfaConfigper CreateUserPoolUpdateUserPoolle operazioni.

Amazon SNS SMS è disponibile nelle versioni Regioni AWS elencate in. Configurazione della regione del percorso Amazon SNS Per inviare messaggi SMS da un pool di utenti a qualsiasi altro Regione AWS, usa AWS End User Messaging SMS direttamente.

Se invii un messaggio di testo tramite Amazon Cognito per la prima volta, ti AWS End User Messaging SMS colloca in un ambiente sandbox. Nell'ambiente sandbox è possibile testare le applicazioni per gli SMS. Nella sandbox, puoi inviare messaggi a un numero limitato di numeri di telefono di destinazione verificati se disponi di un'identità di origine o simulare l'invio di messaggi in caso contrario.

> **Note**
>
> Nel novembre 2024, ha AWS sostituito la messaggistica SMS di Amazon SNS con. AWS End User Messaging SMS Quando utilizzi il percorso di configurazione di Amazon SNS, i pool di utenti avviano i messaggi SMS con l'operazione Amazon SNS Publish, che è un pass-through per. AWS End User Messaging SMS Per questo percorso, devi configurare le autorizzazioni per. `sns:Publish` Quando utilizzi il percorso di AWS End User Messaging SMS configurazione diretto, Amazon Cognito richiama l' AWS End User Messaging SMS `SendTextMessage`operazione e devi configurare le autorizzazioni per. `sms-voice:SendTextMessage`

AWS End User Messaging SMS costi per i messaggi di testo SMS. Per ulteriori informazioni, consultare Prezzi di AWS End User Messaging SMS.

Amazon Cognito invia messaggi SMS agli utenti con un codice che possono inserire. Nella tabella seguente vengono mostrati gli eventi che possono generare un messaggio SMS.

Opzioni relative ai messaggi

Attività	Operazione API	Opzioni di distribuzione	Opzioni di formato	Personalizzabile	Modello di messaggio
Password dimenticata	ForgotPassword, AdminResetUserPassword	E-mail, SMS	code	Sì	Messaggio di verifica

Attività	Operazione API	Opzioni di distribuzione	Opzioni di formato	Personalizzabile	Modello di messaggio
Invito	AdminCreateUser	E-mail, SMS	code	Sì	Messaggio di invito
Self-registration	SignUp, ResendConfirmationCode	E-mail, SMS	codice, link	Sì	messaggio di verifica
Verifica dell'indirizzo e-mail o del numero di telefono	UpdateUserAttributes, AdminUpdateUserAttributes, GetUserAttributeVerificationCode	E-mail, SMS	code	Sì	Messaggio di verifica
Multi-factor autenticazione (MFA)	AdminInitiateAuth, InitiateAuth	Email[1], SMS, app di autenticazione	code	Sì[2]	Messaggio MFA
One-time autenticazione tramite password (OTP)	AdminInitiateAuth, InitiateAuth	E-mail[1], SMS	code	Sì	Messaggio MFA [3]

[1] Richiede un piano di funzionalità Essentials o superiore e la configurazione e-mail di Amazon SES.

[2] Per SMS e messaggi e-mail.

[3] È possibile personalizzare il modello di messaggio MFA solo quando l'MFA è obbligatoria o facoltativa nel pool di utenti. Quando la MFA è inattiva, Amazon Cognito invia password monouso con il modello predefinito.

AWS End User Messaging SMS addebiti per i messaggi SMS. Per ulteriori informazioni, consultare Prezzi di AWS End User Messaging SMS.

Per ulteriori informazioni su MFA, consulta MFA per SMS e messaggi e-mail.

Amazon Cognito potrebbe impedire la consegna di messaggi e-mail o SMS aggiuntivi verso un'unica destinazione in un breve periodo di tempo. Se ritieni che il tuo pool di utenti sia interessato, configura e rivedi i log per individuare eventuali errori di recapito dei messaggi, quindi contatta il team del tuo account.

Scelta tra Amazon SNS e AWS End User Messaging SMS

I pool di utenti di Amazon Cognito possono inviare messaggi SMS attraverso due percorsi: Amazon AWS End User Messaging SMS SNS o direttamente. Per i nuovi pool di utenti, usa il AWS End User Messaging SMS percorso diretto. Amazon SNS SMS è disponibile solo in un set fisso di. Regioni AWS Il AWS End User Messaging SMS percorso diretto supporta regioni aggiuntive ed espone funzionalità che il percorso Amazon SNS non offre. Entrambi i percorsi consentono di recapitare i messaggi. AWS End User Messaging SMS

La tabella seguente confronta i due percorsi di configurazione SMS.

Funzionalità	Percorso Amazon SNS	AWS End User Messaging SMS percorso diretto
Disponibilità nelle regioni	Disponibile nella Regioni AWS tabella delle regioni dei percorsi di Amazon SNS	Disponibile nella maggior parte dei Regioni AWS paesi in cui è disponibile Amazon Cognito. Sono elencate le regioni in cui è disponibile solo il percorso Amazon SNS. AWS End User Messaging SMS percorso diretto: configurazione della regione
Integrazione	Amazon Cognito chiama l'operazione Amazon SNS Publish, che passa a AWS End User Messaging SMS	Amazon Cognito chiama direttamente l'operazione AWS End User Messaging SMS SendTextMessage
Autorizzazione IAM	`sns:Publish`	`sms-voice:SendTextMessage`

Funzionalità	Percorso Amazon SNS	AWS End User Messaging SMS percorso diretto
Set di configurazione	Non configurabile tramite Amazon Cognito	Imposta il `ConfigurationSetName` campo per la registrazione degli eventi e le destinazioni
Identità di origine	Non configurabile tramite Amazon Cognito	Imposta il campo `OriginationIdentity`
Cross-Region invio	Imposta il `SnsRegion` campo	Non supportata
Registrazione DLT in India	Non configurabile tramite Amazon Cognito	Imposta i campi and `InEntityId` `InTemplateId`
Protezione antifrode (Protect)	Account-default si applica la configurazione protect	Account-default si applica. È inoltre possibile applicare una configurazione di protezione a un caso d'uso specifico tramite un set di configurazione.

> **Note**
>
> Una configurazione AWS End User Messaging SMS Protect predefinita si applica a tutti i messaggi, inclusi quelli provenienti da Amazon Cognito. Per applicare una configurazione di protezione a un caso d'uso specifico, collegala a un set di configurazione e utilizza il AWS End User Messaging SMS percorso diretto. Per ulteriori informazioni, consulta Protezione dalle frodi in AWS End User Messaging SMS.

Configurazione SMS nell'API dei pool di utenti di Amazon Cognito

L'SmsConfigurationTypeoggetto nell'API dei pool di utenti di Amazon Cognito contiene la configurazione SMS per il tuo pool di utenti. Questo oggetto ha i seguenti percorsi di configurazione:

Percorso Amazon SNS

Impostato `SnsCallerArn` sull'ARN di un ruolo IAM che concede ad Amazon Cognito l'autorizzazione alla chiamata. `sns:Publish` Opzionalmente imposta e. `SnsRegion` `ExternalId` Il `SnsCallerArn` parametro è facoltativo quando si utilizza il percorso AWS End User Messaging SMS diretto.

AWS End User Messaging SMS percorso diretto

Imposta il tipo di `EumsSms` oggetto`EumsSmsConfigurationType`. Questo oggetto ha i seguenti membri:

- `CallerArn`(Obbligatorio): l'ARN di un ruolo IAM che concede ad Amazon Cognito l'autorizzazione alla chiamata. `sms-voice:SendTextMessage`
- `ExternalId`(Facoltativo): l'ID esterno per la policy di fiducia dei ruoli IAM. Questo valore fornisce una prevenzione confusa.
- `OriginationIdentity`(Facoltativo): il numero di telefono, il pool o l'ID mittente da utilizzare per l'invio.
- `ConfigurationSetName`(Facoltativo): il set AWS End User Messaging SMS di configurazione da utilizzare.
- `InEntityId`(Facoltativo): l'ID di registrazione dell'entità ottenuto dalla registrazione DLT in India. Obbligatorio per l'invio a destinatari in India.
- `InTemplateId`(Facoltativo): l'ID di registrazione del modello di registrazione DLT in India. Obbligatorio per l'invio a destinatari in India.
- `Region`(Facoltativo): Regione AWS le tue AWS End User Messaging SMS risorse. Per ulteriori informazioni, consulta Seleziona Regione AWS per i messaggi SMS.

Per ulteriori informazioni, consulta SmsConfigurationTypeAmazon Cognito User Pools API Reference.

Best practice

A causa del volume di traffico SMS indesiderato in tutto il mondo, alcuni governi impongono barriere tra mittenti e destinatari dei messaggi SMS. Quando utilizzi i messaggi SMS per l'autenticazione MFA e gli aggiornamenti utente, devi adottare misure aggiuntive per assicurarti che i messaggi vengano recapitati. È inoltre necessario monitorare SMS-message-related le normative nei paesi in cui risiedono gli utenti e mantenere aggiornata la configurazione dei messaggi SMS. Per ulteriori

informazioni, consulta le funzionalità e le limitazioni nazionali relative agli SMS e MMS nella Guida per l'AWS End User Messaging SMS utente.

L'uso di messaggi SMS per autenticare e verificare gli utenti non è una best practice di sicurezza. I numeri di telefono possono cambiare proprietario e potrebbero non rappresentare in modo affidabile un fattore di autenticazione MFA elementi che possiedi per gli utenti. Implementa invece l'autenticazione MFA TOTP nell'app o con il gestore dell'identità digitale (IdP) di terze parti. Puoi anche creare fattori di autenticazione personalizzati aggiuntivi con Trigger Lambda di richieste di autenticazione personalizzate.

Consulta i seguenti collegamenti per informazioni sulla protezione dell'architettura di recapito dei messaggi SMS.

- Riduci i rischi di frode nelle iscrizioni degli utenti e di invio di SMS con i pool di utenti di Amazon Cognito
- Difesa dal pompaggio di SMS: nuove AWS funzionalità per aiutare a combattere il traffico gonfiato artificialmente

Prima impostazione degli SMS nei bacini d'utenza di Amazon Cognito

Amazon Cognito utilizza Amazon SNS (e AWS End User Messaging SMS indirettamente) AWS End User Messaging SMS o direttamente per inviare messaggi SMS dai tuoi pool di utenti. Puoi utilizzare un Trigger Lambda del mittente di SMS personalizzato, se desideri servirti delle tue risorse per inviare messaggi SMS. La prima volta che configuri i messaggi di testo SMS in una particolare regione Regione AWS, ti AWS End User Messaging SMS inserisce Account AWS nella sandbox SMS per quella regione. AWS End User Messaging SMS utilizza la sandbox per prevenire frodi e abusi e per soddisfare i requisiti di conformità. Quando sei Account AWS nella sandbox, AWS End User Messaging SMS impone alcune restrizioni. Ad esempio, puoi inviare messaggi di testo a un massimo di 10 numeri di destinazione verificati se disponi di un'identità di origine oppure puoi simulare l'invio di messaggi senza un'identità di origine. Mentre Account AWS rimanete nella sandbox, non inviate messaggi SMS in fase di produzione. Quando sei nella sandbox, Amazon Cognito non può inviare messaggi ai numeri di telefono degli utenti.

Argomenti

- Prepara un ruolo IAM che Amazon Cognito possa utilizzare per inviare messaggi SMS AWS End User Messaging SMS
- Per configurare AWS End User Messaging SMS SMS diretti per un pool di utenti

- Seleziona Regione AWS per i messaggi SMS
- Ottenere un'identità di origine per l'invio di messaggi SMS a numeri di telefono USA
- Verifica di trovarti nella sandbox SMS
- Sposta il tuo account fuori dalla sandbox
- Usa i numeri del simulatore o i numeri di telefono verificati con AWS End User Messaging SMS
- Completamento della configurazione del pool di utenti in Amazon Cognito

Prepara un ruolo IAM che Amazon Cognito possa utilizzare per inviare messaggi SMS AWS End User Messaging SMS

Quando invii un SMS dal tuo pool di utenti, Amazon Cognito assume un ruolo IAM nel tuo account. Le autorizzazioni richieste dal ruolo dipendono dal percorso di configurazione SMS scelto:

- Percorso Amazon SNS: il ruolo deve concedere l'`sns:Publish`autorizzazione. Amazon Cognito utilizza questa autorizzazione per inviare messaggi SMS tramite Amazon SNS.
- AWS End User Messaging SMS percorso diretto: il ruolo deve concedere l'`sms-voice:SendTextMessage`autorizzazione. Amazon Cognito utilizza questa autorizzazione per chiamare AWS End User Messaging SMS direttamente.

Nella console Amazon Cognito, puoi impostare una selezione di ruoli IAM dal menu Metodi di autenticazione del tuo pool di utenti, sotto SMS o effettuare questa selezione durante la procedura guidata di creazione del pool di utenti.

Il seguente esempio di policy di attendibilità dei ruoli IAM garantisce al pool di utenti di Amazon Cognito una capacità limitata di assumere un ruolo IAM. Amazon Cognito può assumere il ruolo solo quando soddisfa le seguenti condizioni:

- L'operazione assume-role avviene per conto del pool di utenti nella condizione. `aws:SourceArn`
- L'operazione Assume-role avviene per conto di un pool di utenti secondo quanto stabilito dalla condizione. Account AWS `aws:SourceAccount`
- L'operazione assume-role include l'ID esterno nella condizione. `sts:externalId`

Queste condizioni relative alla politica di fiducia si applicano sia al percorso Amazon SNS che al percorso AWS End User Messaging SMS diretto.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "cognito-idp.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:cognito-idp:us-
west-2:111122223333:userpool/us-west-2_EXAMPLE"
                }
            }
        }
    ]
}
```

Puoi specificare un valore esatto per l'ARN del pool di utenti o un ARN con caratteri jolly nel valore della condizione `aws:SourceArn`. Cerca gli ARN dei tuoi pool di utenti in Console di gestione AWS o con una richiesta DescribeUserPoolAPI.

Per inviare messaggi SMS per l'autenticazione a più fattori, la tua policy di fiducia dei ruoli IAM deve avere una `sts:ExternalId` condizione. Il valore di questa condizione deve corrispondere alla `ExternalId` proprietà del pool SmsConfigurationdi utenti. Quando crei un ruolo IAM durante il processo di creazione del pool di utenti nella console Amazon Cognito, Amazon Cognito configura l'ID esterno per te nel ruolo e nelle impostazioni del pool di utenti. Questo non è vero quando utilizzi un ruolo IAM esistente.

È necessario aggiornare il `ExternalId` parametro del pool di utenti in una richiesta UpdateUserPoolAPI e aggiornare la policy di fiducia dei ruoli IAM con una `sts:externalId` condizione con lo stesso valore. Per informazioni su come utilizzare l'API per aggiornare un pool di

utenti in modo da preservare la configurazione originale, consultaAggiornamento della configurazione del pool di utenti e del client dell'app.

Per ulteriori informazioni sui ruoli IAM e le policy di attendibilità, consulta Termini e concetti dei ruoli nella Guida per l'utente di AWS Identity and Access Management .

Per configurare AWS End User Messaging SMS SMS diretti per un pool di utenti

Utilizza la seguente procedura per configurare il tuo pool di utenti per l'invio diretto di messaggi SMS tramite AWS End User Messaging SMS. Questa configurazione utilizza l'`EumsSmsoggetto` in SmsConfigurationType.

Per configurare AWS End User Messaging SMS messaggistica SMS diretta

1. Crea o identifica un'identità di AWS End User Messaging SMS origine nella Regione AWS cartella da cui desideri inviare. Un'identità di origine può essere un numero di telefono, un ID mittente o un pool. Puoi anche creare un set di AWS End User Messaging SMS configurazione per applicare le regole di invio. Per istruzioni sulla richiesta di un numero di telefono, consulta Richiedere un numero di telefono nella Guida per l'AWS End User Messaging SMS utente.
2. Crea un ruolo IAM che Amazon Cognito possa assumere di chiamare. AWS End User Messaging SMS La politica di fiducia per questo ruolo è la stessa del percorso Amazon SNS: deve includere `aws:SourceArn` e `sts:ExternalId` condizioni. `aws:SourceAccount` Per i dettagli sulla politica di fiducia, consultaPrepara un ruolo IAM che Amazon Cognito possa utilizzare per inviare messaggi SMS AWS End User Messaging SMS.
3. Allega la seguente politica di identità (autorizzazioni) al ruolo. Questa politica concede ad Amazon Cognito l'autorizzazione a `sms-voice:SendTextMessage` richiamare l'identità di origine da cui invia il pool di utenti. Definisci l'`Resource`elemento in base all'ARN del tuo numero di telefono, pool di telefoni o ID mittente come pratica con privilegi minimi. Sostituisci l'ARN di esempio con l'ARN della tua identità di origine.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sms-voice:SendTextMessage"
            ],
            "Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/
phone-a1b2c3d4e5f6EXAMPLE"
```

```
        }
    ]
}
```

L'ARN dell'identità di origine utilizza uno dei seguenti formati, a seconda del tipo di identità:

- Numero di telefono: `arn:`*`partition`*`:sms-voice:`*`region`*`:`*`account`*`:phone-number/`*`phoneNumberId`*
- Pool telefonico — `arn:`*`partition`*`:sms-voice:`*`region`*`:`*`account`*`:pool/`*`poolId`*
- ID mittente: `arn:`*`partition`*`:sms-voice:`*`region`*`:`*`account`*`:sender-id/`*`senderId`*`/`*`isoCountryCode`*

> **Important**
>
> Quando la policy definisce l'ambito dell'`Resource`elemento in base a un ARN di identità di origine, imposta il `OriginationIdentity` campo sullo stesso ARN, ad esempio. `arn:aws:sms-voice:us-east-1:123456789012:phone-number/phone-a1b2c3d4e5f6EXAMPLE` Un numero di telefono in E.164 formato (ad esempio`+12065550100`) non corrisponde alla politica prevista, quindi la richiesta non è autorizzata e Amazon Cognito non può inviare messaggi SMS.

4. Se invii messaggi SMS a destinatari in India, registra un ID entità DLT e un ID modello su un portale Distributed Ledger Technology (DLT). Fornisci l'ID dell'entità `InEntityId` e l'ID del modello come `InTemplateId` nella configurazione. `EumsSms` Per ulteriori informazioni sulla registrazione DLT in India, consulta Requisiti speciali per l'invio di messaggi SMS a destinatari in India nella Guida per l'AWS End User Messaging SMS utente.
5. Verifica la tua configurazione chiamando l'operazione AWS End User Messaging SMS SendTextMessageAPI direttamente con l'identità di origine e il ruolo IAM che hai configurato. Conferma che il messaggio di prova sia stato recapitato prima di assegnare la configurazione al tuo pool di utenti.
6. Imposta l'`EumsSms`oggetto nel tuo pool di utenti. È possibile impostare questa configurazione in una SetUserPoolMfaConfigrichiesta CreateUserPoolUpdateUserPool, o. Il parametro `CallerArn` è obbligatorio. L'esempio seguente mostra la `EumsSms` configurazione minima:

```
"EumsSms": {
    "CallerArn": "arn:aws:iam::123456789012:role/service-role/CognitoEumsSmsRole",
    "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

```
}
```

Per inviare da un'identità di origine specifica o utilizzare un set di configurazione, includete i `ConfigurationSetName` parametri `OriginationIdentity` and.

Autorizzazioni IAM per AWS End User Messaging SMS SMS diretti

Il ruolo di IAM in `CallerArn` richiede una politica di fiducia che consenta al responsabile del `cognito-idp.amazonaws.com` servizio di assumerlo. Utilizza la stessa struttura di policy di fiducia del percorso Amazon SNS (vediPrepara un ruolo IAM che Amazon Cognito possa utilizzare per inviare messaggi SMS AWS End User Messaging SMS). Il ruolo richiede anche una politica di autorizzazioni che conceda un `sms-voice:SendTextMessage` ARN specifico all'identità di origine (vedere la procedura di configurazione in). Per configurare AWS End User Messaging SMS SMS diretti per un pool di utenti

Seleziona Regione AWS per i messaggi SMS

> **Note**
>
> I messaggi SMS in AWS entrata sono ora gestiti in AWS End User Messaging SMS.

Configurazione della regione del percorso Amazon SNS

In alcuni Regioni AWS, puoi scegliere la regione che contiene le risorse Amazon SNS che desideri utilizzare per i messaggi SMS di Amazon Cognito. Regione AWS Ovunque sia disponibile Amazon Cognito, ad eccezione dell'Asia Pacifico (Seoul), puoi utilizzare le risorse Amazon SNS nel luogo in Regione AWS cui hai creato il tuo pool di utenti. Per rendere l'invio di messaggi SMS più veloce e affidabile quando puoi scegliere tra più regioni, usa le risorse Amazon SNS nella stessa regione del tuo bacino d'utenza.

Scegli una regione per le risorse SMS nel passaggio Configure message delivery (Configura invio di messaggi) della procedura guidata per la configurazione del nuovo bacino d'utenza. Puoi anche scegliere Modifica in SMS nel menu Metodi di autenticazione di un pool di utenti esistente.

Al momento del lancio, per alcuni Regioni AWS, Amazon Cognito inviava messaggi SMS con risorse Amazon SNS in una regione alternativa. Per impostare la tua regione preferita, usa il `SnsRegion` parametro dell'SmsConfigurationTypeoggetto per il tuo pool di utenti. Quando crei a livello di programmazione una risorsa del bacino d'utenza di Amazon Cognito in una Regione di Amazon

Cognito inclusa nella tabella seguente e non fornisci un parametro `SnsRegion`, il bacino d'utenza può inviare messaggi SMS utilizzando le risorse Amazon SNS di una Regione di Amazon SNS legacy.

I pool di utenti di Amazon Cognito in Asia Pacifico (Seoul) Regione AWS devono utilizzare la configurazione Amazon SNS nella regione Asia Pacifico (Tokyo).

Amazon SNS (tramite AWS End User Messaging SMS) imposta la quota di spesa per tutti i nuovi account a 1,00 USD (USD) al mese. Potresti aver aumentato il limite di spesa in una Regione AWS che utilizzi con Amazon Cognito. Prima di modificare Regione AWS i messaggi SMS di Amazon SNS, apri una richiesta di aumento della quota nel AWS Support Center per aumentare il limite nella nuova regione. Per ulteriori informazioni, consulta Passare dalla sandbox AWS End User Messaging SMS MMS and Voice alla produzione nella Guida per l'AWS End User Messaging SMS utente.

Puoi inviare messaggi SMS per qualsiasi regione di Amazon Cognito nella tabella seguente con AWS End User Messaging SMS risorse nella regione di messaggi SMS corrispondente.

Regione di Amazon Cognito	Regione dei messaggi SMS
Stati Uniti orientali (Ohio)	Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale)
Stati Uniti orientali (Virginia settentrionale)	Stati Uniti orientali (Virginia settentrionale)
Stati Uniti occidentali (California settentrionale)	Stati Uniti occidentali (California settentrionale)
Stati Uniti occidentali (Oregon)	Stati Uniti occidentali (Oregon)
Canada (Centrale)	Canada (Centrale), Stati Uniti orientali (Virginia settentrionale)
Canada occidentale (Calgary)	Canada occidentale (Calgary)
Messico (centrale)	Messico (centrale)
Europa (Francoforte)	UE (Francoforte), UE (Irlanda)
Europa (Londra)	Europa (Londra), Europa (Irlanda)
Europa (Irlanda)	Europa (Irlanda)

Regione di Amazon Cognito	Regione dei messaggi SMS
Europa (Parigi)	Europa (Parigi)
Europa (Stoccolma)	Europa (Stoccolma)
Europa (Milano)	Europa (Milano)
Europa (Spagna)	Europa (Spagna)
Europa (Zurigo)	Europa (Zurigo)
Asia Pacifico (Malesia)	Asia Pacifico (Singapore)
Asia Pacifico (Thailandia)	Asia Pacifico (Mumbai)
Asia Pacifico (Mumbai)	Asia Pacifico (Mumbai), Asia Pacifico (Singapore)
Asia Pacifico (Hyderabad)	Asia Pacifico (Hyderabad)
Asia Pacifico (Hong Kong)	Asia Pacifico (Singapore)
Asia Pacifico (Seul)	Asia Pacifico (Tokyo)
Asia Pacifico (Singapore)	Asia Pacifico (Singapore)
Asia Pacifico (Sydney)	Asia Pacifico (Sydney)
Asia Pacifico (Tokyo)	Asia Pacifico (Tokyo)
Asia Pacifico (Giacarta)	Asia Pacifico (Giacarta)
Asia Pacifico (Osaka)	Asia Pacifico (Osaka-Locale)
Asia Pacifico (Melbourne)	Asia Pacifico (Melbourne)
Medio Oriente (Bahrein)	Medio Oriente (Bahrein)
Medio Oriente (Emirati Arabi Uniti)	Medio Oriente (Emirati Arabi Uniti)
Sud America (San Paolo)	Sud America (San Paolo)

Regione di Amazon Cognito	Regione dei messaggi SMS
Israele (Tel Aviv)	Israele (Tel Aviv)
Africa (Città del Capo)	Africa (Città del Capo)

AWS End User Messaging SMS percorso diretto: configurazione della regione

AWS End User Messaging SMS è disponibile in un proprio set di Regioni AWS, che può differire dalle regioni in cui sono disponibili i pool di utenti Amazon SNS o Amazon Cognito. Per l'elenco aggiornato, consulta Regioni e paesi supportati nella Guida per l'AWS End User Messaging SMS utente.

Il `Region` campo dell'`EumsSms`oggetto in SmsConfigurationTypeimposta la Regione AWS destinazione delle chiamate di Amazon Cognito. AWS End User Messaging SMS AWS End User Messaging SMS deve essere disponibile nella regione del tuo pool di utenti. Ometti `Region` e AWS End User Messaging SMS utilizza Amazon Cognito nella regione del tuo pool di utenti o `Region` imposta esplicitamente la regione del tuo pool di utenti. Amazon Cognito rifiuta qualsiasi altro valore.

> **Note**
>
> Se `Region` imposti un valore non supportato,, e le `CreateUserPool` `SetUserPoolMfaConfig` richieste hanno esito negativo e restituiscono un. `UpdateUserPool` `InvalidParameterException`

> **Note**
>
> Il percorso AWS End User Messaging SMS diretto non è disponibile in Asia Pacifico (Tailandia), Asia Pacifico (Hong Kong) e Asia Pacifico (Regioni AWS Malesia). Per inviare messaggi SMS da un pool di utenti in una di queste regioni, utilizza invece il percorso Amazon SNS. Per ulteriori informazioni, consulta Configurazione della regione del percorso Amazon SNS.

Ottenere un'identità di origine per l'invio di messaggi SMS a numeri di telefono USA

Se intendi inviare messaggi SMS a numeri di telefono negli Stati Uniti, devi ottenere un'identità di origine, indipendentemente dalla creazione di un ambiente di test sandbox SMS o di produzione.

I corrieri statunitensi richiedono un'identità di origine per inviare messaggi ai numeri di telefono statunitensi. Se non disponi ancora di un'identità di origine, devi richiederne una. Per informazioni su come ottenere un'identità di origine, consulta Richiedere un numero di telefono nella Guida per l'AWS End User Messaging SMS utente.

Quando hai più di un'identità di origine nella stessa identità Regione AWS, AWS End User Messaging SMS scegli un tipo di identità di origine nel seguente ordine di priorità: codice breve, 10DLC, numero verde. Non puoi modificare questa priorità. Per ulteriori informazioni, consulta FAQ AWS End User Messaging SMS.

Verifica di trovarti nella sandbox SMS

Utilizza la procedura seguente per confermare se ti trovi nella sandbox SMS. Ripeti l'operazione per ogni Regione AWS area in cui hai pool di utenti Amazon Cognito di produzione.

Verifica dello stato della sandbox SMS nella console Amazon Cognito

Per confermare di trovarti nella sandbox SMS

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS .
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli il menu Metodi di autenticazione.
5. Nella sezione Configurazione SMS, espandi Move to Amazon SNS production environment (passare all'ambiente di produzione Amazon SNS). Se l'account si trova nella sandbox SMS, in Amazon Cognito verrà visualizzato il messaggio seguente:

   Configura Servizio AWS le dipendenze per completare la configurazione dei messaggi SMS

   Se non vedi questo messaggio, significa che qualcuno ha già configurato i messaggi SMS nel tuo account. Passa a Completamento della configurazione del pool di utenti in Amazon Cognito.
6. Scegli il link Amazon SNS in Passa all'ambiente di produzione Amazon SNS. La console SNS viene aperta in una nuova scheda.
7. Verifica di trovarti nell'ambiente sandbox. Il messaggio della console indica lo stato della sandbox e Regione AWS, come segue:

   ```
   This account is in the SMS sandbox in US East (N. Virginia).
   ```

Sposta il tuo account fuori dalla sandbox

Per utilizzare la tua app in produzione, sposta il tuo account dall'ambiente della sandbox SMS a quello della produzione. Dopo aver configurato un'identità di origine Regione AWS che contiene le AWS End User Messaging SMS risorse che desideri vengano utilizzate da Amazon Cognito, puoi verificare i numeri di telefono degli Stati Uniti mentre i Account AWS tuoi rimangono nella sandbox SMS. Quando l'ambiente è in produzione, non è necessario verificare i numeri di telefono degli utenti prima di inviare loro messaggi SMS.

Puoi creare una richiesta per uscire dalla sandbox dalla AWS End User Messaging SMS console o dalla console Amazon SNS. Per istruzioni dettagliate, consulta Passare dalla sandbox SMS nella Guida per l'AWS End User Messaging SMS utente.

Usa i numeri del simulatore o i numeri di telefono verificati con AWS End User Messaging SMS

Se hai spostato il tuo account al di fuori dell'ambiente di sperimentazione (sandbox) SMS, ignora questo passaggio.

Se ti trovi nella sandbox ma hai impostato un numero di origine, puoi inviare messaggi a numeri di destinazione verificati. Per configurare destinazioni verificate, consulta Aggiungere un numero di telefono di destinazione verificato nella Guida per l'AWS End User Messaging SMS utente.

Puoi anche inviare messaggi con mittenti e destinazioni simulati. I messaggi del simulatore producono registri ma non vengono inviati tramite la rete dell'operatore. Dal menu Scelte rapide, seleziona Verifica l'invio di SMS con il simulatore SMS. Per ulteriori informazioni, consulta i numeri di telefono del simulatore nella Guida per l'AWS End User Messaging SMS utente.

Completamento della configurazione del pool di utenti in Amazon Cognito

Torna alla scheda del browser in cui stavi creando o modificando il pool di utenti. Completare la procedura. Dopo aver aggiunto con successo la configurazione SMS al tuo pool di utenti, Amazon Cognito invia un messaggio di prova a un numero di telefono interno per verificare che la configurazione funzioni. AWS End User Messaging SMS costi per ogni messaggio SMS di prova.

Utilizzo di Amazon Pinpoint per l'analisi dei pool di utenti

> **Note**
>
> Avviso di fine del supporto: il 30 ottobre 2026, AWS terminerà il supporto per Amazon Pinpoint. Dopo il 30 ottobre 2026, non potrai più accedere alla console Amazon Pinpoint o

alle risorse Amazon Pinpoint (endpoint, segmenti, campagne, percorsi e analisi). Per ulteriori informazioni, consulta Fine del supporto di Amazon Pinpoint. Nota: per quanto APIs riguarda gli SMS, i comandi vocali, i messaggi push su dispositivi mobili, l'OTP e la convalida del numero di telefono non sono interessati da questa modifica e sono supportati da End User Messaging. AWS

I pool di utenti di Amazon Cognito sono integrati con Amazon Pinpoint per fornire analisi per i pool di utenti di Amazon Cognito e per migliorare i dati utente per le campagne di Amazon Pinpoint. Amazon Pinpoint fornisce analisi dei dati e campagne mirate per guidare il coinvolgimento degli utenti nelle app per dispositivi mobili tramite notifiche push. Con il supporto per l'analisi di Amazon Pinpoint nei pool di utenti di Amazon Cognito, puoi tenere traccia delle iscrizioni al pool di utenti, degli accessi, delle autenticazioni non riuscite, degli utenti attivi ogni giorno () e degli utenti attivi mensilmente DAUs () nella console Amazon Pinpoint. MAUs Puoi esplorare i dati in base a intervalli di date o attributi differenti, come piattaforma di dispositivo, dispositivo locale, e versione dell'app.

Puoi inoltre impostare attributi personalizzati per la tua app. Questi possono essere utilizzati per segmentare gli utenti su Amazon Pinpoint e inviare loro notifiche push mirate. Se scegli Condividi i dati degli attributi utente con Amazon Pinpoint nella configurazione Analytics per il tuo client di app nel menu App client nella console Amazon Cognito, Amazon Pinpoint crea endpoint aggiuntivi per gli indirizzi e-mail e i numeri di telefono degli utenti.

Quando attivi le analisi di Amazon Pinpoint nel pool di utenti con la console di Amazon Cognito, crei anche un ruolo collegato ai servizi che Amazon Cognito assume quando effettua una richiesta API ad Amazon Pinpoint per il pool di utenti. Il principale IAM che aggiunge la tua configurazione di analisi deve disporre delle autorizzazioni. CreateServiceLinkedRole Il ruolo collegato al servizio è. AWSServiceRoleForAmazonCognitoIdp Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per Amazon Cognito.

Quando applichi un `AnalyticsConfiguration` al client dell'app nell'API di Amazon Cognito, puoi assegnare un ruolo IAM personalizzato per Amazon Pinpoint e un ID esterno per assumere il ruolo. Il ruolo deve convalidare il principale del servizio `cognito-idp` e, se la policy di attendibilità del ruolo richiede un ID esterno, deve corrispondere a `AnalyticsConfiguration`. Devi concedere le autorizzazioni `cognito-idp:Describe*` del ruolo e le autorizzazioni seguenti per il progetto Amazon Pinpoint.

- `mobiletargeting:UpdateEndpoint`
- `mobiletargeting:PutEvents`

Disponibilità nelle regioni Amazon Cognito e Amazon Pinpoint

La tabella seguente mostra le Regione AWS mappature tra Amazon Cognito e Amazon Pinpoint che soddisfano una delle seguenti condizioni.

- È possibile utilizzare un progetto Amazon Pinpoint solo nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1).
- È possibile utilizzare un progetto Amazon Pinpoint nella stessa regione o nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1)

Per impostazione predefinita, Amazon Cognito può inviare analisi solo a un progetto Amazon Pinpoint all'interno della stessa Regione AWS. Le eccezioni a questa regola sono le regioni riportate nella tabella seguente e le regioni in cui Amazon Pinpoint non è disponibile.

Amazon Pinpoint è disponibile nelle seguenti regioni. I pool di utenti di Amazon Cognito in queste regioni non supportano l'analisi.

- Europa (Milano)
- Medio Oriente (Bahrein)
- Asia Pacifico (Osaka)
- Israele (Tel Aviv)
- Africa (Città del Capo)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Malesia)

La tabella mostra la relazione tra la regione in cui è stato creato il pool di utenti di Amazon Cognito e la regione corrispondente in Amazon Pinpoint. Il progetto Amazon Pinpoint deve essere configurato in una regione disponibile per integrarlo con Amazon Cognito.

Regione del pool di utenti di Amazon Cognito	Regioni per progetto Amazon Pinpoint
ap-northeast-1	us-east-1
ap-northeast-2	us-east-1
ap-south-1	us-east-1, ap-south-1

Regione del pool di utenti di Amazon Cognito	Regioni per progetto Amazon Pinpoint
ap-southeast-1	us-east-1
ap-southeast-2	us-east-1, ap-southeast-2
ca-central-1	us-east-1
eu-central-1	us-east-1, eu-central-1
eu-west-1	us-east-1, eu-west-1
eu-west-2	us-east-1
us-east-1	us-east-1
us-east-2	us-east-1
us-west-2	us-east-1, us-west-2

Esempi di mappatura di regioni

- Se crei un pool di utenti in ap-northest-1, puoi creare il progetto Amazon Pinpoint in us-east-1.
- Se crei un pool di utenti in ap-south-1, puoi creare il progetto Amazon Pinpoint in us-east-1 o ap-south-1.

> **Note**
>
> Per tutti Regioni AWS tranne quelli della tabella precedente, Amazon Cognito può utilizzare solo un progetto Amazon Pinpoint nella stessa regione del pool di utenti. Se Amazon Pinpoint non è disponibile nella regione in cui è stato creato il pool di utenti e non è elencato nella tabella, allora Amazon Cognito non supporta l'analisi Amazon Pinpoint in tale regione. Per informazioni sulla Regione AWS dettagliate, consulta Endpoint e quote di Amazon Pinpoint.

Specifica delle impostazioni di analisi dei dati di Amazon Pinpoint (Console di gestione AWS)

È possibile configurare il pool di utenti di Amazon Cognito per inviare dati di analisi ad Amazon Pinpoint. Amazon Cognito invia dati di analisi solo ad Amazon Pinpoint per utenti locali. Dopo che

hai configurato il pool di utenti per l'associazione a un progetto Amazon Pinpoint, devi includere `AnalyticsMetadata` nelle richieste API. Per ulteriori informazioni, consulta Integrazione di app con Amazon Pinpoint.

Specificazione delle impostazioni di analisi

1. Passa alla console Amazon Cognito. Potrebbe comparire una richiesta di inserimento delle credenziali AWS .
2. Seleziona User Pools (Pool di utenti) e scegli un pool di utenti esistente dall'elenco.
3. Scegli il menu App clients e seleziona il client dell'app che desideri aggiornare.
4. Nella scheda Analisi sotto Pinpoint analytics, scegli Abilita.
5. Scegli una Pinpoint Region (Regione Pinpoint).
6. Scegli un Amazon Pinpoint project (Progetto Amazon Pinpoint) o seleziona Create Amazon Pinpoint project (Crea progetto Amazon Pinpoint).

   > **Note**
   >
   > L'ID progetto di Amazon Pinpoint è una stringa di 32 caratteri univoca per il progetto. È elencato nella console Amazon Pinpoint.
   > Puoi mappare più app di Amazon Cognito a un singolo progetto di Amazon Pinpoint. Tuttavia, ogni app di Amazon Cognito può essere mappata solo a un progetto di Amazon Pinpoint.
   > In Amazon Pinpoint, ogni progetto deve essere una singola app. Ad esempio, se uno sviluppatore di videogiochi ha due videogiochi, ognuno di questi dovrebbe costituire un progetto di Amazon Pinpoint separato, nonostante entrambi usino lo stesso bacino d'utenza di Amazon Cognito. Per ulteriori informazioni sui progetti Amazon Pinpoint, consulta Creazione di un progetto in Amazon Pinpoint.

7. In User data sharing (Condivisione dei dati utente), scegli Share user data with Amazon Pinpoint (Condividi i dati utente con Amazon Pinpoint) se Amazon Cognito deve inviare indirizzi e-mail e numeri di telefono ad Amazon Pinpoint e creare endpoint aggiuntivi per gli utenti. Dopo che gli utenti verificano i rispetti indirizzi e-mail e numeri di telefono, Amazon Cognito li condividi solo con Amazon Pinpoint se sono disponibili per l'account utente.

> **Note**
>
> Un endpoint identifica in modo univoco un dispositivo dell'utente al quale puoi inviare notifiche push con Amazon Pinpoint. Per ulteriori informazioni sugli endpoint, consulta la sezione Aggiunta di endpoint nella Guida per gli sviluppatori di Amazon Pinpoint.

8. Scegli Save changes (Salva modifiche).

Specificazione delle impostazioni di analisi AWS CLI (AWS e dell'API) di Amazon Pinpoint

Utilizza i comandi seguenti per specificare le impostazioni di analisi dei dati di Amazon Pinpoint per il tuo bacino d'utenza.

Specificazione di impostazioni di analisi per l'app client esistente del tuo bacino d'utenza al momento della creazione dell'app.

- AWS CLI: `aws cognito-idp create-user-pool-client`
- AWS API: CreateUserPoolClient

Aggiornamento delle impostazioni di analisi per l'app client esistente del tuo bacino d'utenza

- AWS CLI: `aws cognito-idp update-user-pool-client`
- AWS API: UpdateUserPoolClient

> **Note**
>
> Se utilizzi `ApplicationArn`, Amazon Cognito supporta le integrazioni all'interno della regione.

Integrazione di app con Amazon Pinpoint

Puoi pubblicare metadati di analisi su Amazon Pinpoint per gli utenti locali di Amazon Cognito nell'API dei pool di utenti.

Utenti locali

Sono gli utenti che hanno creato un account mediante la registrazione o sono stati creati nel pool di utenti anziché eseguire l'accesso tramite un gestore dell'identità digitale (IdP) di terze parti.

API dei pool di utenti

Le operazioni che puoi integrare con un AWS SDK, utilizzando un'app con un'interfaccia utente (UI) personalizzata. Non puoi trasmettere metadati di analisi a utenti federati o locali che accedono tramite accesso gestito. Per un elenco di operazioni dei pool di utenti, consultare la documentazione di riferimento delle API di Amazon Cognito.

Dopo aver configurato il pool di utenti per la pubblicazione in una campagna, Amazon Cognito passa i metadati ad Amazon Pinpoint per le seguenti operazioni API.

- `AdminInitiateAuth`
- `AdminRespondToAuthChallenge`
- `ConfirmForgotPassword`
- `ConfirmSignUp`
- `ForgotPassword`
- `InitiateAuth`
- `ResendConfirmationCode`
- `RespondToAuthChallenge`
- `SignUp`

Per passare i metadati relativi alla sessione dell'utente alla campagna Amazon Pinpoint, includi un valore `AnalyticsEndpointId` nel parametro `AnalyticsMetadata` della richiesta API. JavaScript Ad esempio, vedi Perché le analisi del mio pool di utenti di Amazon Cognito non vengono visualizzate nella mia dashboard di Amazon Pinpoint? nel Knowledge Center.AWS

Protezione dei pool di utenti con protezione dalle minacce

Potresti voler proteggere la tua applicazione dalle intrusioni di rete, dall'indovinazione delle password, dall'impersonificazione degli utenti e da iscrizioni e accessi malintenzionati. La configurazione delle funzionalità di sicurezza dei pool di utenti di Amazon Cognito può essere un componente chiave della tua architettura di sicurezza. La sicurezza della tua applicazione è responsabilità del cliente («Sicurezza nel cloud»), come descritto nel Modello di responsabilità AWS condivisa. Gli strumenti di questo capitolo contribuiscono a far sì che la progettazione della sicurezza delle applicazioni sia in linea con questi obiettivi.

Per le linee guida sulla sicurezza a livello di account e a livello di servizio, vale a dire il modello di responsabilità condivisa, la protezione dei dati, la conformità e la sicurezza dell'infrastruttura, vedi. Sicurezza in Amazon Cognito

Una decisione importante da prendere quando si configura il pool di utenti è se consentire la registrazione e l'accesso pubblici. Alcune opzioni relative ai pool di utenti, come i client riservati, la creazione e la conferma amministrativa degli utenti e i pool di utenti senza dominio, sono meno soggette agli attacchi su Internet. Tuttavia, un caso d'uso comune sono i client pubblici che accettano l'iscrizione di chiunque su Internet e inviano tutte le operazioni direttamente al pool di utenti. In qualsiasi configurazione, ma soprattutto nel caso di queste configurazioni pubbliche, consigliamo di pianificare e implementare il pool di utenti tenendo conto delle funzionalità di sicurezza. Una sicurezza insufficiente può influire sulla AWS bolletta anche quando fonti indesiderate creano nuovi utenti attivi o tentano di sfruttare gli utenti esistenti.

La MFA e la protezione dalle minacce si applicano agli utenti locali. Third-party IdPs sono responsabili del livello di sicurezza degli utenti federati.

Funzionalità di sicurezza dei pool di utenti

Multi-factor autenticazione (MFA)

: Richiedi un codice che il tuo pool di utenti invii tramite e-mail (con il piano di funzionalità Essentials o Plus) o SMS o da un'app di autenticazione per confermare l'accesso al pool di utenti.

Protezione dalle minacce

: Monitora l'accesso per individuare gli indicatori di rischio e applica la MFA o blocca l'accesso. Aggiungi attestazioni e ambiti personalizzati per accedere ai token. Invia codici MFA tramite e-mail.

AWS WAF ACL web

Ispeziona il traffico in entrata verso gli endpoint del tuo pool di utenti e l'API di autenticazione per individuare attività indesiderate a livello di rete e di applicazione.

Distinzione tra lettere maiuscole e minuscole

Impedisci la creazione di utenti il cui indirizzo e-mail o nome utente preferito sia identico a quello di un altro utente, ad eccezione delle maiuscole e minuscole.

Protezione da eliminazione

Impedite ai sistemi automatici di eliminare accidentalmente i vostri pool di utenti. Richiedi un'ulteriore conferma dell'eliminazione del pool di utenti in. Console di gestione AWS

Errori di esistenza degli utenti

Evita la divulgazione di nomi utente e alias esistenti nel tuo pool di utenti. Restituisce un errore generico in risposta all'autenticazione non riuscita, indipendentemente dal fatto che il nome utente sia valido o meno.

Argomenti

- Best practice di sicurezza per i bacini d'utenza di Amazon Cognito
- Aggiunta dell'autenticazione MFA a un bacino d'utenza
- Livelli di autenticazione con attestazioni ACR e AMR
- Utilizzo dei dispositivi utente nel pool di utenti
- Sicurezza avanzata con protezione dalle minacce
- Associa un AWS WAF ACL web con un pool di utenti
- Distinzione tra maiuscole e minuscole del bacino d'utenza
- Protezione da eliminazione del bacino d'utenza
- Gestione delle risposte agli errori relativi all'esistenza degli utenti

Best practice di sicurezza per i bacini d'utenza di Amazon Cognito

Questa pagina descrive le migliori pratiche di sicurezza che è possibile implementare quando si desidera proteggersi dalle minacce comuni. La configurazione scelta dipende dal caso d'uso di ciascuna applicazione. Si consiglia di applicare almeno il minimo privilegio alle operazioni

amministrative e di adottare misure per proteggere i segreti delle applicazioni e degli utenti. Un altro passaggio avanzato ma efficace che puoi eseguire è configurare e applicare gli ACL AWS WAF Web ai tuoi pool di utenti.

Proteggi il tuo pool di utenti a livello di rete

AWS WAF gli ACL web possono proteggere le prestazioni e i costi dei meccanismi di autenticazione creati con Amazon Cognito. Con gli ACL web, puoi implementare guardrail davanti alle API e alle richieste di accesso gestite. Gli ACL Web creano filtri a livello di rete e di applicazione che possono ridurre il traffico o richiedere un CAPTCHA in base a regole definite dall'utente. Le richieste non vengono trasferite alle tue risorse Amazon Cognito finché non soddisfano i requisiti delle tue regole ACL web. Per ulteriori informazioni, consulta AWS WAF gli ACL web.

Proteggiti dall'abuso dei messaggi SMS

Quando consenti l'iscrizione pubblica nel tuo pool di utenti, puoi configurare la verifica dell'account con i codici che Amazon Cognito invia nei messaggi di testo SMS. I messaggi SMS possono essere associati ad attività indesiderate e aumentare la AWS bolletta. Configura la tua infrastruttura in modo che sia resistente all'invio di messaggi SMS in circostanze di frode. Per ulteriori informazioni, consulta i seguenti post tratti dai AWS blog.

- Riduci i rischi di frode nelle iscrizioni degli utenti e di invio di SMS con i pool di utenti di Amazon Cognito
- Difesa dal pompaggio di SMS: nuove AWS funzionalità per aiutare a combattere il traffico gonfiato artificialmente

Comprendi l'autenticazione pubblica

I pool di utenti di Amazon Cognito dispongono di funzionalità CIAM (Customer Identity and Access Management) che supportano casi d'uso in cui membri del pubblico generico possono registrare un account utente e accedere alle tue applicazioni. Quando un pool di utenti consente l'iscrizione in modalità self-service, è aperto alle richieste di account utente provenienti dalla rete Internet pubblica. Self-service le richieste provengono da operazioni API come SignUpe InitiateAuthdall'interazione dell'utente con l'accesso gestito. Puoi configurare i pool di utenti per mitigare gli abusi che potrebbero derivare da richieste pubbliche o disabilitare completamente le operazioni di autenticazione pubblica.

Le impostazioni seguenti illustrano alcuni dei modi in cui è possibile gestire le richieste di autenticazione pubbliche e interne nei pool di utenti e nei client delle app.

Esempi di impostazioni del pool di utenti che influiscono sull'accesso al pool di utenti pubblici

Impostazione	Opzioni disponibili	Configurato su	Effetto sull'autenticazione pubblica	Impostazione della console	Funzionamento e parametro dell'API
Self-serviceiscrizione	Consenti agli utenti di registrare un account o creare account utente come amministratore.	Bacino d'utenza	Impedisci l'iscrizione pubblica	Sign-up—Self-service iscriviti	CreateUserPool, UpdateUserPool `AdminCreateUserConfig – AllowAdminCreateUserOnly`
Conferma dell'amministratore	Invia codici di conferma a nuovi utenti o richiedi agli amministratori di confermarli.	Bacino d'utenza	Impedisci la conferma dell'iscrizione senza l'intervento dell'amministratore	Sign-up—Cognito-assisted verifica e conferma	CreateUserPool, UpdateUserPool `AccountRecoverySettings – admin_only`
Divulgazione all'utente	Invia messaggi «utente non trovato» all'accesso e alla reimpostazione della password o impedisci la divulgazione.	Client dell'app	Evita di indovinare il nome, l'indirizzo email o i numeri di telefono di accesso	Client per app: evita gli errori relativi all'esistenza	CreateUserPoolClient, UpdateUserPoolClient

Impostazione	Opzioni disponibili	Configurato su	Effetto sull'autenticazione pubblica	Impostazione della console	Funzionamento e parametro dell'API
				degli utenti	`PreventUserExistenceErrors`
Client secret	Richiedi o non richiedi un hash segreto al momento della registrazione, dell'accesso e della reimpostazione della password	Client dell'app	Proteggiti dalle richieste di autenticazione provenienti da fonti non autorizzate	Client per app: segreto del cliente	CreateUserPoolClient `GenerateSecret`
ACL Web	Abilita o non abilita un firewall di rete per le richieste di autenticazione	Bacino d'utenza	Limita o impedisci l'accesso in base alle caratteristiche delle richieste e alle regole degli indirizzi IP definite dall'amministratore	AWS WAF—Impostazioni WAF	AssociateWebACL `ResourceArn`
IdP esterno	Consenti l'accesso agli utenti di terze parti IdPs, alla directory del pool di utenti o a entrambe	Client dell'app	Escludi gli utenti locali o gli utenti federati dalla registrazione e dall'accesso.	Client di app: provider di identità	CreateUserPoolClient, UpdateUserPoolClient `SupportedIdentityProviders`

Impostazione	Opzioni disponibili	Configurato su	Effetto sull'autenticazione pubblica	Impostazione della console	Funzionamento e parametro dell'API
Server di autorizzazione	Ospita o non ospita pagine Web pubbliche per l'autenticazione	Bacino d'utenza	Disattiva le pagine Web pubbliche e consenti solo l'autenticazione SDK-based	Dominio	CreateUserPoolDomain La creazione di un dominio con pool di utenti rende disponibili le pagine Web pubbliche.

Impostazione	Opzioni disponibili	Configurato su	Effetto sull'autenticazione pubblica	Impostazione della console	Funzionamento e parametro dell'API
Protezione dalle minacce	Abilita o disabilita il monitoraggio di segnali di attività dannose o password non sicure	Pool di utenti o client di app	Può bloccare automaticamente l'accesso o richiedere l'autenticazione a più fattori quando gli utenti mostrano indicatori di compromissione	Protezione dalle minacce: impostazioni di protezione	SetRiskConfiguration I parametri di `SetRiskConfiguration` definizione delle impostazioni di protezione dalle minacce.

Proteggi i clienti riservati con i segreti dei clienti

Il client secret è una stringa opzionale associata a un client di app. Tutte le richieste di autenticazione ai client di app con client secret devono includere un hash segreto generato dal nome utente, dall'ID client e dal segreto del client. Chi non conosce il segreto del client viene escluso dall'applicazione sin dall'inizio.

Tuttavia, i segreti dei clienti hanno dei limiti. Se si incorpora un segreto client in un software client pubblico, il segreto del client può essere esaminato. Questo apre la possibilità di creare utenti, inviare richieste di reimpostazione della password ed eseguire altre operazioni nel client dell'app. I segreti del client devono essere implementati solo quando un'applicazione è l'unica entità che ha accesso al segreto. In genere, ciò è possibile nelle applicazioni client riservate lato server. Questo vale anche per le applicazioni M2M, in cui è richiesto un segreto del client. Archivia il segreto del client in una

memoria locale crittografata oppure AWS Secrets Manager. Non lasciare mai che il segreto del tuo cliente sia visibile sulla rete Internet pubblica.

Proteggi altri segreti

Il tuo sistema di autenticazione con pool di utenti di Amazon Cognito potrebbe gestire dati privati, password e credenziali. AWS Di seguito sono riportate alcune best practice per gestire i segreti a cui l'applicazione potrebbe accedere.

Password

Gli utenti possono inserire le password quando accedono all'applicazione. Amazon Cognito dispone di token di aggiornamento che l'applicazione può utilizzare per continuare le sessioni utente scadute senza richiedere una nuova password. Non inserire password o hash di password nell'archiviazione locale. Progetta la tua applicazione in modo da trattare le password come opache e passarle solo al tuo pool di utenti.

Come best practice, implementate l'autenticazione senza password con passkey. WebAuthn Se è necessario implementare le password, utilizzare il flusso di autenticazione Secure Remote Password (SRP) e l'autenticazione a più fattori (MFA).

AWS credenziali

L'autenticazione amministrativa e le operazioni amministrative del pool di utenti richiedono l'autenticazione con AWS credenziali. Per implementare queste operazioni in un'applicazione, concedi l'accesso sicuro alle AWS credenziali temporanee. Concedi l'accesso alle credenziali solo alle applicazioni eseguite su un componente server che controlli. Non inserire applicazioni che contengono AWS credenziali su sistemi pubblici di controllo delle versioni come. GitHub Non codificate le AWS credenziali in applicazioni pubbliche lato client.

Verificatore di codice PKCE

Proof Key for Code Exchange, o PKCE, serve per la concessione del codice di autorizzazione OpenID Connect (OIDC) con il server di autorizzazione del pool di utenti. Le applicazioni condividono i segreti del codice di verifica del codice con il pool di utenti quando richiedono codici di autorizzazione. Per scambiare codici di autorizzazione in token, i clienti devono riaffermare di conoscere il verificatore di codice. Questa pratica evita l'emissione di token con codici di autorizzazione intercettati.

I client devono generare un nuovo verificatore di codice casuale con ogni richiesta di autorizzazione. L'uso di un verificatore di codice statico o prevedibile significa che solo allora un

utente malintenzionato deve intercettare il verificatore codificato e il codice di autorizzazione. Progetta la tua applicazione in modo che non esponga i valori del verificatore del codice agli utenti.

Privilegio minimo di amministrazione del pool di utenti

Le policy IAM possono definire il livello di accesso dei responsabili alle operazioni di amministrazione del pool di utenti di Amazon Cognito e alle operazioni di autenticazione amministrativa. Ad esempio:

- A un server Web, concedi le autorizzazioni per l'autenticazione con operazioni amministrative dell'API.
- A un Centro identità AWS IAM utente che gestisce un pool di utenti nel tuo Account AWS, concedi le autorizzazioni per la manutenzione e la segnalazione del pool di utenti.

Il livello di granularità delle risorse in Amazon Cognito è limitato a due tipi di risorse ai fini delle policy IAM: pool di utenti e pool di identità. Tieni presente che non puoi applicare autorizzazioni per gestire singoli client di app. Configura i pool di utenti sapendo che le autorizzazioni concesse sono valide per tutti i client dell'app. Quando la tua organizzazione ha più tenant di applicazioni e il tuo modello di sicurezza richiede la separazione delle responsabilità amministrative tra i tenant, implementa la multi-tenancy con un tenant per pool di utenti.

Sebbene sia possibile creare policy IAM con autorizzazioni per operazioni di autenticazione degli utenti, ad esempio`InitiateAuth`, tali autorizzazioni non hanno alcun effetto. Le operazioni API pubbliche e autorizzate tramite token non sono soggette alle autorizzazioni IAM. Tra le operazioni di autenticazione del pool di utenti disponibili, puoi concedere le autorizzazioni solo a operazioni amministrative lato server come. `AdminInitiateAuth`

È possibile limitare i livelli di amministrazione del pool di utenti con elenchi di privilegi minimi. `Action` La seguente politica di esempio è destinata a un amministratore che può gestire i server di risorse IdPs, i client delle app e il dominio del pool di utenti, ma non gli utenti o il pool di utenti.

JSON

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

```
      "Action": [
        "cognito-idp:CreateIdentityProvider",
        "cognito-idp:CreateManagedLoginBranding",
        "cognito-idp:CreateResourceServer",
        "cognito-idp:CreateUserPoolDomain",
        "cognito-idp:DeleteIdentityProvider",
        "cognito-idp:DeleteResourceServer",
        "cognito-idp:DeleteUserPoolDomain",
        "cognito-idp:DescribeIdentityProvider",
        "cognito-idp:DescribeManagedLoginBranding",
        "cognito-idp:DescribeManagedLoginBrandingByClient",
        "cognito-idp:DescribeResourceServer",
        "cognito-idp:DescribeUserPool",
        "cognito-idp:DescribeUserPoolClient",
        "cognito-idp:DescribeUserPoolDomain",
        "cognito-idp:GetIdentityProviderByIdentifier",
        "cognito-idp:GetUICustomization",
        "cognito-idp:ListIdentityProviders",
        "cognito-idp:ListResourceServers",
        "cognito-idp:ListUserPoolClients",
        "cognito-idp:ListUserPools",
        "cognito-idp:SetUICustomization",
        "cognito-idp:UpdateIdentityProvider",
        "cognito-idp:UpdateManagedLoginBranding",
        "cognito-idp:UpdateResourceServer",
        "cognito-idp:UpdateUserPoolClient",
        "cognito-idp:UpdateUserPoolDomain"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_EXAMPLE"
    }
  ]
}
```

La seguente politica di esempio concede la gestione e l'autenticazione di utenti e gruppi a un'applicazione lato server.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "UserAdminAuthN",
      "Action": [
        "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:AdminInitiateAuth",
        "cognito-idp:AdminLinkProviderForUser",
        "cognito-idp:AdminListDevices",
        "cognito-idp:AdminListGroupsForUser",
        "cognito-idp:AdminListUserAuthEvents",
        "cognito-idp:AdminRemoveUserFromGroup",
        "cognito-idp:AdminResetUserPassword",
        "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:ListGroups",
        "cognito-idp:ListUsers",
        "cognito-idp:ListUsersInGroup",
        "cognito-idp:RevokeToken",
        "cognito-idp:UpdateGroup",
        "cognito-idp:VerifySoftwareToken"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_EXAMPLE"
    }
  ]
```

```
}
```

Proteggi e verifica i token

I token possono contenere riferimenti interni all'appartenenza al gruppo e agli attributi utente che potresti non voler divulgare all'utente finale. Non archiviate ID e token di accesso nella memoria locale. I token di aggiornamento sono crittografati con una chiave accessibile solo al pool di utenti e sono opachi per utenti e applicazioni. Revoca i token di aggiornamento quando gli utenti si disconnettono o quando stabilisci che la persistenza della sessione di un utente non è desiderata per motivi di sicurezza.

Utilizza i token di accesso per autorizzare l'accesso solo ai sistemi che verificano in modo indipendente che il token sia valido e non scaduto. Per le risorse di verifica, consulta Verifica di un token web JSON.

Determina i provider di identità di cui ti vuoi fidare

Quando configuri il tuo pool di utenti con provider di identità SAML o OIDC (IdPs), IdPs puoi creare nuovi utenti, impostare gli attributi utente e accedere alle risorse dell'applicazione. I provider SAML e OIDC vengono generalmente utilizzati in scenari business-to-business (B2B) o aziendali in cui tu o il tuo cliente diretto controllate l'appartenenza e la configurazione del provider.

I provider di servizi sociali offrono account utente a chiunque sia su Internet e sono meno soggetti al controllo dell'utente rispetto ai provider aziendali. Attiva i social IdPs nel client dell'app solo quando sei pronto a consentire ai clienti pubblici di accedere e accedere alle risorse della tua applicazione.

Comprendi l'effetto degli ambiti sull'accesso ai profili utente

È possibile richiedere gli ambiti di controllo degli accessi nelle richieste di autenticazione al server di autorizzazione del pool di utenti. Questi ambiti possono concedere agli utenti l'accesso a risorse esterne e possono concedere agli utenti l'accesso per visualizzare e modificare i propri profili utente. Configura i client dell'app per supportare gli ambiti minimi necessari per il funzionamento dell'applicazione.

L'`aws.cognito.signin.user.admin`ambito è presente in tutti i token di accesso emessi dall'autenticazione SDK con operazioni come. InitiateAuth È destinato alle operazioni self-service dei profili utente nell'applicazione. Puoi anche richiedere questo ambito al tuo server di autorizzazione. Questo ambito è necessario per operazioni autorizzate da token come UpdateUserAttributese.

GetUser L'effetto di queste operazioni è limitato dalle autorizzazioni di lettura e scrittura del client dell'app.

Gli phone ambiti `openidprofile`,`email`, e autorizzano le richieste al server di autorizzazione del Endpoint UserInfo pool di utenti. Definiscono gli attributi che l'endpoint può restituire. L'`openid`ambito, se richiesto senza altri ambiti, restituisce tutti gli attributi disponibili, ma quando si richiedono più ambiti nella richiesta, la risposta viene ristretta agli attributi rappresentati dagli ambiti aggiuntivi. L'`openid`ambito indica anche una richiesta di token ID; quando ometti questo ambito dalla richiesta alla tuaEndpoint Authorize, Amazon Cognito emette solo un token di accesso e, se applicabile, un token di aggiornamento. Per ulteriori informazioni, consulta gli ambiti OpenID Connect all'indirizzo. Termini del client dell'app

Disinfetta gli input per gli attributi utente

Gli attributi utente che potrebbero diventare metodi di consegna e nomi utente, ad esempio`email`, hanno restrizioni di formato. Altri attributi possono avere tipi di dati di tipo stringa, booleano o numerico. I valori degli attributi String supportano una varietà di input. Configura la tua applicazione per proteggerti dai tentativi di scrivere dati indesiderati nella tua directory utente o nei messaggi che Amazon Cognito invia agli utenti. Esegui la convalida lato client dei valori degli attributi di stringa inviati dall'utente nella tua applicazione prima di inviarli ad Amazon Cognito.

I pool di utenti mappano gli attributi dal tuo pool di utenti in base IdPs a una mappatura di attributi specificata da te. Mappa solo gli attributi IdP sicuri e prevedibili agli attributi della stringa del pool di utenti.

Aggiunta dell'autenticazione MFA a un bacino d'utenza

La tecnologia MFA aggiunge un fattore di autenticazione «qualcosa che possiedi» al fattore iniziale «qualcosa che conosci», che in genere è un nome utente e una password. Puoi scegliere SMS, messaggi di posta elettronica o password monouso basate sul tempo (TOTP) come fattori aggiuntivi per accedere agli utenti che utilizzano le password come fattore di autenticazione principale.

Multi-factor l'autenticazione (MFA) aumenta la sicurezza per gli utenti locali dell'applicazione. Nel caso di utenti federati, Amazon Cognito delega tutti i processi di autenticazione all'IdP e non offre loro fattori di autenticazione aggiuntivi.

> **Note**
>
> La prima volta che un nuovo utente accede all'app, Amazon Cognito emette i token OAuth 2.0 anche se il bacino d'utenza richiede MFA. Il secondo fattore di autenticazione quando l'utente accede per la prima volta è la conferma del messaggio di verifica inviata da Amazon Cognito. Se il bacino d'utenza richiede MFA, Amazon Cognito richiede all'utente di registrare un ulteriore fattore di accesso da utilizzare per ogni tentativo di accesso dopo il primo.

Con l'autenticazione adattiva, puoi configurare il tuo pool di utenti in modo che richieda un fattore di autenticazione aggiuntivo in risposta a un aumento del livello di rischio. Per aggiungere l'autenticazione adattiva al bacino d'utenza, consulta Sicurezza avanzata con protezione dalle minacce.

Quando imposti l'autenticazione MFA su `required` per un bacino d'utenza, tutti gli utenti devono completare l'MFA per accedere. Per accedere, ogni utente deve impostare almeno un fattore MFA. Quando è richiesta la MFA, è necessario includere la configurazione MFA nell'onboarding degli utenti in modo che il pool di utenti consenta loro di accedere.

L'accesso gestito richiede agli utenti di configurare l'MFA quando si imposta l'MFA come richiesto. Quando imposti l'autenticazione a più fattori come facoltativa nel tuo pool di utenti, l'accesso gestito non richiede alcuna richiesta agli utenti. Per utilizzare MFA opzionale, devi creare un'interfaccia nell'app che richiede agli utenti di scegliere se desiderano configurare MFA, quindi devi guidarli attraverso gli input dell'API per verificare il fattore di accesso aggiuntivo.

Argomenti

- Cose da sapere sull'MFA per pool di utenti
- Preferenze utente MFA
- Dettagli della logica MFA in fase di esecuzione dell'utente
- Configurare un pool di utenti per l'autenticazione a più fattori
- MFA per SMS e messaggi e-mail
- Autenticazione MFA con token di software TOTP

Cose da sapere sull'MFA per pool di utenti

Prima di configurare l'MFA, valuta quanto segue:

- Gli utenti possono disporre di MFA o accedere con fattori senza password, con un'eccezione: le passkey con verifica utente possono soddisfare i requisiti MFA come primo fattore se impostati nel proprio pool di utenti. `FactorConfiguration` `MULTI_FACTOR_WITH_USER_VERIFICATION` `WebAuthnConfiguration` Le passkey non possono essere utilizzate come secondo fattore per l'accesso tramite password.
  - Non è possibile impostare l'autenticazione a più fattori su richiesta nei pool di utenti che supportano password monouso.
  - Non puoi aggiungere `EMAIL_OTP` o `SMS_OTP` aggiungere `AllowedFirstAuthFactors` quando l'MFA è richiesta nel tuo pool di utenti. È possibile aggiungere `WEB_AUTHN` quando `FactorConfiguration` è impostato su. `MULTI_FACTOR_WITH_USER_VERIFICATION`
  - Choice-basedsolo le offerte `PASSWORD` e i `PASSWORD_SRP` fattori di accesso in tutti i client dell'app quando è richiesta l'autenticazione a più fattori nel pool di utenti. Per ulteriori informazioni sui flussi nome utente e password, consulta Sign-in con password persistenti e Sign-in con password persistenti e payload sicuro nel capitolo Autenticazione di questa guida.
  - Nei pool di utenti in cui l'autenticazione MFA è facoltativa, gli utenti che hanno configurato un fattore MFA possono accedere solo con flussi di autenticazione nome utente e password nell'accesso basato sulla scelta. Questi utenti sono idonei a tutti i flussi di accesso basati su client.

La tabella seguente descrive l'effetto delle impostazioni MFA del pool di utenti e della configurazione utente dei fattori MFA sulla capacità degli utenti di accedere con fattori senza password.

Impostazione MFA del pool di utenti	Stato MFA utente	Webauthn/OTP disponibile	Richiesta di autenticazione a più fattori dopo l'accesso con password	Può accedere con WebAuthn/ OTP
Richiesto	Configurato	No	Sì	No
Richiesto	Non configurato	No	No (non riesco ad accedere)	No

Impostazione MFA del pool di utenti	Stato MFA utente	Webauthn/OTP disponibile	Richiesta di autenticazione a più fattori dopo l'accesso con password	Può accedere con WebAuthn/ OTP
Facoltativo	Configurato	Può configura re WebAuthn ma non può accedere con passkey	Sì	No
Facoltativo	Non configurato	Sì	No	Sì
Opzionale (con passkey MFA abilitata)	Passkey MFA configurata	Sì	Sì (la passkey con verifica utente soddisfa l'MFA in modo indipendente)	Sì (la passkey con verifica utente soddisfa l'MFA in modo indipendente)
Obbligatorio (con passkey MFA abilitata)	Passkey MFA configurata	Sì	Sì (la passkey con verifica utente soddisfa l'MFA in modo indipendente)	Sì (la passkey con verifica utente soddisfa l'MFA in modo indipendente)
Disattivata	Qualsiasi	Sì	No	Sì

- Il metodo MFA preferito dall'utente influenza i metodi che può utilizzare per recuperare la password. Gli utenti la cui MFA preferita è tramite messaggio di posta elettronica non possono ricevere un codice di reimpostazione della password tramite e-mail. Gli utenti la cui MFA preferita è tramite SMS non possono ricevere un codice di reimpostazione della password tramite SMS.

  Le impostazioni di recupero della password devono fornire un'opzione alternativa quando gli utenti non sono idonei al metodo preferito di reimpostazione della password. Ad esempio, i meccanismi di ripristino potrebbero avere l'e-mail come priorità assoluta e la MFA e-mail potrebbe essere un'opzione nel tuo pool di utenti. In questo caso, aggiungi il ripristino SMS-message dell'account

come seconda opzione o utilizza le operazioni amministrative dell'API per reimpostare le password di tali utenti.

Amazon Cognito risponde alle richieste di reimpostazione della password di utenti che non dispongono di un metodo di ripristino valido con una risposta di errore. `InvalidParameterException`

Il corpo della richiesta di esempio UpdateUserPoolillustra come gli utenti possono ricorrere al ripristino tramite messaggio SMS quando la reimpostazione della password dei messaggi e-mail non è disponibile. `AccountRecoverySetting`

- Gli utenti non possono ricevere codici MFA e di reimpostazione della password allo stesso indirizzo e-mail o numero di telefono. Se utilizzano password monouso (OTP) dai messaggi e-mail per MFA, devono utilizzare i messaggi SMS per il ripristino dell'account. Se utilizzano le OTP dei messaggi SMS per l'MFA, devono utilizzare i messaggi e-mail per il ripristino dell'account. Nei pool di utenti con MFA, gli utenti potrebbero non essere in grado di completare il ripristino self-service della password se dispongono di attributi per il proprio indirizzo e-mail ma non dispongono del numero di telefono o del numero di telefono ma non dell'indirizzo e-mail.

  Per evitare che gli utenti non possano reimpostare le proprie password nei pool di utenti con questa configurazione, imposta gli `phone_number` attributi `email` e come richiesto. In alternativa, puoi configurare processi che raccolgono e impostano sempre tali attributi quando gli utenti si registrano o quando gli amministratori creano profili utente. Quando gli utenti dispongono di entrambi gli attributi, Amazon Cognito invia automaticamente i codici di reimpostazione della password alla destinazione che non è il fattore MFA dell'utente.
- Quando attivi la MFA nel tuo pool di utenti e scegli il messaggio SMS o il messaggio e-mail come secondo fattore, puoi inviare messaggi a un numero di telefono o a un attributo e-mail che non hai verificato in Amazon Cognito. Dopo che l'utente ha completato la MFA, Amazon Cognito imposta `phone_number_verified` l'attributo or su. `email_verified` `true`
- Dopo cinque tentativi non riusciti di presentazione di un codice MFA, Amazon Cognito avvia il processo di blocco con timeout esponenziale descritto in Comportamento di blocco in caso di tentativi di accesso non riusciti.
- Se il tuo account si trova nella sandbox SMS Regione AWS che contiene le risorse Amazon Simple Notification Service (Amazon SNS) per il tuo pool di utenti, devi verificare i numeri di telefono in Amazon SNS prima di poter inviare un messaggio SMS. Per ulteriori informazioni, consulta Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.

- Per modificare lo stato MFA degli utenti in risposta agli eventi rilevati con protezione dalle minacce, attiva l'MFA e impostala come opzionale nella console del pool di utenti di Amazon Cognito. Per ulteriori informazioni, consulta Sicurezza avanzata con protezione dalle minacce.
- I messaggi e-mail e SMS richiedono che gli utenti abbiano rispettivamente gli attributi dell'indirizzo e-mail e del numero di telefono. È possibile impostare `email` o `phone_number` come attributi obbligatori nel proprio pool di utenti. In questo caso, gli utenti non possono completare la registrazione a meno che non forniscano un numero di telefono. Se non imposti questi attributi come obbligatori ma desideri eseguire la MFA per e-mail o SMS, richiedi agli utenti il loro indirizzo e-mail o numero di telefono al momento della registrazione. Come procedura ottimale, configurate il pool di utenti in modo che inviino automaticamente messaggi agli utenti per verificare questi attributi.

  Amazon Cognito considera verificato un numero di telefono o un indirizzo e-mail se un utente ha ricevuto con successo un codice temporaneo tramite SMS o messaggio e-mail e lo ha restituito in una richiesta VerifyUserAttributeAPI. In alternativa, il tuo team può impostare numeri di telefono e contrassegnarli come verificati con un'applicazione amministrativa che esegue le richieste AdminUpdateUserAttributesAPI.
- Se hai impostato la MFA come obbligatoria e hai attivato più di un fattore di autenticazione, Amazon Cognito richiede ai nuovi utenti di selezionare un fattore di MFA che desiderano utilizzare. Gli utenti devono disporre di un numero di telefono per configurare la MFA dei messaggi SMS e di un indirizzo e-mail per configurare l'autenticazione MFA per i messaggi di posta elettronica. Se un utente non dispone dell'attributo definito per nessuna MFA basata su messaggi disponibile, Amazon Cognito richiede di configurare TOTP MFA. La richiesta di scegliere un fattore MFA `SELECT_MFA_TYPE` () e di impostare un fattore scelto `MFA_SETUP` () si presenta come risposta InitiateAuthalla sfida alle operazioni API AdminInitiateAuth.

Preferenze utente MFA

Gli utenti possono configurare più fattori MFA. Solo uno può essere attivo. Puoi scegliere la preferenza MFA effettiva per i tuoi utenti nelle impostazioni del pool di utenti o nei prompt degli utenti. Un pool di utenti richiede all'utente i codici MFA quando le impostazioni del pool di utenti e le relative impostazioni a livello utente soddisfano le seguenti condizioni:

1. L'MFA viene impostata su opzionale o obbligatoria nel pool di utenti.
2. L'utente ha un `phone_number` attributo `email` or valido oppure ha configurato un'app di autenticazione per TOTP.

3. È attivo almeno un fattore MFA.
4. Un fattore MFA è impostato come preferito.

Impedire l'uso dello stesso fattore per l'accesso e l'MFA

È possibile configurare il pool di utenti in modo da rendere un fattore di accesso l'unica opzione di accesso e MFA disponibile per alcuni o tutti gli utenti. Questo risultato può verificarsi quando il tuo caso di accesso principale è l'utilizzo di messaggi di posta elettronica o SMS-message password monouso (OTP). L'MFA preferita di un utente potrebbe avere lo stesso tipo di fattore di accesso nelle seguenti condizioni:

- L'MFA è richiesta nel pool di utenti.
- Email e SMS OTP sono disponibili opzioni di accesso e MFA nel pool di utenti.
- L'utente accede con un messaggio di posta elettronica o SMS OTP.
- Hanno un attributo di indirizzo e-mail ma nessun attributo di numero di telefono o un attributo di numero di telefono ma nessun attributo di indirizzo e-mail.

In questo scenario, l'utente può accedere con un'e-mail OTP e completare la MFA con un'e-mail OTP. Questa opzione annulla la funzione essenziale della MFA. Gli utenti che accedono con password monouso devono poter utilizzare metodi di consegna diversi per l'accesso rispetto alla MFA. Quando gli utenti hanno sia l'opzione SMS che quella e-mail, Amazon Cognito assegna automaticamente un fattore diverso. Ad esempio, quando un utente accede con l'e-mail OTP, la sua MFA preferita è SMS OTP.

Esegui i passaggi seguenti per affrontare l'autenticazione con lo stesso fattore quando il tuo pool di utenti supporta l'autenticazione OTP sia per l'accesso che per l'MFA.

1. Abilita l'OTP per e-mail e SMS come fattori di accesso.
2. Abilita l'OTP per e-mail e SMS come fattori MFA.
3. Raccogliere

Impostazioni del pool di utenti e relativo effetto sulle opzioni MFA

La configurazione del pool di utenti influenza i metodi MFA che gli utenti possono scegliere. Di seguito sono riportate alcune impostazioni del pool di utenti che influiscono sulla capacità degli utenti di configurare l'MFA.

- Nella configurazione di Multi-factor autenticazione nel Sign-inmenu della console Amazon Cognito, puoi impostare l'autenticazione MFA su opzionale o obbligatoria oppure disattivarla. L'equivalente API di questa impostazione è il MfaConfigurationparametro di `CreateUserPool`Update`UserPool`, e. `SetUserPoolMfaConfig`

  Inoltre, nella configurazione dell'Multi-factor autenticazione, l'impostazione dei metodi MFA determina i fattori MFA che gli utenti possono configurare. L'equivalente API di questa impostazione è l'SetUserPoolMfaConfigoperazione.
- Nel Sign-inmenu, alla voce Recupero dell'account utente, puoi configurare il modo in cui il tuo pool di utenti invia messaggi agli utenti che dimenticano la password. Il metodo MFA di un utente non può avere lo stesso metodo di consegna MFA del pool di utenti per i codici con password dimenticata. Il parametro API per il metodo di consegna della password dimenticata è il parametro and. AccountRecoverySetting`CreateUserPool`Update`UserPool`

  Ad esempio, gli utenti non possono configurare l'MFA e-mail quando l'opzione di ripristino è Solo e-mail. Questo perché non è possibile abilitare l'MFA e-mail e impostare l'opzione di ripristino su Email only nello stesso pool di utenti. Se imposti questa opzione su Invia e-mail se disponibile, altrimenti SMS, l'e-mail è l'opzione di ripristino prioritaria, ma il tuo pool di utenti può ricorrere al messaggio SMS quando un utente non è idoneo per il ripristino dei messaggi e-mail. In questo scenario, gli utenti possono impostare l'MFA e-mail come preferenza e possono ricevere un messaggio SMS solo quando tentano di reimpostare la password.
- Se si imposta un solo metodo MFA come disponibile, non è necessario gestire le preferenze MFA dell'utente.
- Una configurazione SMS attiva rende automaticamente i messaggi SMS un metodo MFA disponibile nel pool di utenti.

  Una configurazione e-mail attiva con le tue risorse Amazon SES in un pool di utenti e il piano di funzionalità Essentials o Plus rendono automaticamente i messaggi e-mail un metodo MFA disponibile nel tuo pool di utenti.
- Quando si imposta l'MFA su obbligatorio in un pool di utenti, gli utenti non possono abilitare o disabilitare alcun metodo MFA. È possibile impostare solo un metodo preferito.
- Quando si imposta l'MFA su opzionale in un pool di utenti, l'accesso gestito non richiede agli utenti di configurare l'MFA, ma richiede agli utenti un codice MFA quando hanno un metodo di MFA preferito.
- Quando attivi la protezione dalle minacce e configuri le risposte di autenticazione adattiva in modalità con funzionalità complete, l'MFA deve essere opzionale nel tuo pool di utenti. Una delle

opzioni di risposta con l'autenticazione adattiva consiste nel richiedere l'autenticazione MFA per un utente il cui tentativo di accesso viene valutato in modo da contenere un livello di rischio.

L'impostazione Attributi richiesti nel Sign-upmenu della console determina se gli utenti devono fornire un indirizzo e-mail o un numero di telefono per registrarsi all'applicazione. I messaggi e-mail e SMS diventano fattori MFA idonei quando un utente dispone dell'attributo corrispondente. Il parametro Schema di `CreateUserPool` imposta gli attributi come richiesto.

- Quando imposti l'autenticazione a più fattori su richiesta in un pool di utenti e un utente accede con accesso gestito, Amazon Cognito gli chiede di selezionare un metodo MFA tra i metodi disponibili per il tuo pool di utenti. L'accesso gestito gestisce la raccolta di un indirizzo e-mail o di un numero di telefono e la configurazione di TOTP. Il diagramma che segue illustra la logica alla base delle opzioni che Amazon Cognito presenta agli utenti.

Configurazione delle preferenze MFA per gli utenti

È possibile configurare le preferenze MFA per gli utenti in un modello self-service con autorizzazione tramite token di accesso o in un modello gestito dall'amministratore con operazioni API amministrative. Queste operazioni abilitano o disabilitano i metodi MFA e impostano uno dei diversi metodi come opzione preferita. Dopo che l'utente ha impostato una preferenza MFA, Amazon Cognito gli chiede al momento dell'accesso di fornire un codice dal metodo di MFA preferito. Gli utenti che non hanno impostato una preferenza ricevono la richiesta di scegliere un metodo preferito in una sfida. `SELECT_MFA_TYPE`

- In un modello self-service utente o in un'applicazione pubblica SetUserMfaPreference, autorizzata con un token di accesso dell'utente registrato, imposta la configurazione MFA.
- In un'applicazione gestita dall'amministratore o riservata, autorizzata con AWS credenziali amministrative AdminSetUserPreference, imposta la configurazione MFA.

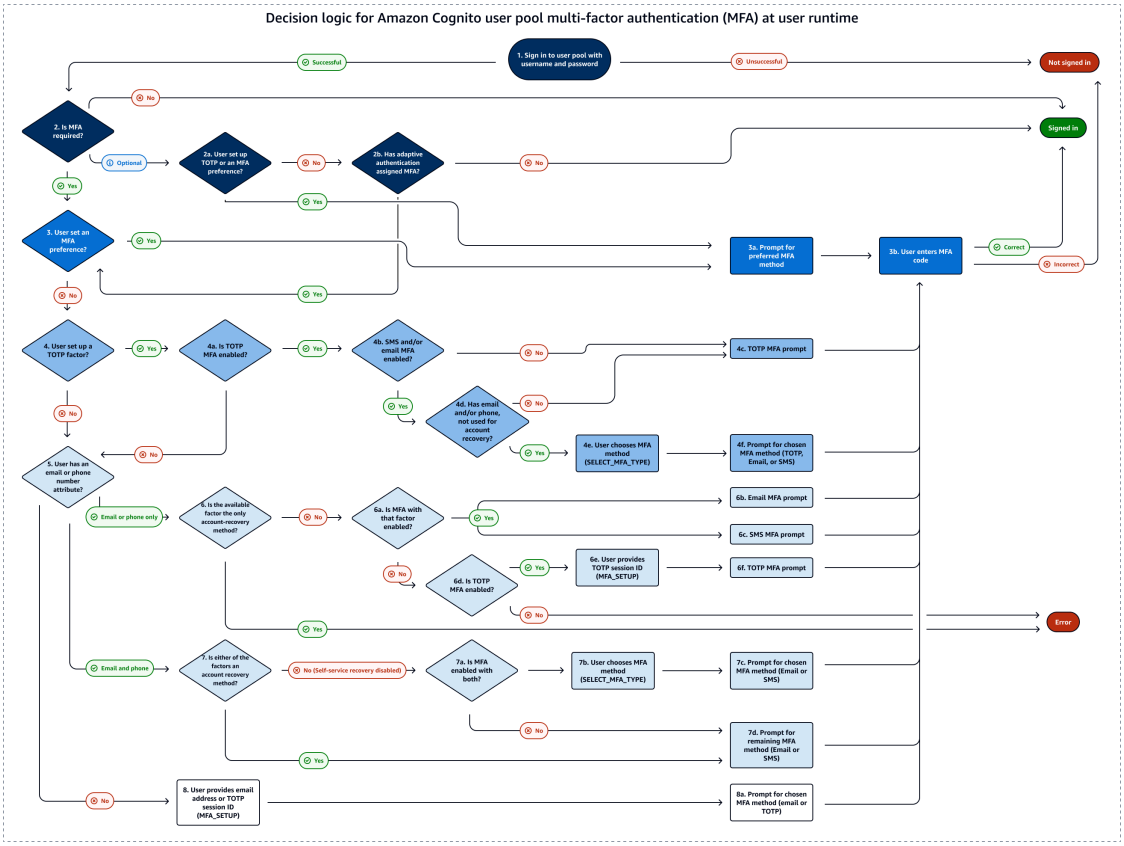
Puoi anche impostare le preferenze MFA degli utenti dal menu Utenti della console Amazon Cognito. Per ulteriori informazioni sui modelli di autenticazione pubblici e riservati nell'API dei pool di utenti di Amazon Cognito, consulta. Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite

Dettagli della logica MFA in fase di esecuzione dell'utente

Per determinare i passaggi da eseguire quando gli utenti accedono, il pool di utenti valuta le preferenze MFA degli utenti, gli attributi utente, l'impostazione MFA del pool di utenti, le azioni di

protezione dalle minacce e le impostazioni di ripristino dell'account self-service. Quindi accede agli utenti, richiede loro di scegliere un metodo di MFA, richiede loro di configurare un metodo MFA o richiede loro di utilizzare la MFA. Per configurare un metodo MFA, gli utenti devono fornire un indirizzo e-mail o un numero di telefono o registrare un autenticatore TOTP. Possono anche configurare le opzioni MFA e registrare in anticipo un'opzione preferita. Il diagramma seguente elenca gli effetti dettagliati della configurazione del pool di utenti sui tentativi di accesso immediatamente dopo la registrazione iniziale.

La logica illustrata qui si applica alle SDK-based applicazioni e all'accesso gestito, ma è meno visibile nell'accesso gestito. Quando risolvi i problemi relativi alla MFA, procedi a ritroso dai risultati degli utenti alle configurazioni del profilo utente e del pool di utenti che hanno contribuito alla decisione.



L'elenco seguente corrisponde alla numerazione nel diagramma della logica decisionale e descrive ogni fase in dettaglio. A



indica l'avvenuta autenticazione e la conclusione del flusso. A



indica che l'autenticazione non è riuscita.

1. Un utente presenta il proprio nome utente o nome utente e password nella schermata di accesso. Se non presenta credenziali valide, la sua richiesta di accesso viene rifiutata.
2. Se riescono ad autenticare nome utente e password, stabilisci se l'autenticazione MFA è obbligatoria, facoltativa o disattivata. Se è disattivata, il nome utente e la password corretti determinano una corretta autenticazione.

   a. Se l'autenticazione MFA è facoltativa, determina se l'utente ha precedentemente configurato un autenticatore TOTP. Se hanno impostato TOTP, richiedi TOTP MFA. Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.

   b. Determina se la funzionalità di autenticazione adattiva della protezione dalle minacce ha richiesto all'utente di configurare la MFA. Se non ha assegnato l'MFA, l'utente ha effettuato l'accesso.

3. Se l'MFA è richiesta o l'autenticazione adattiva ha assegnato l'MFA, determina se l'utente ha impostato un fattore MFA come abilitato e preferito. Se lo hanno fatto, richiedi l'MFA con quel fattore. Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.

4. Se l'utente non ha impostato una preferenza MFA, determina se l'utente ha registrato un autenticatore TOTP.

   a. Se l'utente ha registrato un autenticatore TOTP, determina se TOTP MFA è disponibile nel pool di utenti (TOTP MFA può essere disabilitato dopo che gli utenti hanno precedentemente configurato gli autenticatori).

   b. Determina se nel pool di utenti sono disponibili anche messaggi di posta elettronica o SMS-message MFA.

   c. Se non sono disponibili né MFA per e-mail né SMS, richiedi all'utente l'MFA TOTP. Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.

   d. Se l'MFA tramite e-mail o SMS è disponibile, stabilisci se l'utente dispone dell'attributo `email` o `phone_number` corrispondente. In tal caso, hanno a disposizione qualsiasi attributo che non sia il metodo principale per il ripristino degli account in modalità self-service e sia abilitato per l'MFA.

e. Segnala all'utente una `SELECT_MFA_TYPE` sfida con `MFAS_CAN_SELECT` opzioni che includono TOTP e i fattori MFA disponibili per SMS o e-mail.

f. Richiedi all'utente il fattore che ha selezionato in risposta alla sfida. `SELECT_MFA_TYPE` Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.



5. Se l'utente non ha registrato un autenticatore TOTP o se lo ha ma l'MFA TOTP è attualmente disabilitato, determina se l'utente dispone di un attributo or. `email` `phone_number`

6. Se l'utente ha solo un indirizzo e-mail o solo un numero di telefono, stabilisci se tale attributo è anche il metodo implementato dal pool di utenti per inviare messaggi di ripristino dell'account per la reimpostazione della password. Se vero, non possono completare l'accesso con la MFA richiesta e Amazon Cognito restituisce un errore. Per attivare l'accesso per questo utente, devi aggiungere un attributo non di ripristino o registrare un autenticatore TOTP per lui.



a. Se è disponibile un indirizzo e-mail o un numero di telefono non ripristinabile, stabilisci se il fattore MFA e-mail o SMS corrispondente è abilitato.

b. Se hanno un attributo di indirizzo e-mail non ripristinabile e l'MFA e-mail è abilitata, richiedi loro una richiesta. `EMAIL_OTP` Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.



c. Se hanno un attributo di numero di telefono non ripristinabile e l'MFA SMS è abilitata, inviagli un messaggio di sfida. `SMS_MFA` Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.



d. Se non hanno un attributo idoneo per un fattore MFA e-mail o SMS abilitato, stabilisci se TOTP MFA è abilitato. Se la MFA TOTP è disabilitata, non possono completare l'accesso con la MFA richiesta e Amazon Cognito restituisce un errore. Per attivare l'accesso per questo utente, devi aggiungere un attributo non di ripristino o registrare un autenticatore TOTP per lui.



> **Note**
>
> Questo passaggio è già stato valutato come No se l'utente dispone di un autenticatore TOTP ma l'MFA TOTP è disabilitato.

e. Se la MFA TOTP è abilitata, presenta all'utente `MFA_SETUP` una sfida tra `SOFTWARE_TOKEN_MFA` le opzioni. `MFAS_CAN_SETUP` Per completare questa sfida, devi registrare separatamente un autenticatore TOTP per l'utente e con cui rispondere. `"ChallengeName": "MFA_SETUP", "ChallengeResponses": {"USERNAME": "[username]", "SESSION": "[Session ID from VerifySoftwareToken]}"`

f. Dopo che l'utente ha risposto alla `MFA_SETUP` sfida con il token di sessione contenuto in una VerifySoftwareTokenrichiesta, richiedigli una sfida. `SOFTWARE_TOKEN_MFA` Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.



7. Se l'utente dispone sia di un indirizzo e-mail che di un numero di telefono, determina quale attributo, se presente, è il metodo principale per i messaggi di ripristino dell'account per la reimpostazione della password.

   a. Se il ripristino self-service dell'account è disabilitato, entrambi gli attributi possono essere utilizzati per l'MFA. Determina se uno o entrambi i fattori MFA e-mail e SMS sono abilitati.

   b. Se entrambi gli attributi sono abilitati come fattore MFA, richiedi all'utente una `SELECT_MFA_TYPE` sfida con `MFAS_CAN_SELECT` le opzioni e. `SMS_MFA` `EMAIL_OTP`

   c. Richiedete loro il fattore che scelgono in risposta alla sfida. `SELECT_MFA_TYPE` Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.



   d. Se solo un attributo è un fattore MFA idoneo, proponi loro una sfida per il fattore rimanente. Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.



   Questo risultato si verifica nei seguenti scenari.

   i. Se dispongono di `phone_number` attributi `email` e, l'MFA per SMS ed e-mail sono abilitati e il metodo principale di ripristino dell'account è tramite e-mail o messaggio SMS.

   ii. Se dispongono `email` di `phone_number` attributi, è abilitato solo l'MFA SMS o l'MFA e-mail e il ripristino self-service dell'account è disabilitato.

8. Se l'utente non ha registrato un autenticatore TOTP e non ha né l'`phone_number`attributo `email` nor, richiedigli una sfida. `MFA_SETUP` L'elenco `MFAS_CAN_SETUP` include tutti i fattori MFA abilitati nel pool di utenti che non sono l'opzione principale di ripristino dell'account. Possono rispondere a questa sfida con la posta elettronica o `ChallengeResponses` l'MFA TOTP. Per configurare SMS MFA, aggiungi separatamente un attributo di numero di telefono e riavvia l'autenticazione.

   Per TOTP MFA, rispondi con. `"ChallengeName": "MFA_SETUP", "ChallengeResponses": {"USERNAME": "[username]", "SESSION": "[Session ID from VerifySoftwareToken]"}`

   Per la MFA via e-mail, rispondi con. `"ChallengeName": "MFA_SETUP", "ChallengeResponses": {"USERNAME": "[username]", "email": "[user's email address]"}`

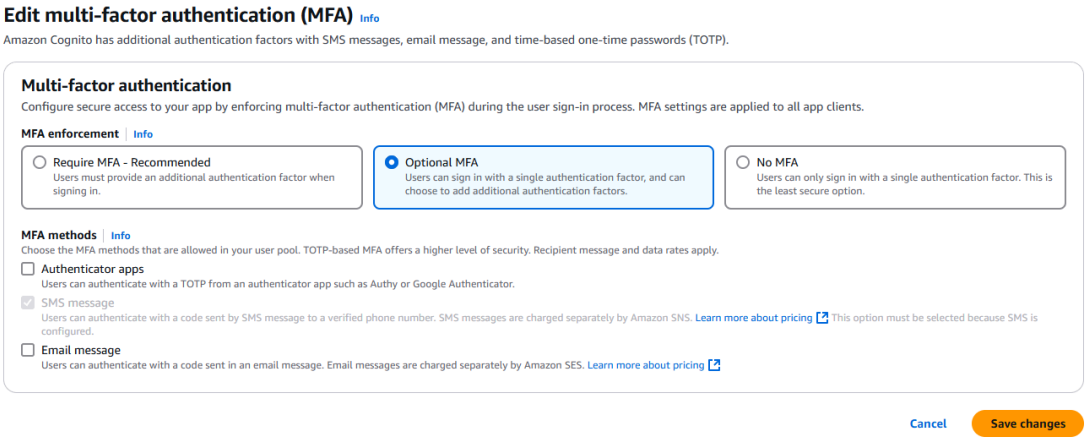
   a. Chiedi loro di indicare il fattore che scelgono in risposta alla `SELECT_MFA_TYPE` sfida. Se rispondono con successo alla sfida MFA, hanno effettuato l'accesso.

Configurare un pool di utenti per l'autenticazione a più fattori

Puoi configurare l'MFA nella console Amazon Cognito o con il funzionamento SetUserPoolMfaConfigdell'API e i metodi SDK.

Come configurare l'autenticazione MFA nella console Amazon Cognito

1. Accedi alla console Amazon Cognito.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu. Sign-in Individua Multi-factorl'autenticazione e scegli Modifica.
5. Scegli il metodo di applicazione dell'MFA che desideri utilizzare con il bacino d'utenza.



a. Richiedi MFA. Tutti gli utenti del tuo pool di utenti devono accedere con un codice SMS, e-mail o password monouso (TOTP) aggiuntivo come fattore di autenticazione aggiuntivo.

b. MFA opzionale. Puoi offrire ai tuoi utenti la possibilità di registrare un fattore di accesso aggiuntivo, ma consentire comunque agli utenti che non hanno configurato l'MFA di accedere. Scegli questa opzione se utilizzi l'autenticazione adattiva. Per ulteriori informazioni sull'autenticazione adattiva, consulta Sicurezza avanzata con protezione dalle minacce.

c. Nessuna MFA. Gli utenti non possono registrare un ulteriore fattore di accesso.

6. Scegli i metodi MFA da supportare nell'app. Puoi impostare messaggi di posta elettronica, messaggi SMS o app TOTP-generating Authenticator come secondo fattore.

7. Se utilizzi gli SMS come secondo fattore e non disponi di un ruolo IAM configurato per l'utilizzo con Amazon Simple Notification Service (Amazon SNS) per gli SMS, puoi crearne uno nella console. Nel menu Metodi di autenticazione per il tuo pool di utenti, individua gli SMS e scegli Modifica. Puoi anche utilizzare un ruolo esistente che consente ad Amazon Cognito di inviare messaggi SMS agli utenti per tuo conto. Per ulteriori informazioni, consulta IAM Roles (Ruoli IAM).

   Se utilizzi i messaggi e-mail come secondo fattore e non hai configurato un'identità di origine da utilizzare con Amazon Simple Email Service (Amazon SES) per i messaggi e-mail, creane una nella console. Devi scegliere l'opzione Invia e-mail con SES. Nel menu Metodi di autenticazione per il tuo pool di utenti, individua Email e scegli Modifica. Seleziona un indirizzo email FROM dalle identità verificate disponibili nell'elenco. Se scegli un dominio verificato, ad esempio`example.com`, devi anche configurare un nome mittente FROM nel dominio verificato, ad esempio. `admin-noreply@example.com`

8. Scegli Save changes (Salva modifiche).

MFA per SMS e messaggi e-mail

I messaggi MFA via SMS ed e-mail confermano che gli utenti hanno accesso a una destinazione dei messaggi prima di poter accedere. Confermano di avere accesso non solo a una password, ma anche ai messaggi SMS o alla casella di posta elettronica dell'utente originale. Amazon Cognito richiede agli utenti di fornire un codice breve che il pool di utenti ha inviato dopo aver fornito correttamente un nome utente e una password.

La MFA per SMS e messaggi e-mail non richiede alcuna configurazione aggiuntiva dopo che l'utente ha aggiunto un indirizzo e-mail o un numero di telefono al proprio profilo. Amazon Cognito può inviare messaggi a indirizzi e-mail e numeri di telefono non verificati. Quando un utente completa la prima MFA, Amazon Cognito contrassegna il proprio indirizzo e-mail o numero di telefono come verificato.

L'autenticazione MFA inizia quando un utente con MFA inserisce nome utente e password nell'applicazione. L'applicazione invia questi parametri iniziali in un metodo SDK che richiama una richiesta o API. InitiateAuthAdminInitiateAuth La risposta `ChallengeParameters` nell'API include un `CODE_DELIVERY_DESTINATION` valore che indica dove è stato inviato il codice di autorizzazione. Nell'applicazione, visualizza un modulo che richiede all'utente di controllare il proprio telefono e include un elemento di input per il codice. Quando inserisce il codice, invialo in una richiesta API di risposta alla sfida per completare la procedura di accesso.

Dopo che un utente con MFA ha effettuato l'accesso con nome utente e password nelle pagine di accesso gestite, gli viene richiesto automaticamente il codice MFA.

I pool di utenti inviano messaggi SMS per MFA e altre notifiche di Amazon Cognito con le risorse Amazon Simple Notification Service (Amazon SNS) presenti nelle tue. Account AWS Allo stesso modo, i pool di utenti inviano messaggi e-mail con le risorse Amazon Simple Email Service (Amazon SES) presenti nel tuo account. Questi servizi collegati comportano costi propri sulla AWS fattura per la consegna dei messaggi. Inoltre, prevedono requisiti aggiuntivi per l'invio di messaggi a volumi di produzione. Per ulteriori informazioni, consulta i seguenti collegamenti:

- Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito
- Prezzi SMS a livello mondiale
- Impostazioni e-mail per i bacini d'utenza di Amazon Cognito
- Prezzi di Amazon SES

Considerazioni sull'MFA per SMS e messaggi di posta elettronica

- Per consentire agli utenti di accedere con l'MFA e-mail, il pool di utenti deve disporre delle seguenti opzioni di configurazione:
  1. Hai il piano di funzionalità Plus o Essentials nel tuo pool di utenti. Per ulteriori informazioni, consulta Piani di funzionalità del pool di utenti.
  2. Il tuo pool di utenti invia messaggi e-mail con le tue risorse Amazon SES. Per ulteriori informazioni, consulta Configurazione e-mail di Amazon SES.
- Il codice MFA è valido per la durata della sessione del flusso di autenticazione impostata per il client dell'app.

  Imposta la durata di una sessione del flusso di autenticazione nella console Amazon Cognito nel menu App client quando modifichi il client dell'app. È possibile impostare la durata della sessione del flusso di autenticazione anche all'interno della richiesta API `CreateUserPoolClient` o `UpdateUserPoolClient`. Per ulteriori informazioni, consulta Un esempio di sessione di autenticazione.
- Quando un utente fornisce correttamente un codice da un messaggio SMS o e-mail inviato da Amazon Cognito a un numero di telefono o indirizzo e-mail non verificato, Amazon Cognito contrassegna l'attributo corrispondente come verificato.
- Affinché un utente possa apportare una modifica self-service al valore di un numero di telefono o di un indirizzo e-mail associato alla MFA, deve accedere e autorizzare la richiesta con un token di accesso. Se non riescono ad accedere al loro attuale numero di telefono o indirizzo e-mail, non possono accedere. Il team deve modificare questi valori con AWS le credenziali di amministratore nelle richieste AdminUpdateUserAttributesAPI.
- Dopo aver configurato gli SMS nel tuo pool di utenti, non puoi disabilitare i messaggi SMS come fattore MFA disponibile.

Autenticazione MFA con token di software TOTP

Quando configuri l'autenticazione MFA con token di software TOTP nel pool di utenti, l'utente accede con nome utente e password, quindi utilizza un TOTP per completare l'autenticazione. Dopo che l'utente imposta e verifica nome utente e password, può attivare un token software TOTP per l'autenticazione MFA. Se la tua app utilizza l'accesso gestito di Amazon Cognito per accedere agli utenti, l'utente invia nome utente e password, quindi invia la password TOTP in una pagina di accesso aggiuntiva.

Puoi attivare l'autenticazione MFA con TOTP per il bacino d'utenza nella console Amazon Cognito oppure utilizzando le operazioni API di Amazon Cognito. A livello di pool di utenti, puoi chiamare SetUserPoolMfaConfigper configurare l'MFA e abilitare TOTP MFA.

> **Note**
>
> Se l'autenticazione MFA con token software TOTP non è attivata per il bacino d'utenza, Amazon Cognito non può usare i token per associare o verificare gli utenti. In questo caso, gli utenti ricevono un'eccezione `SoftwareTokenMFANotFoundException` con la descrizione `Software Token MFA has not been enabled by the userPool`. Se in seguito disattivi l'autenticazione MFA tramite token software per il bacino d'utenza, gli utenti che hanno precedentemente associato e verificato un token TOTP possono continuare a utilizzarlo per l'MFA.

La configurazione del metodo TOTP per gli utenti è un processo in più fasi in cui l'utente riceve un codice segreto che convalida immettendo una password monouso. Quindi, puoi abilitare l'MFA con TOTP per l'utente o impostare la password TOTP come metodo preferito di MFA per l'utente.

Quando configuri il tuo pool di utenti per richiedere la MFA TOTP e gli utenti si iscrivono alla tua app nell'accesso gestito, Amazon Cognito automatizza il processo utente. Amazon Cognito richiede all'utente di scegliere un metodo di autenticazione MFA, visualizza un codice QR per configurare l'app Authenticator e verifica la registrazione MFA. Nei pool di utenti in cui si è consentito agli utenti di scegliere tra l'autenticazione MFA con SMS e con token di software TOTP, Amazon Cognito consente all'utente di scegliere uno dei metodi disponibili.

> **Important**
>
> Quando hai un ACL AWS WAF web associato a un pool di utenti e una regola nell'ACL web presenta un CAPTCHA, ciò può causare un errore irreversibile nella registrazione TOTP con accesso gestito. Per creare una regola che abbia un'azione CAPTCHA e non influisca sul TOTP dell'accesso gestito, consulta. Configurazione del AWS WAF ACL web per accesso gestito TOTP MFA Per ulteriori informazioni sulle liste ACL Web AWS WAF e Amazon Cognito, consulta Associa un AWS WAF ACL web con un pool di utenti.

Per implementare TOTP MFA in un'interfaccia utente personalizzata con AWS un SDK e l'API dei pool di utenti di Amazon Cognito, consulta. Configurazione dell'MFA TOTP per un utente

Per aggiungere l'autenticazione MFA al bacino d'utenza, consulta Aggiunta dell'autenticazione MFA a un bacino d'utenza.

Considerazioni e limitazioni riguardo all'MFA con TOTP

1. Amazon Cognito supporta l'MFA con token software tramite un'app di autenticazione che genera i codici TOTP. Amazon Cognito non supporta l'MFA basata su hardware.
2. Quando il tuo bacino d'utenza richiede una TOTP per un utente che non l'ha configurata, l'utente riceve un token di accesso monuso che la tua app può utilizzare per attivare l'MFA con TOTP per l'utente. I tentativi di accesso successivi non riescono finché l'utente non registra un ulteriore fattore di accesso con TOTP.
   - Un utente che si iscrive al tuo pool di utenti con l'operazione `SignUp` API o tramite accesso gestito riceve token monouso quando completa la registrazione.
   - Dopo aver creato un utente e aver impostato la password iniziale, Amazon Cognito emette token monouso dall'accesso gestito all'utente. Se imposti una password permanente per l'utente, Amazon Cognito emette token monouso quando l'utente accede per la prima volta.
   - Amazon Cognito non rilascia token monouso a un utente creato dall'amministratore che accede con le operazioni o API. InitiateAuthAdminInitiateAuth Dopo che l'utente ha impostato correttamente la password iniziale o che tu hai impostato una password permanente per l'utente, Amazon Cognito richiede immediatamente all'utente di configurare l'MFA.
3. Se un utente in un pool di utenti che richiede l'autenticazione a più fattori ha già ricevuto un token di accesso monouso ma non ha configurato TOTP MFA, l'utente non può accedere con l'accesso gestito finché non ha configurato l'MFA. Invece del token di accesso, puoi utilizzare il valore di `session` risposta di una `MFA_SETUP` sfida o contenuto in una richiesta. InitiateAuthAdminInitiateAuthAssociateSoftwareToken
4. Se gli utenti hanno impostato la TOTP, possono utilizzarla per l'MFA anche se in seguito la disattivi per il bacino d'utenza.
5. Amazon Cognito accetta i TOTP solo dalle app di autenticazione che generano codici con la funzione hash. HMAC-SHA1 I codici generati con SHA-256 l'hashing restituiscono un errore. `Code mismatch`

Configurazione dell'MFA TOTP per un utente

Quando un utente accede per la prima volta, l'app utilizza il token di accesso monouso per generare la chiave privata TOTP e presentarla all'utente in formato testo o codice QR. L'utente configura

la propria app di autenticazione e fornisce un TOTP per i successivi tentativi di accesso. L'app o l'accesso gestito presentano il TOTP ad Amazon Cognito nelle risposte alle sfide MFA.

In alcune circostanze, l'accesso gestito richiede ai nuovi utenti di configurare un autenticatore TOTP. Per ulteriori informazioni, consulta. Dettagli della logica MFA in fase di esecuzione dell'utente

Argomenti

- Associazione del token di software TOTP
- Verifica del token TOTP
- Accesso usando l'MFA con TOTP
- Rimozione del token TOTP

Associazione del token di software TOTP

Per associare il token TOTP, invia al tuo utente un codice segreto che deve convalidare con una password monouso. Per l'associazione del token sono necessari tre passaggi.

1. Quando l'utente sceglie il token software TOTP MFA, chiama AssociateSoftwareTokenper restituire un codice chiave segreto condiviso generato univoco per l'account utente. È possibile autorizzare AssociateSoftwareToken con un token di accesso o una stringa di sessione.
2. La tua app presenta all'utente la chiave privata o un codice QR generato dalla chiave privata. L'utente deve inserire la chiave in un' TOTP-generating app come Google Authenticator, scansionando il codice QR generato dall'applicazione dalla chiave privata o inserendo manualmente la chiave.
3. L'utente inserisce la chiave o scansiona il codice QR in un'app di autenticazione come Google Authenticator che inizia a generare codici.

Verifica del token TOTP

Poi, verifica del token TOTP. Richiedi codici di esempio dal tuo utente e forniscili al servizio Amazon Cognito per confermare che l'utente sta generando correttamente i codici TOTP, come indicato di seguito.

1. L'app richiede all'utente un codice per dimostrare di aver configurato correttamente l'app di autenticazione.
2. L'app di autenticazione dell'utente mostra una password temporanea. L'app di autenticazione crea la password in base alla chiave segreta che hai fornito all'utente.

3. L'utente inserisce la password temporanea. La tua app passa la password temporanea ad Amazon Cognito in una Richiesta API `VerifySoftwareToken`.
4. Amazon Cognito ha mantenuto la chiave segreta associata all'utente, genera una TOTP e la confronta con quella fornita dall'utente. Se corrispondono,`VerifySoftwareToken` restituisce una risposta `SUCCESS`.
5. Amazon Cognito associa il fattore TOTP all'utente.
6. Se l'operazione `VerifySoftwareToken` restituisce una risposta `ERROR`, assicurati che il segnale di clock dell'utente sia corretto e che l'utente non abbia superato il numero massimo di tentativi. Amazon Cognito accetta token TOTP entro 30 secondi prima o dopo il tentativo, per tenere conto del minore sfasamento del segnal di clock. Una volta risolto il problema, riprova a VerifySoftwareToken eseguire l'operazione.

Accesso usando l'MFA con TOTP

A questo punto, l'utente accede con la password una tantum a tempo. Il procedimento è riportato di seguito.

1. L'utente inserisce il nome utente e la password per accedere all'app client.
2. Viene invocata la richiesta dell'MFA con TOTP e l'app chiede all'utente di immettere una password temporanea.
3. L'utente ottiene la password temporanea da un' TOTP-generatingapp associata.
4. L'utente inserisce il codice TOTP nell'app client. L'app notifica al servizio Amazon Cognito di verificarlo. Per ogni accesso, RespondToAuthChallengedeve essere chiamato per ottenere una risposta alla nuova sfida di autenticazione TOTP.
5. Se il token è verificato da Amazon Cognito, l'accesso viene completato correttamente e l'utente procede nel flusso di autenticazione.

Rimozione del token TOTP

Infine, l'app dovrebbe consentire all'utente di disattivare la configurazione TOTP. Per eliminare il token software TOTP dell'utente, usa. AdminDeleteSoftwareToken Dopo aver eliminato il token software dell'utente, quest'ultimo non può più accedere con TOTP MFA e può registrare un nuovo token software. Utilizza questa operazione quando l'utente perde l'accesso al proprio TOTP-generating dispositivo, ad esempio un telefono smarrito o ripristinato. Quando il tuo pool di utenti richiede l'autenticazione a più fattori e non è disponibile nessun altro fattore, Amazon Cognito

restituisce `MFA_SETUP` una sfida al successivo accesso dell'utente. Quando l'utente dispone di un altro fattore disponibile, ad esempio l'MFA dei messaggi SMS, l'accesso torna a tale fattore.

Se l'utente ha ancora accesso al proprio TOTP-generating dispositivo, non è necessario eliminare il token software. Amazon Cognito dissocia il token software esistente quando l'utente ne verifica uno nuovo. Per disattivare TOTP MFA per il tuo utente, chiama SetUserMFAPreferenceper modificare l'utente in modo che non utilizzi MFA o solo SMS MFA.

1. Crea un'interfaccia nell'app per gli utenti che desiderano reimpostare MFA. Richiedi a un utente in questa interfaccia di inserire la password.
2. Se Amazon Cognito restituisce una sfida MFA TOTP, aggiorna la preferenza MFA dell'utente con. SetUserMFAPreference
3. Nell'app, comunica all'utente che MFA è stata disattivata e chiedi di effettuare nuovamente l'accesso.

Configurazione del AWS WAF ACL web per accesso gestito TOTP MFA

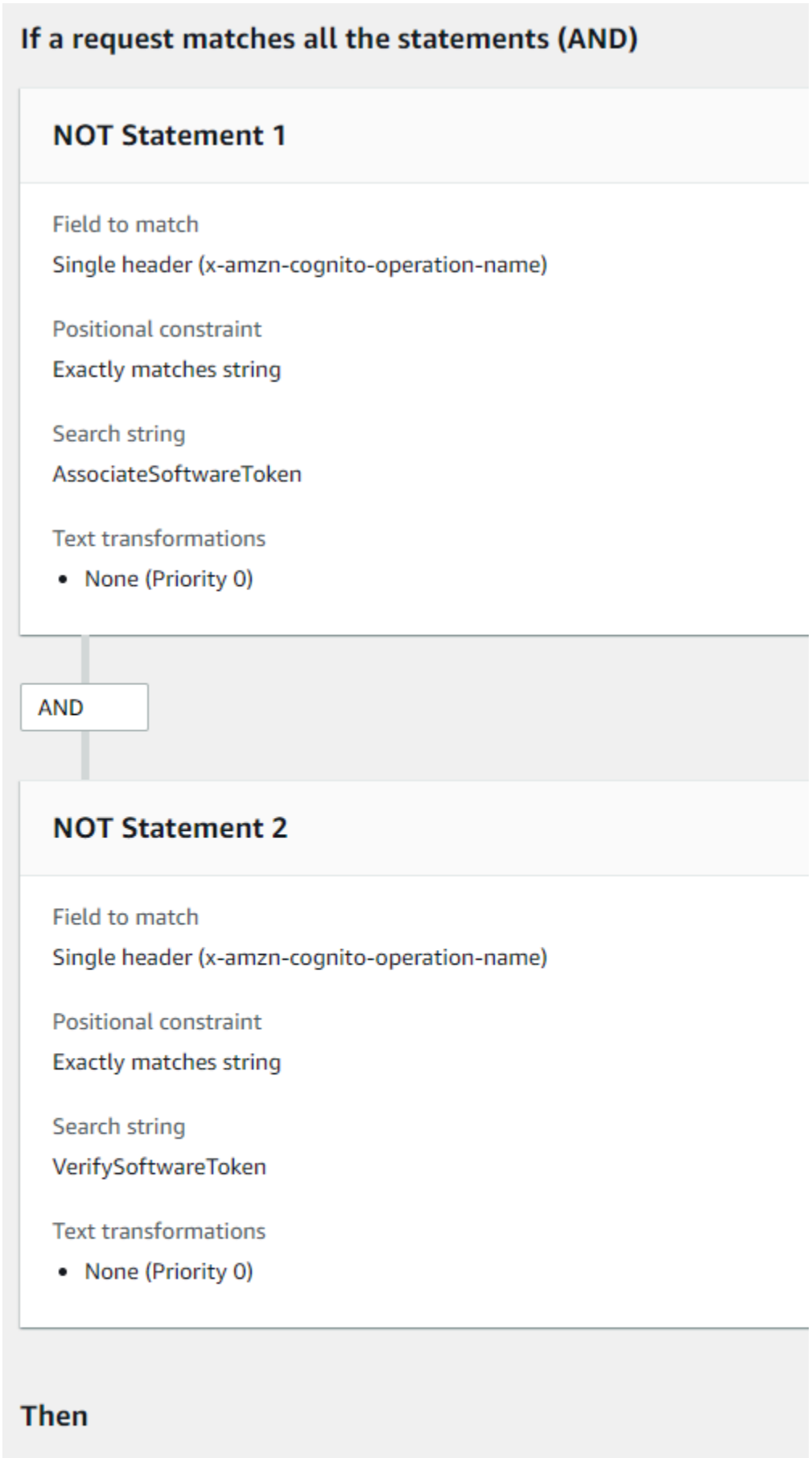
Quando hai un ACL AWS WAF web associato a un pool di utenti e una regola nell'ACL web presenta un CAPTCHA, ciò può causare un errore irreversibile nella registrazione TOTP con accesso gestito. AWS WAF Le regole CAPTCHA hanno questo effetto solo sull'MFA TOTP nell'accesso gestito e nella classica interfaccia utente ospitata. L'autenticazione MFA con SMS rimane invariata.

Amazon Cognito visualizza il seguente errore quando la regola CAPTCHA non consente a un utente di completare la configurazione dell'autenticazione MFA con token di software TOTP.

Request not allowed due to WAF captcha (Richiesta non consentita a causa di un'azione CAPTCHA WAF).

Questo errore si verifica quando viene AWS WAF richiesto un CAPTCHA in risposta alle AssociateSoftwareTokenrichieste VerifySoftwareTokenAPI effettuate in background dal pool di utenti. Per creare una regola che abbia un'azione CAPTCHA e non influisca sul TOTP nelle pagine di accesso gestite, escludi i valori di `x-amzn-cognito-operation-name` intestazione di `AssociateSoftwareToken` e `VerifySoftwareToken` dall'azione CAPTCHA nella regola.

La schermata seguente mostra una AWS WAF regola di esempio che applica un'azione CAPTCHA a tutte le richieste che non hanno un valore di intestazione pari a o. `x-amzn-cognito-operation-name` `AssociateSoftwareToken` `VerifySoftwareToken`


If a request matches all the statements (AND)
NOT Statement 1
Field to match
Single header (x-amzn-cognito-operation-name)
Positional constraint
Exactly matches string
Search string
AssociateSoftwareToken
Text transformations
None (Priority 0)
AND
NOT Statement 2
Field to match
Single header (x-amzn-cognito-operation-name)
Positional constraint
Exactly matches string
Search string
VerifySoftwareToken
Text transformations
None (Priority 0)
Then
Action
The action to take when a web request matches the rule statement.

Per ulteriori informazioni sugli ACL AWS WAF Web e Amazon Cognito, consulta. Associa un AWS WAF ACL web con un pool di utenti

Livelli di autenticazione con attestazioni ACR e AMR

Amazon Cognito è in grado di registrare il livello di autenticazione di un utente e i metodi utilizzati. Fornisci un livello di autenticazione target come input quando un utente accede e Amazon Cognito riporta il livello raggiunto dall'utente e i metodi che ha completato come attestazioni nei token emessi. Queste affermazioni si applicano a qualsiasi accesso, incluso l'accesso iniziale di un utente.

Si richiede un livello di autenticazione con il `acr_values` parametro sugli endpoint di autorizzazione e accesso gestiti o con il `TARGET_ACR_VALUES` parametro equivalente nel flusso API. `USER_AUTH` Entrambi accettano un elenco ordinato di livelli di autenticazione di destinazione. Puoi aggiungere il `max_age` parametro opzionale (`MAX_AGE`nell'API) per richiedere che l'utente si sia autenticato entro una finestra temporale recente. In risposta a questi input, Amazon Cognito aggiunge due affermazioni ai token emessi:

- L'`acr`attestazione è una stringa che indica, ad esempio, il livello di autenticazione soddisfatto dall'utente. `urn:cognito:loa:3`
- L'`amr`attestazione è una matrice di stringhe che indica, ad esempio`["pwd", "otp", "mfa"]`, i metodi di autenticazione completati dall'utente.

Le `amr` affermazioni `acr` and seguono le definizioni nella specifica OpenID Connect Core. Amazon Cognito utilizza i propri URI a livello ACR. Puoi tradurre i livelli di Amazon Cognito in qualsiasi standard richiesto dalla tua applicazione.

Step-up l'autenticazione è un caso d'uso comune per queste affermazioni. Nell'autenticazione step-up, l'applicazione richiede che l'utente si autentichi con un metodo più efficace prima di eseguire un'operazione sensibile. Per decidere se richiedere o meno l'autorizzazione, l'applicazione confronta il `acr` livello e l'`auth_time`attestazione contenuti nel token ID dell'utente con il livello e l'aggiornamento richiesti dall'operazione e le richieste di step-up solo quando non sono già state soddisfatte. Quando è necessario eseguire l'installazione, reindirizza l'utente all'endpoint di autorizzazione all'accesso gestito o all'endpoint di accesso con, oppure chiama il flusso API con. `acr_values` `USER_AUTH` `TARGET_ACR_VALUES` Per un esempio, vedi lo scenario di autenticazione. Step-up

Puoi richiedere ACR e AMR da qualsiasi punto di accesso ad Amazon Cognito. I meccanismi di ogni punto di ingresso sono documentati con quel punto di ingresso.

- Accesso a SDK e API: richiedi un livello target nel `USER_AUTH` flusso con i parametri and. `TARGET_ACR_VALUES` `MAX_AGE` Per i parametri e il comportamento avanzato, consulta. Step-up autenticazione con il flusso USER_AUTH
- Accesso gestito e interfaccia utente ospitata: passa i `max_age` parametri `acr_values` and sugli endpoint di autorizzazione e accesso. Per i dettagli, consulta l'endpoint di autorizzazione e l'endpoint di accesso.
- Accesso federato: Amazon Cognito ricava i valori ACR e AMR di un utente federato dalla risposta del provider di identità (IdP) e li mappa con il campo. `AcrMapping` Per i provider di identità OIDC, consulta. Step-up autenticazione per utenti federati

La richiesta dell'autenticazione avanzata richiede il piano di funzionalità Essentials o Plus. Per ulteriori informazioni, consulta Requisiti del piano di funzionalità.

Argomenti

- Livelli e metodi di autenticazione
- Configurazione dei nomi dei livelli ACR
- Requisiti del piano di funzionalità
- Le tue responsabilità in materia di autenticazione avanzata

Livelli e metodi di autenticazione

Amazon Cognito valuta i fattori di autenticazione completati da un utente ed esprime il risultato come un livello di autenticazione nell'attestazione e un elenco di `acr` metodi di autenticazione nell'attestazione. `amr`

Livelli di autenticazione ACR

Amazon Cognito definisce quattro livelli di autenticazione fissi. Le combinazioni di fattori di autenticazione che soddisfano ogni livello sono fisse e non è possibile modificarle. Puoi personalizzare solo il nome URI di ogni livello per il tuo pool di utenti. Per ulteriori informazioni, consulta Configurazione dei nomi dei livelli ACR.

Amazon Cognito assegna il livello di autenticazione più elevato che i fattori completati dell'utente soddisfano pienamente.

Livelli ACR e combinazioni di fattori soddisfacenti

Livello ACR	Valore ACR predefinito	Combinazioni di fattori soddisfacenti
1	`urn:cognito:loa:1`	Solo password
2	`urn:cognito:loa:2`	Un'unica password monouso inviata tramite SMS o e-mail. Il livello 2 non include le password
3	`urn:cognito:loa:3`	Una passkey (piattaforma o roaming) a sé stante; o una password con un codice monouso SMS; o una password con un codice monouso e-mail
4	`urn:cognito:loa:4`	Una password con una password monouso basata sul tempo (TOTP) fornita da un'app di autenticazione

Il livello 1 rappresenta l'autenticazione tramite sola password. La federazione non soddisfa alcun livello indicato in questa tabella. Invece, Amazon Cognito ricava il livello di autenticazione di un utente federato dalla risposta del provider di identità (IdP). Per ulteriori informazioni, consulta Step-up autenticazione per utenti federati.

Valori AMR

L'`amr`affermazione è un elenco di stringhe. Ogni fattore di autenticazione completato dall'utente aggiunge il proprio valore all'elenco. Amazon Cognito può aggiungere i seguenti valori.

Valori AMR

Metodo di autenticazione	Valore AMR	Note
Password	`pwd`	Password-family accesso, inclusa Secure Remote Password (SRP)
TOTP da un'app di autentica zione	`otp`	Una password monouso basata sul tempo fornita da un'app, ad esempio un'app di autenticazione

Metodo di autenticazione	Valore AMR	Note
Codice monouso SMS	`sms`	Un codice monouso inviato tramite SMS
Codice monouso per e-mail	`email_otp`	Un codice monouso inviato via e-mail
Passkey con un autenticatore di piattaforma	`swk`	Una chiave associata al software, ad esempio una passkey protetta da un dispositivo biometrico
Passkey con un autenticatore di roaming	`hwk`	Una chiave legata all'hardware, ad esempio una chiave di sicurezza
Multi-factor combinazione	`mfa`	Aggiunto quando l'utente completa due o più fattori distinti
Accesso federato	`fed`	Un segnaposto predefinito che Amazon Cognito aggiunge quando un utente esegue la federazione tramite un IdP che non supporta ACR e AMR o non restituisce valori riconosciuti da Amazon Cognito

Dichiarazioni relative ai token ACR e AMR

Amazon Cognito aggiunge `acr` e `amr` dichiara di accedere a token e token ID. L'applicazione ispeziona `acr` e `amr` dichiara il token di accesso o il token ID per valutare il livello di autenticazione. Quando un utente aggiorna i propri token, Amazon Cognito conserva gli `acr` stessi valori dei nuovi token di accesso `amr` e ID, in modo che all'utente non venga chiesto di eseguire nuovamente l'operazione dopo ogni aggiornamento. Per ulteriori informazioni sui contenuti dei token, consulta Utilizzo del token ID e Utilizzo del token di accesso.

L'esempio seguente mostra un token ID dopo che un utente si è autenticato con una password e un TOTP.

```
{
  "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
  "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
  "aud": "abc123def456",
  "exp": 1759348800,
```

```
  "iat": 1759345200,
  "auth_time": 1759345200,
  "token_use": "id",
  "acr": "urn:cognito:loa:4",
  "amr": ["pwd", "otp", "mfa"],
  "email": "user@example.com",
  "cognito:username": "a1b2c3d4-5678-90ab-cdef-1234567890ab"
}
```

Quando un utente aggiorna i propri token, Amazon Cognito imposta gli `acr` stessi valori sui nuovi token di accesso `amr` e ID. I valori vengono trasferiti senza modifiche. Questo comportamento offre un'esperienza più fluida perché agli utenti non viene chiesto di intervenire nuovamente dopo ogni aggiornamento.

L'aggiornamento di un token non reimposta l'attestazione. `auth_time` Se desideri forzare un utente a eseguire nuovamente l'autenticazione dopo un certo periodo di tempo, utilizza il `max_age` parametro, che confronta l'ora corrente con quella a `auth_time` prescindere dal numero di volte in cui l'utente ha aggiornato i token.

Quando un utente completa l'autenticazione avanzata, Amazon Cognito emette nuovi token di accesso, ID e aggiornamento con le dichiarazioni aggiornate `acr` e `amr` un nuovo periodo di validità.

Configurazione dei nomi dei livelli ACR

Puoi personalizzare il nome URI di ogni livello di autenticazione per il tuo pool di utenti con il `AcrConfiguration` campo. Configura questo campo quando crei o aggiorni un pool di utenti con l'UpdateUserPooloperazione CreateUserPoolo. Le combinazioni di fattori che soddisfano ogni livello rimangono fisse. Puoi personalizzare solo i nomi.

La configurazione di nomi di livello ACR personalizzati con il `AcrConfiguration` campo richiede il piano di funzionalità Essentials o Plus. Per ulteriori informazioni, consulta Requisiti del piano di funzionalità.

Il `AcrConfiguration` campo è una mappa con i tasti premuti. `Level1` `Level4` Ogni valore è un oggetto con una singola `AcrValue` stringa. È possibile sovrascrivere solo i livelli che si desidera personalizzare. La seguente richiesta di esempio personalizza i nomi dei livelli 2 e 3.

```
{
  "AcrConfiguration": {
    "Level2": { "AcrValue": "urn:my-standard:loa:2" },
```

```
    "Level3": { "AcrValue": "urn:my-standard:loa:3" }
  }
}
```

La risposta restituisce sempre la configurazione effettiva, con i nomi predefiniti uniti per tutti i livelli che non hai personalizzato.

```
{
  "AcrConfiguration": {
    "Level1": { "AcrValue": "urn:cognito:loa:1" },
    "Level2": { "AcrValue": "urn:my-standard:loa:2" },
    "Level3": { "AcrValue": "urn:my-standard:loa:3" },
    "Level4": { "AcrValue": "urn:cognito:loa:4" }
  }
}
```

Le seguenti regole di convalida si applicano a ciascuno di essi. `AcrValue`

- Ciascuno `AcrValue` deve essere unico in tutti e quattro i livelli. Sono inclusi i nomi predefiniti utilizzati per i livelli che non personalizzi. Un valore che non coincide con quello predefinito per un livello non personalizzato viene rifiutato.
- Ciascuna `AcrValue` è una stringa da 1 a 64 caratteri.
- Ciascuno di essi non `AcrValue` può contenere spazi, virgolette doppie o barre rovesciate.

Se modifichi i nomi dei livelli ACR, un utente potrebbe successivamente presentare un token di accesso il cui `acr` valore non corrisponde più a nessun livello configurato. Amazon Cognito non rispetta il valore obsoleto e lo considera un livello non riconosciuto, ma questo non è un errore. Amazon Cognito rispetta comunque i `amr` valori del token e richiede solo i fattori che non sono già elencati nella `amr` dichiarazione, senza accreditare alcun livello particolare.

Requisiti del piano di funzionalità

Le seguenti funzionalità richiedono il piano di funzionalità Essentials o Plus.

- Richiesta di autenticazione avanzata. Questo vale per il `TARGET_ACR_VALUES` parametro on `InitiateAuth` e il `acr_values` parametro on managed login.
- Configurazione di nomi di livello ACR personalizzati con il `AcrConfiguration` campo attivo o. `CreateUserPool` `UpdateUserPool`

Il comportamento quando si richiede l'autenticazione avanzata con un piano di funzionalità insufficiente dipende dal punto di ingresso.

- L'`InitiateAuth`API restituisce un. `FeatureUnavailableInTierException`
- L'endpoint di autorizzazione all'accesso gestito ignora silenziosamente `acr_values` e procede con la normale autenticazione, senza errori.

Per ulteriori informazioni sui piani di funzionalità, consulta. Piani di funzionalità del pool di utenti

Le tue responsabilità in materia di autenticazione avanzata

Amazon Cognito emette token con il livello di autenticazione raggiunto dall'utente. L'applicazione è responsabile di quanto segue.

- Controlla il token ID corrente dell'utente prima di reindirizzare. Valuta `auth_time` in base al tuo `max_age` `acr` livello e a quello del token ID rispetto al livello richiesto dall'operazione, con i token già presenti nell'applicazione. Reindirizza l'utente ad Amazon Cognito per l'installazione solo se il livello o la freschezza richiesti non sono già stati raggiunti.
- Imposta l'`TARGET_ACR_VALUES`o corretto e l'o `acr_values` corretto. `MAX_AGE` `max_age` Amazon Cognito agisce solo in base a ciò che fornisci. Non deduce il livello o la freschezza che intendi. Se richiedi il livello sbagliato o ometti i parametri, Amazon Cognito non migliora l'utente come previsto.
- Convalida il `acr` reclamo nella tua domanda. Dopo che Amazon Cognito ha restituito i token, l'applicazione deve verificare che l'`acr`attestazione corrisponda al livello richiesto dall'operazione prima di concedere l'accesso. Amazon Cognito emette il token con il livello raggiunto. Far sì che un'operazione richieda un determinato livello è responsabilità dell'applicazione.
- Traduci i livelli di Amazon Cognito in altri standard esterni al token. Amazon Cognito calcola le `amr` affermazioni `acr` e a partire dal flusso di autenticazione. Non puoi personalizzare, sovrascrivere, aggiungere o sopprimere queste attestazioni con un trigger Lambda precedente alla generazione del token o con qualsiasi altro trigger Lambda. Se devi tradurre i livelli di Amazon Cognito in un altro standard, come NIST o eIDAS, fallo a livello di applicazione.

Utilizzo dei dispositivi utente nel pool di utenti

Quando accedi agli utenti del pool di utenti locali con l'API dei pool di utenti di Amazon Cognito, puoi associare i registri delle attività degli utenti dalla protezione dalle minacce a ciascuno dei loro dispositivi e, facoltativamente, consentire agli utenti di ignorare l'autenticazione a più fattori (MFA) se

utilizzano un dispositivo affidabile. Amazon Cognito include una chiave del dispositivo nella risposta a qualsiasi accesso che non includa già le informazioni sul dispositivo. La chiave del dispositivo è nel formato *Region*_*UUID*. Con una chiave del dispositivo, una libreria SRP (Secure Remote Password) e un pool di utenti che consente l'autenticazione del dispositivo, puoi richiedere agli utenti dell'app di definire attendibile il dispositivo corrente e non richiedere più un codice di autenticazione a più fattori (MFA) all'accesso.

Argomenti

- Configurazione dei dispositivi memorizzati
- Recupero della chiave di un dispositivo
- Accesso con un dispositivo
- Visualizzazione, aggiornamento e annullamento della memorizzazione dei dispositivi

Configurazione dei dispositivi memorizzati

Con i pool di utenti di Amazon Cognito, puoi associare ogni dispositivo degli utenti a un identificatore univoco, ovvero a una chiave del dispositivo. Quando presenti la chiave del dispositivo ed esegui l'autenticazione del dispositivo al momento dell'accesso, puoi configurare l'applicazione con un flusso di autenticazione del dispositivo affidabile. In questo flusso, l'applicazione può offrire agli utenti la possibilità di accedere senza MFA fino a un momento successivo, in base ai requisiti di sicurezza dell'app o alle preferenze degli utenti. Al termine di tale periodo di tempo, l'applicazione deve modificare lo stato del dispositivo in Non memorizzato e l'utente deve accedere con MFA fino a quando non conferma di voler ricordare un dispositivo. Ad esempio, l'applicazione potrebbe richiedere agli utenti di considerare attendibile un dispositivo per 30, 60 o 90 giorni. È possibile memorizzare questa data in un attributo personalizzato e, in quella data, modificare lo stato memorizzato del dispositivo. È quindi necessario richiedere nuovamente all'utente di inviare un codice MFA e impostare il dispositivo in modo che venga nuovamente memorizzato dopo l'autenticazione riuscita.

1. I dispositivi memorizzati possono sovrascrivere l'autenticazione a più fattori (MFA) solo nei pool di utenti con l'autenticazione a più fattori (MFA) attiva.

Quando l'utente accede con un dispositivo memorizzato, è necessario eseguire un'autenticazione aggiuntiva del dispositivo durante il flusso di autenticazione. Per ulteriori informazioni, consulta Accesso con un dispositivo.

Configura il tuo pool di utenti per ricordare i dispositivi nel menu di accesso del tuo pool di utenti, sotto Device tracking. Quando imposti le funzionalità dei dispositivi memorizzati nella console Amazon Cognito, hai tre opzioni: Always, User Opt-In e No.

Non memorizzare

Il pool di utenti non richiede agli utenti di memorizzare i dispositivi quando effettuano l'accesso.

Memorizza sempre

Quando l'app conferma il dispositivo di un utente, il pool di utenti ricorda sempre il dispositivo e non visualizza richieste di autenticazione a più fattori (MFA) in caso di futuri accessi riusciti da parte del dispositivo.

Accettazione dall'utente

Quando l'app conferma il dispositivo di un utente, il pool di utenti non elimina automaticamente le richieste di autenticazione a più fattori (MFA). Devi richiedere all'utente di scegliere se memorizzare o meno il proprio dispositivo.

Quando scegli Memorizza sempre o Accettazione dall'utente, Amazon Cognito genera una chiave identificativa del dispositivo e un segreto ogni volta che un utente accede da un dispositivo non identificato. La chiave del dispositivo è l'identificatore iniziale che l'app invia al pool di utenti quando l'utente esegue l'autenticazione del dispositivo.

Con ogni dispositivo utente confermato (memorizzato automaticamente o accettato dall'utente), puoi utilizzare la chiave identificativa del dispositivo e il segreto per autenticare un dispositivo a ogni accesso utente.

Puoi anche configurare le impostazioni dei dispositivi ricordati per il tuo pool di utenti in una richiesta CreateUserPoolo UpdateUserPoolAPI. Per ulteriori informazioni, consulta la DeviceConfigurationproprietà.

L'API dei pool di utenti Amazon Cognito dispone di operazioni aggiuntive per i dispositivi memorizzati.

1. ListDevicese AdminListDevicesrestituisce un elenco delle chiavi del dispositivo e dei relativi metadati per un utente.
2. GetDevicee AdminGetDevicerestituisce la chiave del dispositivo e i metadati per un singolo dispositivo.
3. UpdateDeviceStatuse AdminUpdateDeviceStatusimposta il dispositivo di un utente come ricordato o non ricordato.

4. ForgetDevicee AdminForgetDevicerimuovi il dispositivo confermato di un utente dal suo profilo.

Le operazioni API con nomi che iniziano con `Admin` sono destinate all'uso nelle app lato server e devono essere autorizzate mediante credenziali IAM. Per ulteriori informazioni, consulta Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite.

Recupero della chiave di un dispositivo

Ogni volta che l'utente accede con l'API dei pool di utenti e non include una chiave del dispositivo nei parametri di autenticazione nel formato `DEVICE_KEY`, Amazon Cognito restituisce una nuova chiave del dispositivo nella risposta. Nell'app pubblica lato client, inserisci la chiave del dispositivo nell'archivio dell'app in modo da poterla includere nelle richieste future. Nell'app riservata lato server, imposta un cookie del browser o un altro token lato client con la chiave del dispositivo dell'utente.

Prima che l'utente possa accedere con il proprio dispositivo attendibile, l'app deve confermare la chiave del dispositivo e fornire informazioni aggiuntive. Genera una ConfirmDevicerichiesta ad Amazon Cognito che confermi il dispositivo dell'utente con la chiave del dispositivo, un nome descrittivo, un verificatore di password e un codice identificativo. Se hai configurato il pool di utenti per l'autenticazione del dispositivo tramite l'opzione Accettazione dall'utente, Amazon Cognito risponde alla richiesta `ConfirmDevice` chiedendo all'utente di scegliere se memorizzare il dispositivo corrente. Rispondi con la selezione dell'utente in una richiesta. UpdateDeviceStatus

Quando confermi il dispositivo dell'utente ma non lo imposti come dispositivo memorizzato, Amazon Cognito memorizza l'associazione ma procede con l'accesso senza dispositivo quando fornisci la chiave del dispositivo. I dispositivi possono generare log utili per la sicurezza e la risoluzione dei problemi degli utenti. Un dispositivo confermato ma non memorizzato non usa la funzionalità di accesso, ma sfrutta la funzionalità dei log di monitoraggio della sicurezza. Quando attivi la protezione dalle minacce per il client dell'app e codifichi l'impronta digitale del dispositivo nella tua richiesta, Amazon Cognito associa gli eventi utente al dispositivo confermato.

Recupero di una nuova chiave del dispositivo

1. Inizia la sessione di accesso dell'utente con una richiesta API. InitiateAuth
2. Rispondi a tutte le sfide di autenticazione RespondToAuthChallengefino a quando non ricevi i token web JSON (JWTs) che contrassegnano come completa la sessione di accesso dell'utente.
3. Nella tua app, registra i valori restituiti da Amazon Cognito in `NewDeviceMetadata` nella relativa risposta `RespondToAuthChallenge` o `InitiateAuth`: `DeviceGroupKey` e `DeviceKey`.

4. Genera un nuovo segreto SRP per il tuo utente: un salt e un verificatore di password. Questa funzione è disponibile in coloro SDKs che forniscono librerie SRP.
5. Richiedi all'utente il nome del dispositivo o generane uno in base alle caratteristiche del dispositivo dell'utente.
6. Fornisci il token di accesso, la chiave del dispositivo, il nome del dispositivo e il segreto SRP dell'utente in una richiesta ConfirmDeviceAPI. Se per il pool di utenti è impostata la memorizzazione dei dispositivi mediante l'opzione Memorizza sempre, la registrazione dell'utente è completa.
7. Se Amazon Cognito ha risposto `ConfirmDevice` con `"UserConfirmationNecessary": true`, chiedi all'utente di scegliere se desidera memorizzare il dispositivo. Se affermano di voler ricordare il dispositivo, genera una richiesta UpdateDeviceStatusAPI con il token di accesso, la chiave del dispositivo e la chiave del dispositivo dell'utente. `"DeviceRememberedStatus": "remembered"`
8. Se hai richiesto ad Amazon Cognito di memorizzare il dispositivo, la prossima volta che l'utente accede, anziché una richiesta di autenticazione a più fattori (MFA), viene visualizzata una richiesta di autenticazione `DEVICE_SRP_AUTH`.

Accesso con un dispositivo

Dopo aver configurato il dispositivo di un utente in modo che venga memorizzato, Amazon Cognito non richiede più che l'utente invii un codice di autenticazione a più fattori (MFA) quando accede con la stessa chiave del dispositivo. L'autenticazione dei dispositivi si limita a sostituire la richiesta di autenticazione a più fattori (MFA) con una richiesta di autenticazione del dispositivo. Non è possibile effettuare l'accesso degli utenti solo con l'autenticazione del dispositivo. L'utente deve prima completare l'autenticazione con la propria password o una richiesta di autenticazione personalizzata. Di seguito è riportato il processo di autenticazione per un utente su un dispositivo memorizzato.

Per eseguire l'autenticazione del dispositivo in un flusso che utilizza i trigger Lambda della sfida di autenticazione personalizzata, passa un `DEVICE_KEY` parametro nella richiesta InitiateAuthAPI. Dopo che l'utente ha completato con successo tutte le richieste di autorizzazione e la richiesta di autenticazione `CUSTOM_CHALLENGE` ha restituito `true` come valore `issueTokens`, Amazon Cognito restituisce un'ultima richiesta `DEVICE_SRP_AUTH`.

Accesso con un dispositivo

1. Recupera la chiave del dispositivo dell'utente dall'archivio del client.

2. Inizia la sessione di accesso dell'utente con una InitiateAuthrichiesta API. Come valore di `AuthFlow` scegli `USER_SRP_AUTH`, `REFRESH_TOKEN_AUTH`, `USER_PASSWORD_AUTH` o `CUSTOM_AUTH`. In `AuthParameters`, aggiungi la chiave del dispositivo dell'utente al parametro `DEVICE_KEY` e includi gli altri parametri obbligatori per il flusso di accesso selezionato.
   a. Puoi anche passare `DEVICE_KEY` nei parametri di una risposta `PASSWORD_VERIFIER` a una richiesta di autenticazione.
3. Completa le risposte alle richieste di autenticazione finché non ricevi una richiesta `DEVICE_SRP_AUTH` nella risposta.
4. In una richiesta RespondToAuthChallengeAPI, invia un `ChallengeName` of `DEVICE_SRP_AUTH` e i parametri per `USERNAMEDEVICE_KEY`, e`SRP_A`.
5. Amazon Cognito risponde con una richiesta `DEVICE_PASSWORD_VERIFIER`. La risposta a questa richiesta include i valori per `SECRET_BLOCK` e `SRP_B`.
6. Con la libreria SRP, genera e invia i parametri `PASSWORD_CLAIM_SIGNATURE`, `PASSWORD_CLAIM_SECRET_BLOCK`, `TIMESTAMP`. `USERNAME` e `DEVICE_KEY` in una richiesta `RespondToAuthChallenge` aggiuntiva.
7. Completa le sfide aggiuntive finché non ricevi quelle dell'utente JWTs.

Il seguente pseudocodice mostra come calcolare i valori per la risposta alla richiesta `DEVICE_PASSWORD_VERIFIER`. Per l'autenticazione SRP con un dispositivo, genera un nuovo segreto SRP per il tuo utente: una nuova password ad alta entropia`DeviceSecret`, un salt e il verificatore di password associato. Questi valori sono diversi dalla password, dal sale e dal verificatore utilizzati per l'autenticazione SRP dell'utente. Vengono utilizzati solo per l'autenticazione del dispositivo e vengono memorizzati solo sul dispositivo. Le funzioni per la generazione dei segreti SRP per i dispositivi degli utenti sono disponibili nelle librerie SRP disponibili in diverse librerie. SDKs

```
PASSWORD_CLAIM_SECRET_BLOCK = SECRET_BLOCK
TIMESTAMP = "Tue May 7 00:09:40 UTC 2025"
k = SHA256(N || g) as a non-negative integer in big-endian
u = SHA256(SRP_A || SRP_B) as a non-negative integer in big-endian
x = SHA256(salt || SHA256(DeviceGroupKey || DeviceKey || ":" || DeviceSecret)) as a
 non-negative integer in big-endian
S_USER = (SRP_B - k * g^x)^(a + u * x) % N
K_USER = HKDF_HMAC_SHA256(salt=u, ikm=S_USER, info="Caldera Derived Key", length=16
 bytes)
PASSWORD_CLAIM_SIGNATURE = Base64(HMAC_SHA256(key=K_USER, message=(DeviceGroupKey ||
 DeviceKey || PASSWORD_CLAIM_SECRET_BLOCK || TIMESTAMP)))
```

Visualizzazione, aggiornamento e annullamento della memorizzazione dei dispositivi

Puoi implementare le seguenti funzionalità nella tua app con l'API Amazon Cognito.

1. Visualizzazione delle informazioni sul dispositivo corrente di un utente.
2. Visualizzazione dell'elenco di tutti i dispositivi dell'utente.
3. Annullamento della memorizzazione di dispositivo.
4. Aggiornamento dello stato di memorizzazione di un dispositivo.

I token di accesso che autorizzano le richieste API nelle seguenti descrizioni devono includere l'ambito `aws.cognito.signin.user.admin`. Amazon Cognito aggiunge una richiesta per questo ambito a tutti i token di accesso generati con l'API dei pool di utenti di Amazon Cognito. IdPs Le terze parti devono gestire separatamente i dispositivi e l'MFA per i propri utenti che si autenticano su Amazon Cognito. Nell'accesso gestito, puoi richiedere l'`aws.cognito.signin.user.admin`ambito, ma l'accesso gestito aggiunge automaticamente le informazioni sul dispositivo ai log utente di sicurezza avanzata e non offre la possibilità di ricordare i dispositivi.

Visualizzazione delle informazioni su un dispositivo

È possibile eseguire una query sulle informazioni sul dispositivo di un utente per determinare se è ancora in uso. Ad esempio, potresti voler disattivare i dispositivi memorizzati se non hanno effettuato l'accesso per 90 giorni consecutivi.

- Per visualizzare le informazioni sul dispositivo dell'utente in un'app client pubblica, invia la chiave di accesso e la chiave del dispositivo dell'utente in una GetDevicerichiesta API.
- Per visualizzare le informazioni sul dispositivo dell'utente in un'app client riservata, firma una richiesta AdminGetDeviceAPI con AWS le credenziali e invia il nome utente, la chiave del dispositivo e il pool di utenti dell'utente.

Visualizzazione dell'elenco di tutti i dispositivi dell'utente.

Puoi visualizzare l'elenco di tutti i dispositivi dell'utente e delle relative proprietà. Ad esempio, potresti voler verificare se il dispositivo corrente corrisponde a un dispositivo memorizzato.

- In un'app per client pubblici, invia il token di accesso dell'utente in una richiesta API. ListDevices

- In un'app riservata ai clienti, firma una richiesta AdminListDevicesAPI con AWS credenziali e invia il nome utente e il pool di utenti dell'utente.

Annullamento della memorizzazione di un dispositivo

Puoi eliminare la chiave del dispositivo di un utente. È consigliabile eseguire questa operazione quando un utente non utilizza più un dispositivo o quando vengono rilevate attività insolite e si desidera richiedere a un utente di rieseguire l'autenticazione a più fattori (MFA). Per registrare nuovamente il dispositivo in un secondo momento, è necessario generare e archiviare una nuova chiave del dispositivo.

- In un'app per client pubblici, invia la chiave del dispositivo e il token di accesso dell'utente nella richiesta API. ForgetDevice
- In un'app riservata ai client, invia la chiave del dispositivo e il token di accesso dell'utente nella richiesta API. AdminForgetDevice

Sicurezza avanzata con protezione dalle minacce

Dopo aver creato il pool di utenti, puoi accedere alla protezione dalle minacce nel menu di navigazione della console Amazon Cognito. Puoi attivare le funzionalità di protezione dalle minacce e personalizzare le azioni intraprese in risposta a diversi rischi. In alternativa, si può utilizzare la modalità di controllo per raccogliere i parametri sui rischi rilevati, senza applicare mitigazioni della sicurezza. In modalità di controllo, la protezione dalle minacce pubblica i parametri su Amazon. CloudWatch Puoi visualizzare le metriche dopo che Amazon Cognito ha generato il primo evento. Consulta Visualizzazione delle metriche di protezione dalle minacce.

La protezione dalle minacce, precedentemente denominata funzionalità di sicurezza avanzate, è un insieme di strumenti di monitoraggio delle attività indesiderate nel pool di utenti e strumenti di configurazione per arrestare automaticamente le attività potenzialmente dannose. La protezione dalle minacce offre diverse opzioni di configurazione per le operazioni di autenticazione standard e personalizzate. Ad esempio, potresti voler inviare una notifica a un utente con un accesso di autenticazione personalizzato sospetto, in cui hai impostato fattori di sicurezza aggiuntivi, ma bloccare un utente allo stesso livello di rischio con l'autenticazione di base tra nome utente e password.

La protezione dalle minacce è disponibile nel piano di funzionalità Plus. Per ulteriori informazioni, consulta Piani di funzionalità del pool di utenti.

Le seguenti opzioni del pool di utenti sono i componenti della protezione dalle minacce.

Credenziali compromesse

Gli utenti riutilizzano le password per più account utente. La funzionalità delle credenziali compromesse di Amazon Cognito raccoglie i dati dalle fughe pubbliche di nomi utente e password e confronta le credenziali degli utenti con gli elenchi di credenziali trapelate. Il rilevamento delle credenziali compromesse verifica anche le password più comuni. È possibile verificare la presenza di credenziali compromesse nei flussi di autenticazione standard relativi a nome utente e password nei pool di utenti. Amazon Cognito non rileva credenziali compromesse nella password remota sicura (SRP) o nell'autenticazione personalizzata.

Puoi scegliere le azioni dell'utente che richiedono la verifica della presenza di credenziali compromesse e l'azione che deve effettuare Amazon Cognito in risposta. Per gli eventi di accesso, registrazione e modifica della password, Amazon Cognito può scegliere tra le opzioni Blocca l'accesso o Consenti l'accesso. In entrambi i casi, Amazon Cognito genera un registro delle attività degli utenti in cui è possibile trovare ulteriori informazioni sull'evento.

Ulteriori informazioni

Utilizzo del rilevamento di credenziali compromesse

Autenticazione adattiva

Amazon Cognito può esaminare le informazioni sulla posizione e il dispositivo delle richieste di accesso degli utenti e applicare una risposta automatica per proteggere gli account utente del pool di utenti da attività sospette. È possibile monitorare l'attività degli utenti e automatizzare le risposte ai livelli di rischio rilevati in termini di nome utente-password, SRP e autenticazione personalizzata.

Quando attivi la protezione dalle minacce, Amazon Cognito assegna un punteggio di rischio all'attività degli utenti. Puoi assegnare una risposta automatica all'attività sospetta: puoi scegliere tra le opzioni Richiedi MFA, Blocca l'accesso o semplicemente registrare i dettagli dell'attività e il punteggio di rischio. Puoi anche inviare automaticamente messaggi e-mail che informano l'utente dell'attività sospetta in modo che possa reimpostare la password o eseguire altre azioni autogestite.

Ulteriori informazioni

Utilizzo dell'autenticazione adattiva

Elenco di indirizzi IP consentiti e denylist

Con la protezione dalle minacce di Amazon Cognito in modalità Full function, puoi creare indirizzi IP Always block e Always allow eccezioni. A una sessione il cui indirizzo IP è nella lista delle eccezioni Blocca sempre non viene assegnato un livello di rischio tramite autenticazione adattiva e non può accedere al bacino d'utenza.

Cose da sapere sulle liste IP-address consentite e sulle liste bloccate

- È necessario esprimere Always block e Always allow in formato CIDR`192.0.2.0/24`, ad esempio una maschera a 24 bit o un singolo indirizzo `192.0.2.252/32` IP.
- I dispositivi con indirizzi IP in un intervallo IP Always block non possono registrarsi o accedere con SDK-based o gestire applicazioni di accesso, ma possono accedere con applicazioni di accesso di terze parti. IdPs
- Gli elenchi Consenti sempre e blocca sempre non influiscono sull'aggiornamento dei token.
- Amazon Cognito non applica regole MFA di autenticazione adattiva ai dispositivi appartenenti a un intervallo IP Always allow, ma applica regole sulle credenziali compromesse.

Esportazione dei log

La protezione dalle minacce registra i dettagli granulari delle richieste di autenticazione degli utenti nel pool di utenti. Questi registri contengono valutazioni delle minacce, informazioni sugli utenti e metadati delle sessioni come posizione e dispositivo. È possibile creare archivi esterni di questi registri per la conservazione e l'analisi. I pool di utenti di Amazon Cognito esportano i log di protezione dalle minacce su Amazon S3, Logs CloudWatch e Amazon Data Firehose. Per ulteriori informazioni, consulta Visualizzazione ed esportazione della cronologia degli eventi degli utenti.

Ulteriori informazioni

Esportazione dei log delle attività degli utenti per la protezione dalle minacce

Argomenti

- Considerazioni e limitazioni per la protezione dalle minacce
- Attivazione della protezione dalle minacce nei pool di utenti
- Concetti di applicazione della protezione dalle minacce
- Protezione dalle minacce per l'autenticazione standard e l'autenticazione personalizzata
- Prerequisiti per la protezione dalle minacce

- Impostazione della protezione dalle minacce
- Utilizzo del rilevamento di credenziali compromesse
- Utilizzo dell'autenticazione adattiva
- Raccolta di dati per la protezione dalle minacce nelle applicazioni

Considerazioni e limitazioni per la protezione dalle minacce

Le opzioni di protezione dalle minacce differiscono tra i flussi di autenticazione

Amazon Cognito supporta sia l'autenticazione adattiva che il rilevamento delle credenziali compromesse con i flussi di autenticazione e. `USER_PASSWORD_AUTH` `ADMIN_USER_PASSWORD_AUTH` Puoi abilitare solo l'autenticazione adattiva per. `USER_SRP_AUTH` Non puoi utilizzare la protezione dalle minacce con l'accesso federato.

Always-block Gli IP contribuiscono alla richiesta di quote

Le richieste bloccate provenienti da indirizzi IP nella lista delle eccezioni Blocca sempre nel tuo baclno d'utenza, contribuiscono alle quote tariffarie per le richieste dei tuoi bacini d'utenza.

La protezione dalle minacce non applica limiti di velocità

Parte del traffico malevolo presenta la caratteristica di un elevato volume di richieste, come gli attacchi DDoS (Distributed Denial of Service). Le valutazioni di rischio che Amazon Cognito applica al traffico in entrata si basano sulle singole richieste e non tengono conto del volume delle richieste. Le richieste individuali relative a un evento ad alto volume potrebbero ricevere un punteggio di rischio e una risposta automatica per motivi a livello di applicazione che non sono correlati al loro ruolo in un attacco volumetrico. Per implementare difese contro gli attacchi volumetrici nei tuoi pool di utenti, aggiungi gli ACL web. AWS WAF Per ulteriori informazioni, consulta Associa un AWS WAF ACL web con un pool di utenti.

La protezione dalle minacce non influisce sulle richieste M2M

Le credenziali dei client sono destinate all'autorizzazione da macchina a macchina (M2M) senza connessione agli account utente. La protezione dalle minacce monitora solo gli account utente e le password del pool di utenti. Per implementare funzionalità di sicurezza nella tua attività M2M, prendi in considerazione le funzionalità di monitoraggio della frequenza delle AWS WAF richieste e dei contenuti. Per ulteriori informazioni, consulta Associa un AWS WAF ACL web con un pool di utenti.

Attivazione della protezione dalle minacce nei pool di utenti

Amazon Cognito user pools console

Per attivare la protezione dalle minacce per un pool di utenti

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Se non l'hai già fatto, attiva il piano di funzionalità Plus dal menu Impostazioni.
5. Scegli il menu Protezione dalle minacce e seleziona Attiva.
6. Scegli Save changes (Salva modifiche).

API

Imposta il tuo piano di funzionalità su Plus in una richiesta CreateUserPoolo UpdateUserPoolAPI. Il seguente esempio parziale del corpo della richiesta imposta la protezione dalle minacce sulla modalità a funzionamento completo. Per una richiesta di esempio completa, vedi Esempi.

```
"UserPoolAddOns": {
      "AdvancedSecurityMode": "ENFORCED"
   }
```

La protezione dalle minacce è il termine collettivo per le funzionalità che monitorano le operazioni degli utenti alla ricerca di segnali di acquisizione dell'account e rispondono automaticamente alla sicurezza degli account utente interessati. È possibile applicare le impostazioni di protezione dalle minacce agli utenti quando accedono con flussi di autenticazione standard e personalizzati.

La protezione dalle minacce genera registri che descrivono in dettaglio l'accesso, la disconnessione e altre attività degli utenti. È possibile esportare questi registri su un sistema di terze parti. Per ulteriori informazioni, consulta Visualizzazione ed esportazione della cronologia degli eventi degli utenti.

Concetti di applicazione della protezione dalle minacce

La protezione dalle minacce inizia in una modalità di solo controllo in cui il pool di utenti monitora l'attività degli utenti, assegna i livelli di rischio e genera registri. Come best practice, esegui in modalità di solo controllo per due o più settimane prima di abilitare la modalità con funzionalità

complete. Full-function la modalità include una serie di reazioni automatiche alle attività rischiose rilevate e alle password compromesse. Con la modalità di solo controllo, puoi monitorare le valutazioni delle minacce eseguite da Amazon Cognito. Puoi anche fornire un feedback per addestrare la funzionalità a prevenire falsi positivi e negativi.

È possibile configurare l'applicazione della protezione dalle minacce a livello di pool di utenti per coprire tutti i client di app del pool di utenti e a livello di singoli client di app. Le configurazioni di protezione dalle minacce del client dell'app hanno la precedenza sulla configurazione del pool di utenti. Per configurare la protezione dalle minacce per un client di app, accedi alle impostazioni del client dell'app dal menu App client del tuo pool di utenti nella console Amazon Cognito. Qui puoi utilizzare le impostazioni a livello di client e configurare l'applicazione esclusivamente per il client dell'app.

Inoltre, è possibile configurare la protezione dalle minacce separatamente per i tipi di autenticazione standard e personalizzati.

Protezione dalle minacce per l'autenticazione standard e l'autenticazione personalizzata

I modi in cui è possibile configurare la protezione dalle minacce dipendono dal tipo di autenticazione che si esegue nel pool di utenti e nei client delle app. Ciascuno dei seguenti tipi di autenticazione può avere una propria modalità di applicazione e risposte automatiche.

Autenticazione standard

L'autenticazione standard prevede l'accesso, la disconnessione e la gestione delle password degli utenti con flussi nome utente-password e accesso gestito. La protezione dalle minacce di Amazon Cognito monitora le operazioni alla ricerca di indicatori di rischio quando accedono con accesso gestito o utilizzano i seguenti parametri API: `AuthFlow`

InitiateAuth

`USER_PASSWORD_AUTH`,. `USER_SRP_AUTH` La funzionalità delle credenziali compromesse non ha accesso alle password durante `USER_SRP_AUTH` l'accesso e non monitora né agisce sugli eventi con questo flusso.

AdminInitiateAuth

`ADMIN_USER_PASSWORD_AUTH`,. `USER_SRP_AUTH` La funzionalità delle credenziali compromesse non ha accesso alle password durante `USER_SRP_AUTH` l'accesso e non monitora né agisce sugli eventi con questo flusso.

È possibile impostare la modalità Enforcement per l'autenticazione standard su Audit only o Full function. Per disabilitare il monitoraggio delle minacce per l'autenticazione standard, imposta la protezione dalle minacce su Nessuna applicazione.

Autenticazione personalizzata

L'autenticazione personalizzata è l'accesso dell'utente con trigger Lambda di sfida personalizzati. Non è possibile eseguire l'autenticazione personalizzata nell'accesso gestito. La protezione dalle minacce di Amazon Cognito monitora le operazioni alla ricerca di indicatori di rischio quando accedono con il `AuthFlow` parametro `CUSTOM_AUTH` API di e. `InitiateAuth` `AdminInitiateAuth`

Puoi impostare la modalità di imposizione per l'autenticazione personalizzata su Solo controllo, Funzione completa o Nessuna imposizione. L'opzione No enforcement disabilita il monitoraggio delle minacce per l'autenticazione personalizzata senza influire sulle altre funzionalità di protezione dalle minacce.

Prerequisiti per la protezione dalle minacce

Prima di iniziare, avrai bisogno di:

- Un bacino d'utenza con un client dell'app. Per ulteriori informazioni, consulta Crea una nuova applicazione nella console Amazon Cognito.
- Imposta l'autenticazione a più fattori (MFA) su Facoltativa nella console Amazon Cognito per utilizzare la caratteristica di autenticazione adattiva basata sul rischio. Per ulteriori informazioni, consulta Aggiunta dell'autenticazione MFA a un bacino d'utenza.
- Se utilizzi l'indirizzo e-mail per le notifiche, passa alla console Amazon SES per configurare e verificare un indirizzo e-mail o un dominio da utilizzare con le notifiche e-mail. Per ulteriori informazioni su Amazon SES, consulta Verifica delle identità in Amazon SES.

Impostazione della protezione dalle minacce

Segui queste istruzioni per configurare la protezione dalle minacce del pool di utenti.

> **Note**
>
> Per configurare una diversa configurazione di protezione dalle minacce per un client di app nella console dei pool di utenti di Amazon Cognito, seleziona il client dell'app dal menu App client e scegli Usa impostazioni a livello di client.

Console di gestione AWS

Per configurare la protezione dalle minacce per un pool di utenti

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Protezione dalle minacce e seleziona Attiva.
5. Scegli il metodo di protezione dalle minacce che desideri configurare: autenticazione standard e personalizzata. È possibile impostare diverse modalità di applicazione per l'autenticazione personalizzata e standard, ma entrambe condividono la configurazione delle risposte automatiche in modalità Full function.
6. Seleziona Edit (Modifica).
7. Scegliete una modalità di applicazione. Per iniziare a rispondere immediatamente ai rischi rilevati, seleziona Funzione completa e configura le risposte automatiche per le credenziali compromesse e l'autenticazione adattiva. Per raccogliere informazioni nei log a livello di utente e in accesso, seleziona Solo controllo. CloudWatch

   Si consiglia di mantenere la protezione dalle minacce in modalità di controllo per due settimane prima di abilitare le azioni. Durante questo periodo, Amazon Cognito può conoscere i modelli di utilizzo degli utenti della tua app e puoi fornire feedback sugli eventi per modificare le risposte.
8. Se hai selezionato Audit only (Solo audit), scegli Save changes (Salva modifiche). Se hai selezionato Full function (Funzione completa):

   a. Seleziona se intraprendere un'azione personalizzata o utilizzare le impostazioni predefinite di Cognito per rispondere a credenziali sospette compromesse. Le Impostazioni predefinite di Cognito sono:

i. Rileva le credenziali compromesse attivando e modificando la password. Sign-inSign-up

ii. Rispondi alla compromissione delle credenziali con l'operazione Blocca accesso.

b. Se hai selezionato Azioni personalizzate per credenziali compromesse, scegli le azioni del pool di utenti che Amazon Cognito utilizza per il rilevamento degli eventi e le risposte relative alle credenziali compromesse che desideri che Amazon Cognito esegua. Puoi eseguire le operazioni Block sign-in (Blocca accesso) o Allow sign-in (Consenti accesso) nel caso in cui vi sia il sospetto che le credenziali siano state compromesse.

c. Scegli come rispondere ai tentativi di accesso dannosi alla voce Adaptive authentication (Autenticazione adattiva). Seleziona se intraprendere un'azione personalizzata o utilizzare le impostazioni predefinite di Cognito per rispondere a sospette attività dannose. Selezionando le Impostazioni di default di Cognito, Amazon Cognito bloccherà l'accesso a tutti i livelli di rischio e non notificherà l'utente.

d. Se hai selezionato Azioni personalizzate per l'autenticazione adattiva, scegli le azioni di risposta automatica al rischio che Amazon Cognito intraprende in risposta ai rischi rilevati in base al livello di gravità. Quando assegni una risposta a un livello di rischio, non puoi assegnare una risposta meno restrittiva a un livello di rischio più elevato. È possibile assegnare le seguenti risposte ai livelli di rischio:

i. Allow sign-in (Consenti l'accesso): non intraprendere alcuna operazione preventiva.

ii. MFA opzionale: se l'utente ha configurato l'MFA, Amazon Cognito richiede sempre all'utente di fornire un SMS aggiuntivo o un fattore di password monouso (TOTP) basato sul tempo al momento dell'accesso. Se l’utente non ha configurato MFA, può continuare ad accedere normalmente.

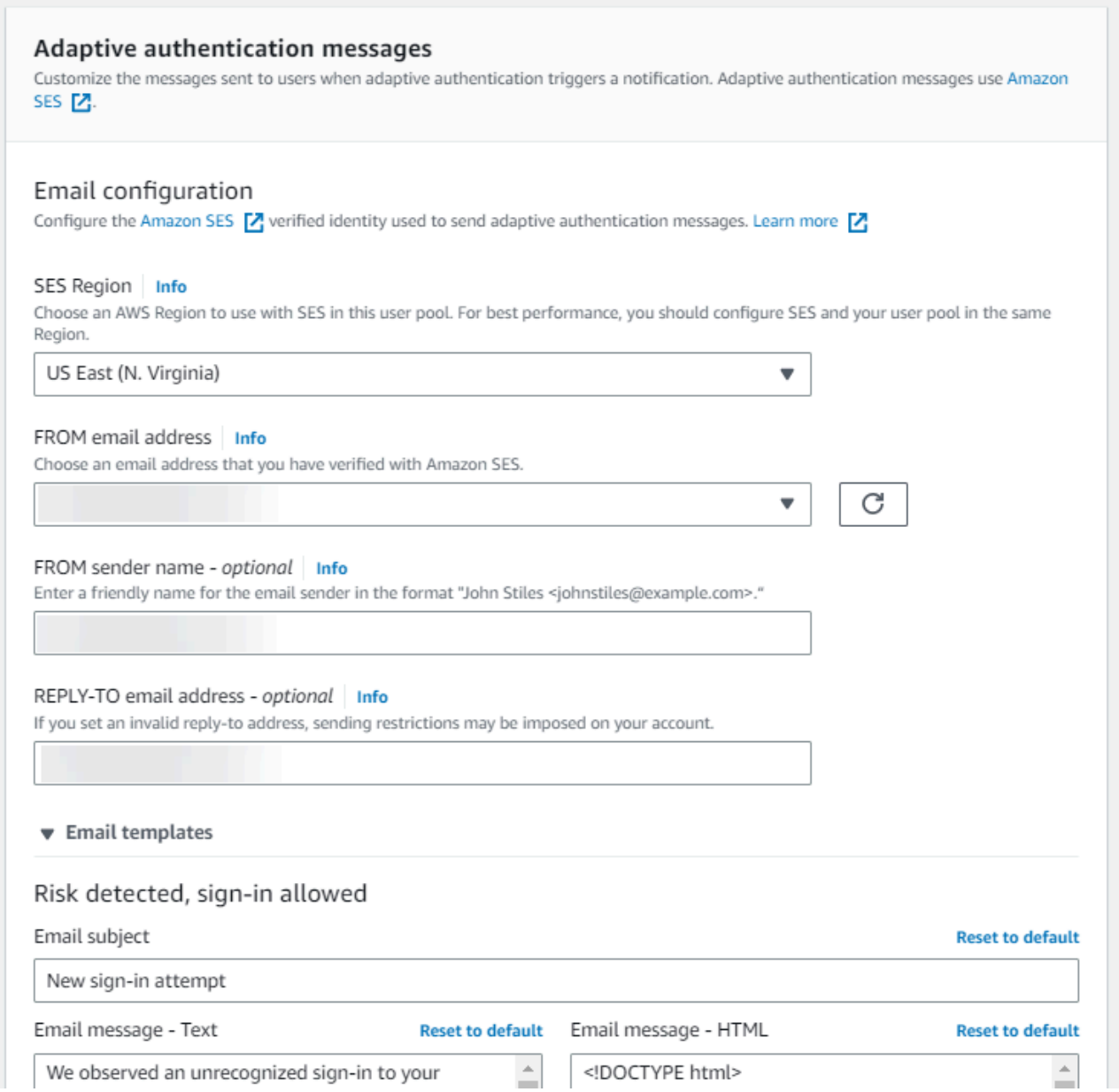
iii. Richiedi MFA: se l'utente ha configurato l'MFA, Amazon Cognito richiede sempre all'utente di fornire un SMS o un fattore TOTP aggiuntivo al momento dell'accesso. Se l'utente non ha configurato l'MFA, Amazon Cognito richiede di configurare l'MFA. Prima di richiedere automaticamente l'MFA per i tuoi utenti, configura un sistema nell'app che consenta di acquisire i numeri di telefono per l'MFA tramite SMS o di registrare app di autenticazione per l'MFA tramite TOTP.

iv. Block sign-in (Blocca l'accesso): impedisci all'utente di accedere.

v. Notify user (Notifica l'utente): invia un'e-mail all'utente con informazioni sul rischio rilevato da Amazon Cognito e sulla risposta che hai messo in atto. È possibile personalizzare i modelli di messaggio e-mail per i messaggi inviati.

9. Se hai scelto Notify user (Notifica l'utente) nel passaggio precedente, puoi personalizzare le impostazioni di recapito e i modelli di messaggio e-mail per l'autenticazione adattiva.

   a. In Configurazione e-mail, scegli la regione SES, l'indirizzo e-mail FROM, il nome del mittente FROM e l'indirizzo REPLY-TO e-mail che desideri utilizzare con l'autenticazione adattiva. Per ulteriori informazioni sull'integrazione di messaggi e-mail per il bacino d'utenza con Amazon Simple Email Service, consulta Impostazioni e-mail per i bacini d'utenza di Amazon Cognito.



   b. Espandi Email templates (Modelli e-mail) per personalizzare le notifiche di autenticazione adattiva per i messaggi e-mail in versione HTML e in testo normale. Per ulteriori informazioni sui modelli di messaggio e-mail, consulta Modelli dei messaggi.

10. Espandi le eccezioni degli indirizzi IP per creare uno Always-allowo un Always-blockelenco di intervalli di indirizzi IPv4 o IPv6 sempre consentiti o bloccati, indipendentemente dalla valutazione del rischio di protezione dalle minacce. Specificate gli intervalli di indirizzi IP in notazione CIDR (ad esempio 192.168.100). 0/24).
11. Scegli Save changes (Salva modifiche).

API (user pool)

Per impostare la configurazione della protezione dalle minacce per un pool di utenti, invia una richiesta SetRiskConfigurationAPI che includa un `UserPoolId` parametro, ma non un `ClientId` parametro. Di seguito è riportato un esempio di corpo della richiesta per un pool di utenti. Questa configurazione del rischio esegue una serie crescente di azioni in base alla gravità del rischio e avvisa gli utenti a tutti i livelli di rischio. Applica un blocco di credenziali compromesse alle operazioni di registrazione.

Per applicare questa configurazione, è necessario impostarla su `ENFORCED` in una richiesta API `AdvancedSecurityMode` o separata. CreateUserPoolUpdateUserPool Per ulteriori informazioni sui modelli di segnaposto come `{username}` in questo esempio, consulta. Configurazione MFA, autenticazione, verifica e messaggi di invito

```
{
   "AccountTakeoverRiskConfiguration": {
      "Actions": {
         "HighAction": {
            "EventAction": "MFA_REQUIRED",
            "Notify": true
         },
         "LowAction": {
            "EventAction": "NO_ACTION",
            "Notify": true
         },
         "MediumAction": {
            "EventAction": "MFA_IF_CONFIGURED",
            "Notify": true
         }
      },
      "NotifyConfiguration": {
         "BlockEmail": {
            "Subject": "You have been blocked for suspicious activity",
            "TextBody": "We blocked {username} at {login-time} from {ip-address}."
```

```
         },
         "From": "admin@example.com",
         "MfaEmail": {
            "Subject": "Suspicious activity detected, MFA required",
            "TextBody": "Unexpected sign-in from {username} on device {device-name}.
 You must use MFA."
         },
         "NoActionEmail": {
            "Subject": "Suspicious activity detected, secure your user account",
            "TextBody": "We noticed suspicious sign-in activity by {username} from
 {city}, {country} at {login-time}. If this was not you, reset your password."
         },
         "ReplyTo": "admin@example.com",
         "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
      }
   },
   "CompromisedCredentialsRiskConfiguration": {
      "Actions": {
         "EventAction": "BLOCK"
      },
      "EventFilter": [ "SIGN_UP" ]
   },
   "RiskExceptionConfiguration": {
      "BlockedIPRangeList": [ "192.0.2.0/24","198.51.100.0/24" ],
      "SkippedIPRangeList": [ "203.0.113.0/24" ]
   },
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

API (app client)

Per impostare la configurazione di protezione dalle minacce per un client di app, invia una richiesta SetRiskConfigurationAPI che includa un `UserPoolId` parametro e un `ClientId` parametro. Di seguito è riportato un esempio di corpo di richiesta per un client di app. Questa configurazione a rischio è più severa della configurazione del pool di utenti e blocca le voci ad alto rischio. Inoltre, applica blocchi di credenziali compromesse alle operazioni di registrazione, accesso e reimpostazione della password.

Per applicare questa configurazione, devi impostarla su in una richiesta API o separata. `AdvancedSecurityMode` `ENFORCED` CreateUserPoolUpdateUserPool Per ulteriori informazioni

sui modelli di segnaposto come `{username}` in questo esempio, consulta. [Configurazione MFA, autenticazione, verifica e messaggi di invito]

```
{
   "AccountTakeoverRiskConfiguration": {
      "Actions": {
         "HighAction": {
            "EventAction": "BLOCK",
            "Notify": true
         },
         "LowAction": {
            "EventAction": "NO_ACTION",
            "Notify": true
         },
         "MediumAction": {
            "EventAction": "MFA_REQUIRED",
            "Notify": true
         }
      },
      "NotifyConfiguration": {
         "BlockEmail": {
            "Subject": "You have been blocked for suspicious activity",
            "TextBody": "We blocked {username} at {login-time} from {ip-address}."
         },
         "From": "admin@example.com",
         "MfaEmail": {
            "Subject": "Suspicious activity detected, MFA required",
            "TextBody": "Unexpected sign-in from {username} on device {device-name}.
 You must use MFA."
         },
         "NoActionEmail": {
            "Subject": "Suspicious activity detected, secure your user account",
            "TextBody": "We noticed suspicious sign-in activity by {username} from
 {city}, {country} at {login-time}. If this was not you, reset your password."
         },
         "ReplyTo": "admin@example.com",
         "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
      }
   },
   "ClientId": "1example23456789",
   "CompromisedCredentialsRiskConfiguration": {
      "Actions": {
```

```
            "EventAction": "BLOCK"
         },
         "EventFilter": [ "SIGN_UP", "SIGN_IN", "PASSWORD_CHANGE" ]
      },
      "RiskExceptionConfiguration": {
         "BlockedIPRangeList": [ "192.0.2.1/32","192.0.2.2/32" ],
         "SkippedIPRangeList": [ "192.0.2.3/32","192.0.2.4/32" ]
      },
      "UserPoolId": "us-west-2_EXAMPLE"
}
```

Utilizzo del rilevamento di credenziali compromesse

Amazon Cognito è in grado di rilevare se il nome utente e la password di un utente sono stati compromessi. Questo può accadere quando gli utenti utilizzano le stesse credenziali su più di un sito oppure quando utilizzano password non sicure. Amazon Cognito verifica gli utenti locali che accedono con nome utente e password, con accesso gestito e con l'API Amazon Cognito.

Dal menu Protezione dalle minacce della console Amazon Cognito, puoi configurare le credenziali compromesse. Configura Event detection (Rilevamento di eventi) per scegliere gli eventi utente da monitorare per individuare le credenziali compromesse. Configura Compromised credentials responses (Risposte delle credenziali compromesse) per scegliere se consentire o bloccare l'utente se vengono rilevate credenziali compromesse. Amazon Cognito può controllare la presenza di credenziali compromesse durante gli accessi, le registrazioni e le modifiche delle password.

Quando scegli Consenti l'accesso, puoi esaminare Amazon CloudWatch Logs per monitorare le valutazioni effettuate da Amazon Cognito sugli eventi degli utenti. Per ulteriori informazioni, consulta Visualizzazione delle metriche di protezione dalle minacce. Quando si sceglie Block sign-in (Blocca l'accesso), Amazon Cognito impedisce l'accesso da parte degli utenti che utilizzano credenziali compromesse. Quando Amazon Cognito blocca l'accesso a un utente, imposta il parametro `UserStatus` dell'utente su `RESET_REQUIRED`. Un utente con uno stato `RESET_REQUIRED` deve modificare la password prima di poter accedere di nuovo.

Le credenziali compromesse possono controllare le password per le seguenti attività degli utenti.

Sign-up

Il tuo pool di utenti controlla le password che gli utenti trasmettono SignUpdurante l'operazione e dalla pagina di registrazione dell'accesso gestito per individuare eventuali indicatori di compromissione.

Sign-in

Il tuo pool di utenti controlla le password che gli utenti inviano durante l'accesso basato su password per individuare eventuali indicatori di compromissione. Amazon Cognito può esaminare il `ADMIN_USER_PASSWORD_AUTH` flusso in entrata AdminInitiateAuth, il `USER_PASSWORD_AUTH` flusso in InitiateAuthentrata e l'`PASSWORD`opzione del `USER_AUTH` flusso in entrambi.

Al momento, Amazon Cognito non verifica la presenza di credenziali compromesse per operazioni di accesso con flusso Secure Remote Password (SRP). SRP invia una prova di password con hash durante l'accesso. Amazon Cognito non dispone dell'accesso alle password internamente, quindi può solo valutare una password passata dal client in testo normale.

Reimpostazione della password

Il tuo pool di utenti verifica la presenza di indicatori di compromissione nelle operazioni che impostano nuove password utente con l'operazione ConfirmForgotPasswordself-service di reimpostazione delle password. Il codice necessario per questa operazione viene generato da ForgotPassworde. AdminResetUserPassword

Le credenziali compromesse non controllano le password temporanee o permanenti impostate dall'amministratore con. AdminSetUserPassword Tuttavia, con le password temporanee, il gruppo di utenti verifica le password in base alle risposte alla sfida in and. `NEW_PASSWORD_REQUIRED` RespondToAuthChallengeAdminRespondToAuthChallenge

Per aggiungere misure di protezione contro le credenziali compromesse al bacino d'utenza, consulta Sicurezza avanzata con protezione dalle minacce.

Utilizzo dell'autenticazione adattiva

Con l'autenticazione adattiva, puoi configurare il bacino d'utenza in modo da bloccare i tentativi di accesso sospetti o aggiungere l'autenticazione tramite secondo fattore in risposta all'aumento del livello di rischio. Per ogni tentativo di accesso, Amazon Cognito genera un punteggio di rischio che indica la probabilità che la richiesta di accesso provenga da un'origine compromessa. Questo punteggio di rischio si basa sui fattori relativi al dispositivo e all'utente forniti dall'applicazione e su altri fattori che Amazon Cognito ricava dalla richiesta. Alcuni fattori che contribuiscono alla valutazione del rischio da parte di Amazon Cognito sono l'indirizzo IP, lo user agent e la distanza geografica da altri tentativi di accesso. L'autenticazione adattiva può attivare o richiedere l'autenticazione a più fattori (MFA) per un utente del pool di utenti quando Amazon Cognito rileva un rischio nella sessione di un utente che non ha ancora scelto un metodo MFA. Quando attivi l'autenticazione MFA,

l'utente riceve sempre la richiesta di fornire o impostare un secondo fattore durante l'autenticazione, indipendentemente da come è stata configurata l'autenticazione adattiva. Dal punto di vista dell'utente, l'app offre le indicazioni per configurare l'autenticazione MFA e, facoltativamente, Amazon Cognito gli impedisce di accedere nuovamente fino a quando non configura un fattore aggiuntivo.

Amazon Cognito pubblica parametri relativi ai tentativi di accesso, ai relativi livelli di rischio e alle sfide fallite per Amazon. CloudWatch Per ulteriori informazioni, consulta Visualizzazione delle metriche di protezione dalle minacce.

Per aggiungere l'autenticazione adattiva al bacino d'utenza, consulta Sicurezza avanzata con protezione dalle minacce.

Argomenti

- Panoramica dell'autenticazione adattiva
- Aggiunta di dati di dispositivo e sessione utente alle richieste API
- Visualizzazione ed esportazione della cronologia degli eventi degli utenti
- Fornitura del feedback sugli eventi
- Invio di messaggi di notifica

Panoramica dell'autenticazione adattiva

Dal menu Protezione dalle minacce nella console Amazon Cognito, puoi scegliere le impostazioni per l'autenticazione adattiva, incluse le azioni da intraprendere a diversi livelli di rischio e la personalizzazione dei messaggi di notifica agli utenti. Puoi assegnare una configurazione globale di protezione dalle minacce a tutti i client dell'app, ma applicare una configurazione a livello di client ai singoli client di app.

L'autenticazione adattiva di Amazon Cognito assegna uno dei seguenti livelli di rischio a ciascuna sessione utente: Alto, Medio, Basso o Nessun rischio.

Valuta attentamente le opzioni a tua disposizione quando modifichi il metodo di applicazione da a. Audit-onlyFull-function Le risposte automatiche che applichi ai livelli di rischio influiscono sul livello di rischio che Amazon Cognito assegna alle sessioni utente successive che presentano le stesse caratteristiche. Ad esempio, dopo aver scelto di non intraprendere alcuna azione, o Allow (Consentire), le sessioni utente che Amazon Cognito inizialmente ha definito ad alto rischio, le successive sessioni simili vengono considerate da Amazon Cognito a basso rischio.

Per ogni livello di rischio, puoi scegliere tra le seguenti opzioni:

Opzione	Azione
Consenti	Gli utenti possono accedere senza un ulteriore fattore.
MFA facoltativa	Gli utenti per i quali è configurata l'autenti cazione con un secondo fattore, devono completare una seconda sfida per poter accedere. I secondi fattori disponibili sono la verifica tramite SMS e token di software TOTP. Gli utenti senza un secondo fattore configurato possono accedere con un solo set di credenzia li.
Richiedi MFA	Gli utenti per i quali è configurata l'autenti cazione con un secondo fattore, devono completare una seconda sfida per poter accedere. Amazon Cognito blocca l'accesso agli utenti che non hanno un secondo fattore configurato.
Blocco	Amazon Cognito blocca tutti i tentativi di accesso al livello di rischio designato.

> **Note**
>
> Non è necessario verificare i numeri di telefono da utilizzare per la ricezione di SMS come secondo fattore di autenticazione.

Aggiunta di dati di dispositivo e sessione utente alle richieste API

Puoi raccogliere e trasmettere informazioni sulla sessione dell'utente alla protezione dalle minacce di Amazon Cognito quando utilizzi l'API per registrarlo, farlo accedere e reimpostare la sua password. Queste informazioni includono l'indirizzo IP dell'utente e un identificatore univoco del dispositivo.

Potrebbe essere presente un dispositivo di rete intermedio tra gli utenti e Amazon Cognito, ad esempio un servizio proxy o un server applicazioni. Puoi raccogliere i dati contestuali degli utenti e passarli ad Amazon Cognito in modo che l'autenticazione adattiva calcoli il rischio in base alle caratteristiche dell'endpoint utente, anziché al server o al proxy. Se l'app lato client chiama direttamente le operazioni dell'API Amazon Cognito, l'autenticazione adattiva registra automaticamente l'indirizzo IP di origine. Tuttavia, non registra altre informazioni sul dispositivo, ad esempio `user-agent`, a meno che non venga acquisita anche un'impronta del dispositivo.

Genera questi dati con la libreria di raccolta dati contestuali di Amazon Cognito e inviali alla protezione dalle minacce di Amazon Cognito con ContextDatai parametri and. UserContextData La libreria di raccolta dei dati contestuali è inclusa negli AWS SDK. Per ulteriori informazioni, consulta Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili. Puoi inviarlo `ContextData` se disponi del piano di funzionalità Plus. Per ulteriori informazioni, consulta Impostazione della protezione dalle minacce.

Quando si chiamano le seguenti operazioni API autenticate di Amazon Cognito dal server applicazioni, viene passato l'IP del dispositivo dell'utente nel parametro `ContextData`. Inoltre, passa il nome e il percorso del server, nonché i dati relativi all'impronta del dispositivo.

- AdminInitiateAuth
- AdminRespondToAuthChallenge

Quando richiami operazioni API non autenticate di Amazon Cognito, puoi inviare una richiesta di protezione dalle minacce di Amazon `UserContextData` Cognito. Questi dati includono un'impronta del dispositivo nel parametro `EncodedData`. Puoi anche passare un parametro `IpAddress` in `UserContextData` se vengono soddisfatte le seguenti condizioni:

- Il tuo pool di utenti è incluso nel piano di funzionalità Plus. Per ulteriori informazioni, consulta Piani di funzionalità del pool di utenti.
- Il client dell'app ha un segreto del client. Per ulteriori informazioni, consulta Application-specific impostazioni con i client dell'app.
- Hai attivato l'opzione Accept additional user context data (Accetta dati utente contestuali aggiuntivi) nel client dell'app. Per ulteriori informazioni, consulta Accettazione di dati contestuali utente aggiuntivi (Console di gestione AWS).

L'app può popolare il parametro `UserContextData` con dati di impronte del dispositivo codificati e l'indirizzo IP del dispositivo dell'utente nelle seguenti operazioni API non autenticate di Amazon Cognito.

- InitiateAuth
- RespondToAuthChallenge
- SignUp
- ConfirmSignUp
- ForgotPassword
- ConfirmForgotPassword
- ResendConfirmationCode

Accettazione di dati contestuali utente aggiuntivi (Console di gestione AWS)

Il pool di utenti accetta un indirizzo IP nel parametro `UserContextData` dopo aver attivato la funzionalità Accept additional user context data (Accetta dati utente contestuali aggiuntivi). Non è necessario attivare questa funzionalità se:

- I tuoi utenti accedono solo con operazioni API autenticate, ad esempio AdminInitiateAuth , e tu utilizzi il `ContextData` parametro.
- Vuoi solo che le tue operazioni API non autenticate inviino un'impronta digitale del dispositivo, ma non un indirizzo IP, alla protezione dalle minacce di Amazon Cognito.

Aggiorna il client dell'app come indicato di seguito nella console Amazon Cognito per aggiungere il supporto per ulteriori dati utente contestuali.

1. Accedi alla console Amazon Cognito.
2. Nel riquadro di navigazione, scegli Manage your User Pools (Gestisci i tuoi bacini d'utenza) e scegli i bacini d'utenza che intendi modificare.
3. Scegli il menu App client.
4. Scegli o crea un client per l'app. Per ulteriori informazioni, consulta Configurazione di un client dell'app per un pool di utenti.
5. Scegli Edit (Modifica) nel container App client information (Informazioni sul client dell'app).
6. Nell'area Advanced authentication settings (Impostazioni di autenticazione avanzate) per il client dell'app, scegli Accept additional user context data (Accetta dati utente contestuali aggiuntivi).

7. Scegli Save changes (Salva modifiche).

Per configurare il client dell'app in modo che accetti i dati contestuali dell'utente nell'API Amazon Cognito, imposta su `EnablePropagateAdditionalUserContextData` `true` in una richiesta CreateUserPoolClientor UpdateUserPoolClient. Per informazioni su come utilizzare la protezione dalle minacce nella tua app web o mobile, consultaRaccolta di dati per la protezione dalle minacce nelle applicazioni. Quando l'app chiama Amazon Cognito dal server, raccogli i dati di contesto degli utenti dal lato client. Di seguito è riportato un esempio che utilizza il metodo JavaScript `getData` SDK.

```
var EncodedData =
 AmazonCognitoAdvancedSecurityData.getData(username, userPoolId, clientId);
```

Quando progetti la tua app per utilizzare l'autenticazione adattiva, ti consigliamo di integrare l'ultimo SDK Amazon Cognito nell'app. L'ultima versione dell'SDK raccoglie informazioni di impronta digitale del dispositivo come ID, modello e fuso orario del dispositivo. Per ulteriori informazioni sugli SDK di Amazon Cognito, consulta Installazione di un SDK del bacino d'utenza. La protezione dalle minacce di Amazon Cognito salva e assegna un punteggio di rischio solo agli eventi inviati dall'app nel formato corretto. Se Amazon Cognito restituisce una risposta di errore, controlla che la richiesta includa un hash segreto valido e che il parametro `IPaddress` sia un indirizzo IPv4 o IPv6 valido.

Risorse **`ContextData`** e **`UserContextData`**

- AWS Amplify SDK per Android: GetUserContextData
- AWS Amplify SDK per iOS: utente ContextData
- JavaScript: amazon-cognito-advanced-security-data.min.js

Visualizzazione ed esportazione della cronologia degli eventi degli utenti

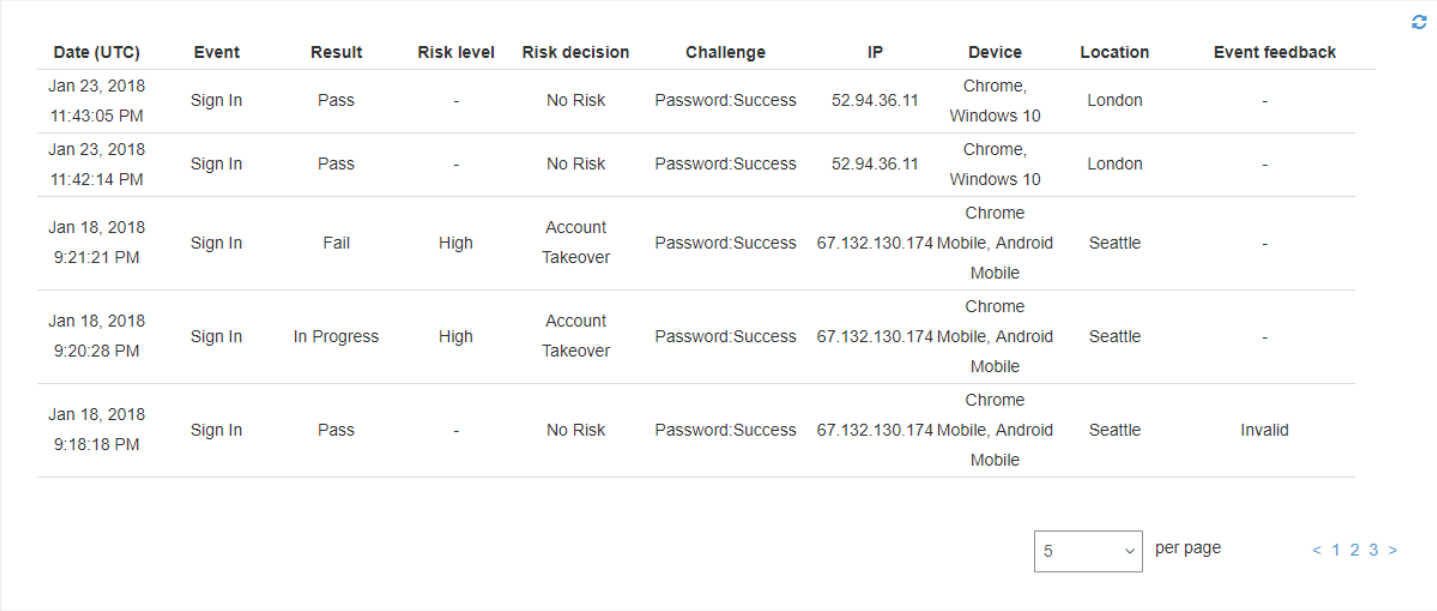
Amazon Cognito genera un registro per ogni evento di autenticazione di un utente quando abiliti la protezione dalle minacce. Per impostazione predefinita, puoi visualizzare i log degli utenti nel menu Utenti nella console Amazon Cognito o con AdminListUserAuthEventsil funzionamento dell'API. Puoi anche esportare questi eventi su un sistema esterno come CloudWatch Logs, Amazon S3 o Amazon Data Firehose. La funzionalità di esportazione può rendere le informazioni di sicurezza sull'attività degli utenti nella tua applicazione più accessibili ai tuoi sistemi di analisi della sicurezza.

Argomenti

- Visualizzazione della cronologia degli eventi degli utenti (Console di gestione AWS)
- Visualizzazione della cronologia degli eventi dell'utente () API/CLI
- Esportazione degli eventi di autenticazione degli utenti

Visualizzazione della cronologia degli eventi degli utenti (Console di gestione AWS)

Per visualizzare la cronologia degli accessi di un utente, puoi scegliere l'utente dal menu Utenti nella console Amazon Cognito. Amazon Cognito mantiene la cronologia eventi dell'utente per due anni.



Date (UTC)	Event	Result	Risk level	Risk decision	Challenge	IP	Device	Location	Event feedback
Jan 23, 2018 11:43:05 PM	Sign In	Pass	-	No Risk	Password:Success	52.94.36.11	Chrome, Windows 10	London	-
Jan 23, 2018 11:42:14 PM	Sign In	Pass	-	No Risk	Password:Success	52.94.36.11	Chrome, Windows 10	London	-
Jan 18, 2018 9:21:21 PM	Sign In	Fail	High	Account Takeover	Password:Success	67.132.130.174	Chrome Mobile, Android Mobile	Seattle	-
Jan 18, 2018 9:20:28 PM	Sign In	In Progress	High	Account Takeover	Password:Success	67.132.130.174	Chrome Mobile, Android Mobile	Seattle	-
Jan 18, 2018 9:18:18 PM	Sign In	Pass	-	No Risk	Password:Success	67.132.130.174	Chrome Mobile, Android Mobile	Seattle	Invalid

5 per page < 1 2 3 >

Ogni evento di accesso ha un ID evento. L'evento contiene anche dati di contesto corrispondenti, come la posizione, i dettagli del dispositivo e i risultati del rilevamento dei rischi.

Puoi anche correlare l'ID evento con il token emesso da Amazon Cognito nel momento in cui ha registrato l'evento. Gli ID e i token di accesso includono questo ID evento nel payload. Amazon Cognito mette inoltre in correlazione l'utilizzo del token di aggiornamento con l'ID evento originale. Puoi tenere traccia dell'ID evento originale risalendo fino all'ID dell'evento di accesso che ha avuto come esito il rilascio dei token di Amazon Cognito. Puoi tracciare l'utilizzo di un token all'interno del sistema di autenticazione fino a un determinato evento di autenticazione. Per ulteriori informazioni, consulta Comprensione dei token web JSON (JWT) del pool di utenti.

Visualizzazione della cronologia degli eventi dell'utente () API/CLI

Puoi interrogare la cronologia degli eventi degli utenti con l'operazione dell'API Amazon Cognito AdminListUserAuthEventso con AWS Command Line Interface (AWS CLI) con admin-list-user-auth-events.

AdminListUserAuthEvents request

Il seguente corpo della richiesta `AdminListUserAuthEvents` restituisce il registro delle attività più recente per un utente.

```
{
  "UserPoolId": "us-west-2_EXAMPLE",
  "Username": "myexampleuser",
  "MaxResults": 1
}
```

admin-list-user-auth-events request

La seguente richiesta `admin-list-user-auth-events` restituisce il registro delle attività più recente per un utente.

```
aws cognito-idp admin-list-user-auth-events --max-results 1 --username myexampleuser
 --user-pool-id us-west-2_EXAMPLE
```

Response

Amazon Cognito restituisce lo stesso corpo di risposta JSON a entrambe le richieste. Di seguito è riportato un esempio di risposta per un evento di accesso gestito che non è risultato contenere fattori di rischio:

```
{
    "AuthEvents": [
        {
            "EventId": "[event ID]",
            "EventType": "SignIn",
            "CreationDate": "[Timestamp]",
            "EventResponse": "Pass",
            "EventRisk": {
                "RiskDecision": "NoRisk",
                "CompromisedCredentialsDetected": false
            },
            "ChallengeResponses": [
                {
                    "ChallengeName": "Password",
                    "ChallengeResponse": "Success"
                }
            ],
```

```
            "EventContextData": {
                "IpAddress": "192.168.2.1",
                "DeviceName": "Chrome 125, Windows 10",
                "Timezone": "-07:00",
                "City": "Bellevue",
                "Country": "United States"
            }
        }
    ],
    "NextToken": "[event ID]#[Timestamp]"
}
```

Esportazione degli eventi di autenticazione degli utenti

Configura il tuo pool di utenti per esportare gli eventi degli utenti dalla protezione dalle minacce a un sistema esterno. I sistemi esterni supportati (Amazon S3 CloudWatch , Logs e Amazon Data FireHose) potrebbero aggiungere costi alla AWS fattura per i dati inviati o recuperati. Per ulteriori informazioni, consulta Esportazione dei log delle attività degli utenti per la protezione dalle minacce.

Console di gestione AWS

1. Accedi alla console Amazon Cognito.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Log streaming. Seleziona Edit (Modifica).
5. In Stato della registrazione, seleziona la casella di controllo accanto a Attiva l'esportazione del registro delle attività degli utenti.
6. In Logging destination, scegli il tipo di log con Servizio AWS cui vuoi gestire i log: gruppo di CloudWatch log, flusso Amazon Data Firehose o bucket S3.
7. La tua selezione compila il selettore di risorse con il tipo di risorsa corrispondente. Seleziona un gruppo di log, uno stream o un bucket dall'elenco. Puoi anche selezionare il pulsante Crea Console di gestione AWS per accedere al servizio selezionato e creare una nuova risorsa.
8. Seleziona Salva modifiche.

API

Scegli un tipo di destinazione per i registri delle attività degli utenti.

Di seguito è riportato un esempio di corpo della `SetLogDeliveryConfiguration` richiesta che imposta un flusso Firehose come destinazione del registro.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "FirehoseConfiguration": {
            "StreamArn": "arn:aws:firehose:us-west-2:123456789012:deliverystream/
example-user-pool-activity-exported"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Di seguito è riportato un esempio di corpo di `SetLogDeliveryConfiguration` richiesta che imposta un bucket Amazon S3 come destinazione del log.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "S3Configuration": {
            "BucketArn": "arn:aws:s3:::amzn-s3-demo-logging-bucket"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Di seguito è riportato un esempio di corpo della `SetLogDeliveryConfiguration` richiesta che imposta un gruppo di CloudWatch log come destinazione del log.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "CloudWatchLogsConfiguration": {
```

```
            "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-group:DOC-
EXAMPLE-LOG-GROUP"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Fornitura del feedback sugli eventi

Il feedback sugli eventi influisce in tempo reale sulla valutazione dei rischi e migliora l'algoritmo di valutazione dei rischi nel corso del tempo. Puoi fornire un feedback sulla validità dei tentativi di accesso tramite la console Amazon Cognito e le operazioni delle API.

> **Note**
>
> Il tuo feedback sugli eventi influisce sul livello di rischio che Amazon Cognito assegna alle sessioni utente successive con le stesse caratteristiche.

Nella console Amazon Cognito, scegli un utente dal menu Utenti e seleziona Fornisci feedback sull'evento. Puoi controllare i dettagli dell'evento e quindi scegliere Set as valid (Imposta come valido) o Set as invalid (Imposta come non valido).

La console elenca la cronologia degli accessi nei dettagli utente nel menu Utenti. Se selezioni una voce, puoi contrassegnare l'evento come valido o non valido. Puoi anche fornire feedback tramite l'operazione AdminUpdateAuthEventFeedbackAPI del pool di utenti e tramite il AWS CLI comando admin-update-auth-event-feedback.

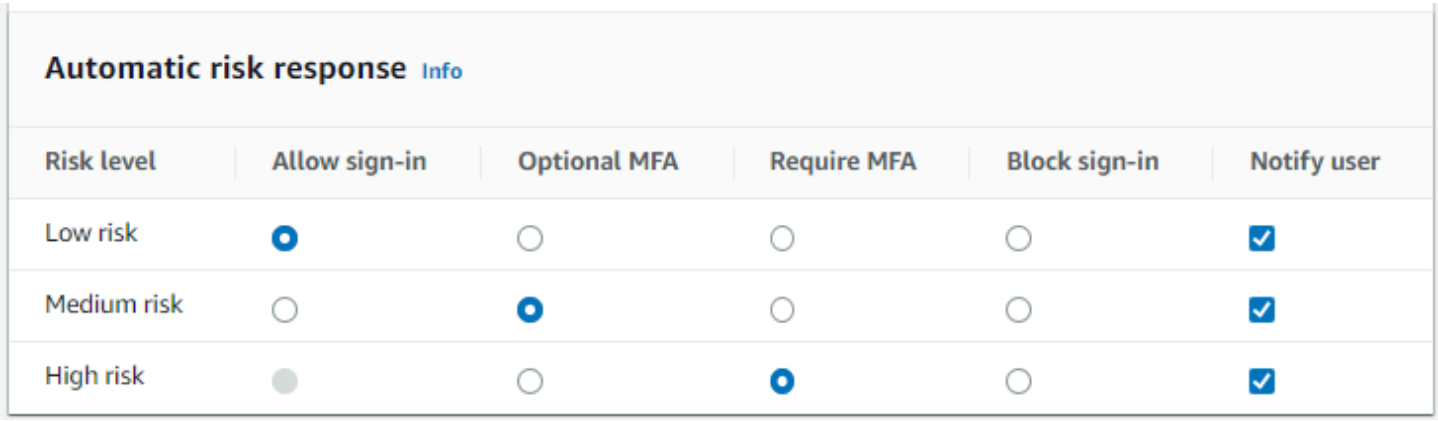
Quando nella console Amazon Cognito selezioni Set as valid (Imposta come valido) o nell'API fornisci il valore `valid` per `FeedbackValue`, indichi ad Amazon Cognito che la sessione utente che ha valutato con un certo livello di rischio è invece attendibile. Quando nella console Amazon Cognito selezioni Set as invalid (Imposta come non valido) o nell'API fornisci il valore `invalid` per `FeedbackValue`, indichi ad Amazon Cognito che la sessione utente non è attendibile o che non ha valutato un livello di rischio sufficientemente elevato.

Invio di messaggi di notifica

Con la protezione dalle minacce, Amazon Cognito può avvisare gli utenti dei tentativi di accesso rischiosi. Amazon Cognito può anche chiedere agli utenti di selezionare un link per indicare se l'accesso è valido o non valido. Amazon Cognito utilizza questo feedback per migliorare l'accuratezza del rilevamento del rischio per il tuo bacino d'utenza

> **Note**
>
> Amazon Cognito invia messaggi di notifica agli utenti solo quando la loro azione genera una risposta automatica al rischio: blocca l'accesso, consenti l'accesso, imposta la MFA su opzionale o richiedi l'autenticazione a più fattori. Ad alcune richieste potrebbe essere assegnato un livello di rischio, ma non generano risposte automatiche di autenticazione adattiva al rischio; per queste, il tuo pool di utenti non invia notifiche. Ad esempio, è possibile che vengano registrate password errate con una classificazione del rischio, ma la risposta di Amazon Cognito è di non riuscire ad accedere, non di applicare una regola di autenticazione adattiva.

Nella sezione Automatic risk response (Risposta automatica al rischio) scegli Notify Users (Notifica agli utenti) per i casi di basso, medio e alto rischio.



Amazon Cognito invia le notifiche e-mail agli utenti a prescindere dalla verifica dell'indirizzo e-mail.

Puoi personalizzare i messaggi e-mail di notifica e fornire i messaggi nelle versioni in formato di testo normale e in formato HTML. Per personalizzare le notifiche e-mail, apri i modelli di posta elettronica da Adaptive authentication messages nella configurazione di protezione dalle minacce. Per ulteriori informazioni sui modelli di e-mail, consulta Modelli dei messaggi.

Raccolta di dati per la protezione dalle minacce nelle applicazioni

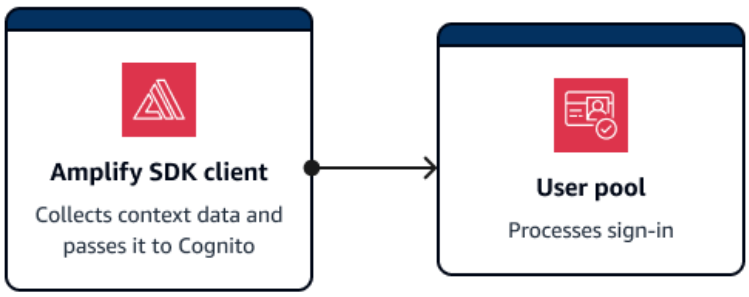
L'autenticazione adattiva di Amazon Cognito valuta i livelli di rischio per il tentativo di acquisizione dell'account sulla base dei dettagli contestuali dei tentativi di accesso degli utenti. L'applicazione deve aggiungere dati contestuali alle richieste API in modo che la protezione dalle minacce di Amazon Cognito possa valutare con maggiore precisione il rischio. I dati contestuali sono informazioni quali indirizzo IP, agente del browser, informazioni sul dispositivo e intestazioni di richiesta che forniscono informazioni contestuali su come un utente si è connesso al tuo pool di utenti.

La responsabilità principale di un'applicazione che invia questo contesto ad Amazon Cognito è `EncodedData` un parametro nelle richieste di autenticazione ai pool di utenti. Per aggiungere questi dati alle tue richieste, puoi implementare Amazon Cognito con un SDK che genera automaticamente queste informazioni per te oppure puoi implementare un modulo per JavaScript iOS o Android che raccolga questi dati. Client-onlyle applicazioni che effettuano richieste dirette ad Amazon Cognito devono implementare AWS Amplify gli SDK. Client-serverle applicazioni che dispongono di un server o di un componente API intermedio devono implementare un modulo SDK separato.

Nei seguenti scenari, il front-end di autenticazione gestisce la raccolta dei dati contestuali degli utenti senza alcuna configurazione aggiuntiva:

- L'accesso gestito raccoglie e invia automaticamente i dati contestuali per la protezione dalle minacce.
- Tutte le AWS Amplify librerie dispongono di una raccolta di dati contestuali integrata nei metodi di autenticazione.

Invio di dati contestuali utente in applicazioni solo client con Amplify



Gli SDK Amplify supportano i client mobili che si autenticano direttamente con Amazon Cognito. I clienti di questo tipo effettuano richieste API dirette alle operazioni API pubbliche di Amazon Cognito.

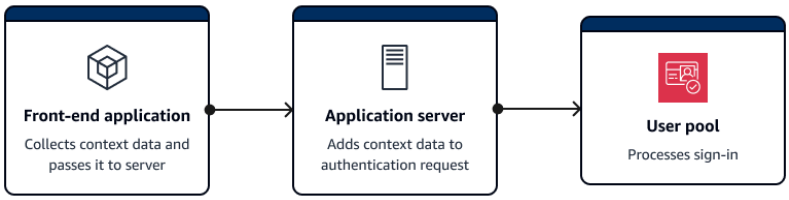
Per impostazione predefinita, i client Amplify raccolgono automaticamente dati contestuali per la protezione dalle minacce.

Le applicazioni Amplify JavaScript with sono un'eccezione. Richiedono l'aggiunta di un JavaScript modulo che raccolga i dati relativi al contesto dell'utente.

In genere, un'applicazione in questa configurazione utilizza operazioni API non autenticate come e. InitiateAuthRespondToAuthChallenge L'UserContextDataoggetto aiuta a valutare i rischi in modo più accurato per queste operazioni. Gli SDK Amplify aggiungono informazioni sul dispositivo e sulla sessione a un parametro di. `EncodedData` `UserContextData`

Raccolta di dati contestuali nelle applicazioni client-server

Alcune applicazioni dispongono di un livello front-end che raccoglie i dati di autenticazione degli utenti e un livello di back-end dell'applicazione che invia le richieste di autenticazione ad Amazon Cognito. Si tratta di un'architettura comune nei server Web e nelle applicazioni supportate da microservizi. In queste applicazioni, è necessario importare una libreria pubblica di raccolta di dati contestuali.



In genere, un server applicativo in questa configurazione utilizza operazioni API autenticate come e. AdminInitiateAuthAdminRespondToAuthChallenge L'ContextDataoggetto aiuta Amazon Cognito a valutare i rischi in modo più accurato per queste operazioni. I contenuti `ContextData` sono i dati codificati che il front-end ha passato al server e dettagli aggiuntivi derivanti dalla richiesta HTTP dell'utente al server. Questi dettagli contestuali aggiuntivi, come le intestazioni HTTP e l'indirizzo IP, forniscono al server delle applicazioni le caratteristiche dell'ambiente dell'utente.

Il server delle applicazioni potrebbe anche effettuare l'accesso con operazioni API non autenticate come e. InitiateAuthRespondToAuthChallenge L'UserContextDataoggetto consente l'analisi del rischio di protezione dalle minacce in queste operazioni. Le operazioni nelle librerie di raccolta dati di contesto pubblico disponibili aggiungono informazioni di sicurezza al `EncodedData` parametro nelle richieste di autenticazione. Inoltre, configura il tuo pool di utenti per accettare dati contestuali aggiuntivi e aggiungi l'IP di origine dell'utente al `IpAddress` parametro di`UserContextData`.

Per aggiungere dati contestuali alle applicazioni client-server

1. Nella tua applicazione front-end, raccogli dati contestuali codificati dal client con un iOS, Android o un modulo. JavaScript
2. Trasmetti i dati codificati e i dettagli della richiesta di autenticazione al tuo server delle applicazioni.
3. Nel server delle applicazioni, estrai l'indirizzo IP dell'utente, le intestazioni HTTP pertinenti, il nome del server richiesto e il percorso richiesto dalla richiesta HTTP. Inserisci questi valori nel ContextDataparametro della tua richiesta API ad Amazon Cognito.
4. Compila il `EncodedData` parametro of `ContextData` nella tua richiesta API con i dati codificati del dispositivo raccolti dal modulo SDK. Aggiungi questi dati contestuali alla richiesta di autenticazione.

Librerie di dati contestuali per applicazioni client-server

JavaScript

Il `amazon-cognito-advanced-security-data.min.js` modulo raccoglie dati `EncodedData` che potete passare al vostro server delle applicazioni.

Aggiungi il `amazon-cognito-advanced-security-data.min.js` modulo alla tua JavaScript configurazione. Regione AWS Sostituiscilo `<region>` con uno dei seguenti:`us-east-1`,`us-east-2`,`us-west-2`,`eu-west-1`,`eu-west-2`, o`eu-central-1`.

```
<script src="https://amazon-cognito-assets.<region>.amazoncognito.com/amazon-cognito-advanced-security-data.min.js"></script>
```

Per generare un `encodedContextData` oggetto da utilizzare nel `EncodedData` parametro, aggiungete quanto segue all'origine JavaScript dell'applicazione:

```
var encodedContextData = AmazonCognitoAdvancedSecurityData.getData(_username,
 _userpoolId, _userPoolClientId);
```

iOS/Swift

Per generare dati contestuali, le applicazioni iOS possono integrare il modulo Mobile SDK for iOS AWSCognitoIdentityProviderASF.

Per raccogliere dati contestuali codificati per la protezione dalle minacce, aggiungi il seguente frammento all'applicazione:

```
import AWSCognitoIdentityProviderASF

let deviceId = getDeviceId()
let encodedContextData = AWSCognitoIdentityProviderASF.userContextData(
                            userPoolId,
                            username: username,
                            deviceId: deviceId,
                            userPoolClientId: userPoolClientId)

/**
 * Reuse DeviceId from keychain or generate one for the first time.
 */
func getDeviceId() -> String {
    let deviceIdKey = getKeyChainKey(namespace: userPoolId, key:
 "AWSCognitoAuthAsfDeviceId")

   if let existingDeviceId = self.keychain.string(forKey: deviceIdKey) {
        return existingDeviceId
    }

    let newDeviceId = UUID().uuidString
    self.keychain.setString(newDeviceId, forKey: deviceIdKey)
    return newDeviceId
}

/**
 * Get a namespaced keychain key given a namespace and key
 */
func getKeyChainKey(namespace: String, key: String) -> String {
    return "\(namespace).\(key)"
}
```

Android

Per generare dati contestuali, le applicazioni Android possono integrare il modulo Mobile SDK for Android aws-android-sdk-cognitoidentityprovider-asf.

Per raccogliere dati contestuali codificati per la protezione dalle minacce, aggiungi il seguente frammento all'applicazione:

```
UserContextDataProvider provider = UserContextDataProvider.getInstance();
// context here is android application context.
String encodedContextData = provider.getEncodedContextData(context, username,
 userPoolId, userPoolClientId);
```

Associa un AWS WAF ACL web con un pool di utenti

AWS WAF è un firewall per applicazioni Web. Con una AWS WAF lista di controllo degli accessi Web (Web ACL), puoi proteggere il tuo pool di utenti da richieste indesiderate all'interfaccia utente ospitata classica, all'accesso gestito e agli endpoint del servizio API Amazon Cognito. Una ACL Web offre un controllo granulare su tutte le richieste Web HTTPS a cui risponde il pool di utenti. Per ulteriori informazioni sulle ACL Web AWS WAF , consulta Gestione e utilizzo di una lista di controllo degli accessi Web (ACL Web) nella Guida per gli sviluppatori di AWS WAF .

Quando hai un ACL AWS WAF Web associato a un pool di utenti, Amazon Cognito inoltra le intestazioni e i contenuti selezionati non riservati delle richieste degli utenti a. AWS WAF AWS WAF ispeziona il contenuto della richiesta, la confronta con le regole che hai specificato nell'ACL web e restituisce una risposta ad Amazon Cognito. I componenti della richiesta inoltrati da Amazon Cognito AWS WAF dipendono dal tipo di richiesta. Per l'accesso gestito e l'interfaccia utente ospitata classica, Amazon Cognito non inoltra il corpo della richiesta. Per ulteriori informazioni, consulta Cose da sapere AWS WAF ACL web e Amazon Cognito.

Cose da sapere AWS WAF ACL web e Amazon Cognito

- Non puoi configurare le regole Web ACL in modo che corrispondano alle informazioni di identificazione personale (PII) nelle richieste dei pool di utenti, ad esempio nomi utente, password, numeri di telefono o indirizzi e-mail. Questi dati non saranno disponibili per. AWS WAF Configura invece le regole ACL Web in modo che corrispondano ai dati della sessione nelle intestazioni, nel percorso e nel corpo, come gli indirizzi IP, gli agenti del browser e le operazioni API richieste.
- Amazon Cognito supporta le regole ACL Web basate sul corpo solo per le richieste all'API dei pool di utenti. Per l'accesso gestito e la classica interfaccia utente ospitata, Amazon Cognito non inoltra il corpo della richiesta e ispeziona solo le intestazioni e gli altri componenti non del corpo.
- Le regole Web ACL possono restituire una risposta di blocco personalizzata solo quando il browser carica una pagina di accesso gestita. Le richieste inviate dalla pagina senza ricaricarsi, ad esempio quando un utente invia il modulo di accesso, ricevono invece una risposta di blocco predefinita`403`.

- Le richieste bloccate da AWS WAF non vengono conteggiate ai fini della quota di richiesta per nessun tipo di richiesta. Il AWS WAF gestore viene chiamato prima dei gestori di API-level limitazione.
- Quando si crea una ACL Web, passa del tempo prima che l'ACL Web si propaghi completamente e sia disponibile per Amazon Cognito. Il tempo di propagazione può variare da pochi secondi a diversi minuti. AWS WAF restituisce a `WAFUnavailableEntityException`quando si tenta di associare un ACL Web prima che si sia completamente propagato.
- È possibile associare un ACL Web a ciascun pool di utenti.
- La richiesta potrebbe comportare un payload superiore ai limiti di quanto AWS WAF può controllare. Consulta Gestione dei componenti di richieste sovradimensionate nella AWS WAF Developer Guide per scoprire come configurare come AWS WAF gestire le richieste sovradimensionate da Amazon Cognito.
- Non puoi associare un ACL Web che utilizza la prevenzione dell'acquisizione di account AWS WAF Fraud Control (ATP) a un pool di utenti Amazon Cognito. La funzionalità ATP fa parte del gruppo di regole gestite. `AWS-AWSManagedRulesATPRuleSet` Prima di associare un ACL Web a un pool di utenti, assicurati che non utilizzi questo gruppo di regole gestito.
- Quando hai un ACL AWS WAF web associato a un pool di utenti e una regola nell'ACL web presenta un CAPTCHA, ciò può causare un errore irreversibile nella registrazione TOTP con accesso gestito. Per creare una regola che abbia un'azione CAPTCHA e non influisca sul TOTP dell'accesso gestito, consulta. Configurazione del AWS WAF ACL web per accesso gestito TOTP MFA
- Quando una regola nel tuo ACL Web restituisce una risposta di blocco personalizzata, Amazon Cognito aggiunge Content-Security-Policy un'intestazione a tale risposta. Questa politica consente l'utilizzo di script solo dalla stessa origine del dominio del pool di utenti e dagli script AWS WAF CAPTCHA e challenge forniti da Amazon Cognito. Blocca qualsiasi altro script in linea o di terze parti, incluso uno script utilizzato dal corpo di risposta personalizzato per reindirizzare l'utente. Per ulteriori informazioni, consulta Content-Security-Policy sulle risposte personalizzate ai blocchi.
- Quando una regola nel tuo ACL Web restituisce una risposta di blocco personalizzata con un codice di stato HTTP pari o superiore a 500 in risposta a un accesso gestito o a una richiesta di interfaccia utente ospitata classica, Amazon Cognito restituisce invece `403 (Forbidden)` un codice di stato. Per ulteriori informazioni, consulta Codici di stato nelle risposte ai blocchi personalizzate.

AWS WAF esamina le richieste ai seguenti endpoint.

Accesso gestito e interfaccia utente classica ospitata

Richieste a tutti gli endpoint in Endpoint del pool di utenti e riferimento per l'accesso gestito.

Operazioni API pubbliche

Richieste dalla tua app all'API Amazon Cognito che non utilizzano AWS credenziali per l'autorizzazione. Ciò include operazioni API come InitiateAuth, e RespondToAuthChallenge. GetUser Le operazioni API incluse nell'ambito di AWS WAF non richiedono l'autenticazione con AWS credenziali. Non sono autenticate o sono autorizzate con una stringa di sessione o un token di accesso. Per ulteriori informazioni, consulta Elenco delle operazioni API raggruppate per modello di autorizzazione.

È possibile configurare le regole nell'ACL Web con operazioni di regole Count (Conteggio), Allow (Consenti), Block (Blocca), oppure che presentano un CAPTCHA in risposta a una richiesta che corrisponde a una regola. Per ulteriori informazioni, consulta Regole personalizzate AWS WAF nella Guida per gli sviluppatori di AWS WAF . A seconda dell'operazione della regola, è possibile personalizzare la risposta che Amazon Cognito restituisce agli utenti.

> **Important**
>
> Le opzioni per personalizzare la risposta all'errore dipendono dal modo in cui si effettua una richiesta API.
>
> - È possibile personalizzare il codice di errore e il corpo di risposta delle richieste di accesso gestite. Amazon Cognito restituisce codici di errore personalizzati pari o superiori a 500. `403` Per ulteriori informazioni, consulta Codici di stato nelle risposte ai blocchi personalizzate. Puoi solo presentare un CAPTCHA che il tuo utente deve risolvere nell'accesso gestito.
> - Per le richieste effettuate con l'API dei pool di utenti di Amazon Cognito, puoi personalizzare il corpo della risposta di una richiesta che riceve una risposta Blocca. È inoltre possibile specificare un codice di errore personalizzato compreso tra 400 e 499.
> - Il AWS Command Line Interface (AWS CLI) e gli AWS SDK restituiscono un **`ForbiddenException`** errore alle richieste che producono una risposta Block o CAPTCHA.

Content-Security-Policy sulle risposte personalizzate ai blocchi

Se la risposta ai blocchi personalizzata si basa su script, questa politica può influire sul suo comportamento. Quando una regola ACL Web restituisce una risposta di blocco personalizzata all'accesso gestito o alla classica interfaccia utente ospitata, Amazon Cognito aggiunge Content-Security-Policy un'intestazione (CSP) a tale risposta. Il CSP è un meccanismo di sicurezza del browser che limita gli script e le altre risorse che una pagina può caricare. La politica limita la risposta come segue:

- La risposta può includere script e altre risorse provenienti dalla stessa origine del dominio del pool di utenti.
- La risposta può includere il AWS WAF CAPTCHA e gli script di sfida forniti da Amazon Cognito. Amazon Cognito consente di inserire questi script in modo che AWS WAF CAPTCHA e le azioni relative alle regole di sfida continuino a funzionare.

Qualsiasi altro script in linea e qualsiasi script ospitato su un dominio diverso dal dominio del pool di utenti non viene eseguito. Ciò include uno script utilizzato da un corpo di risposta personalizzato per reindirizzare l'utente, inviare un modulo o inviare una richiesta di rete.

Se la risposta personalizzata deve reindirizzare l'utente, restituisci un codice di stato HTTP 3xx e un'`Location`intestazione anziché un reindirizzamento basato su script. Questo modello non dipende dall'esecuzione dello script e non è influenzato dal CSP.

Questo comportamento si applica a tutti i Regioni AWS casi in cui è possibile associare un ACL Web a un pool di utenti.

Codici di stato nelle risposte ai blocchi personalizzate

Amazon Cognito non restituisce i codici di `5xx` stato delle risposte ai blocchi AWS WAF personalizzate. Quando una regola Web ACL restituisce una risposta di blocco personalizzata con un codice di stato pari o superiore a 500, Amazon Cognito risponde invece con `403 (Forbidden)` un codice di stato. La risposta include ancora le intestazioni e il corpo della risposta personalizzata. Questo comportamento si applica come segue:

- Pagine di accesso gestite: quando il browser carica una pagina di accesso gestita, Amazon Cognito restituisce `403` le intestazioni e il testo della risposta personalizzata. Le richieste inviate dalla pagina senza ricaricamento, ad esempio l'invio di moduli, ricevono la risposta di `403` blocco predefinita senza intestazioni o testo personalizzati.

- Pagine dell'interfaccia utente ospitate nella versione classica e altri endpoint del pool di utenti: Amazon Cognito `403` restituisce le intestazioni e il corpo della risposta personalizzata per le pagine dell'interfaccia utente ospitate classiche. Il provider di identità e gli endpoint relying party si comportano allo stesso modo, indipendentemente dal fatto che il pool di utenti utilizzi l'accesso gestito o il branding dell'interfaccia utente ospitata classica.
- API dei pool di utenti: l'API dei pool di utenti restituisce il codice di stato personalizzato solo quando è compreso tra 400 e 499 e restituisce altri codici di stato. `403`

Associazione di una ACL Web al pool di utenti

Per utilizzare un ACL Web nel tuo pool di utenti, il tuo principale AWS Identity and Access Management (IAM) deve disporre dei seguenti Amazon Cognito AWS WAF e delle autorizzazioni seguenti. Per informazioni sulle AWS WAF autorizzazioni, consulta le autorizzazioni AWS WAF API nella Guida per gli sviluppatori.AWS WAF

JSON

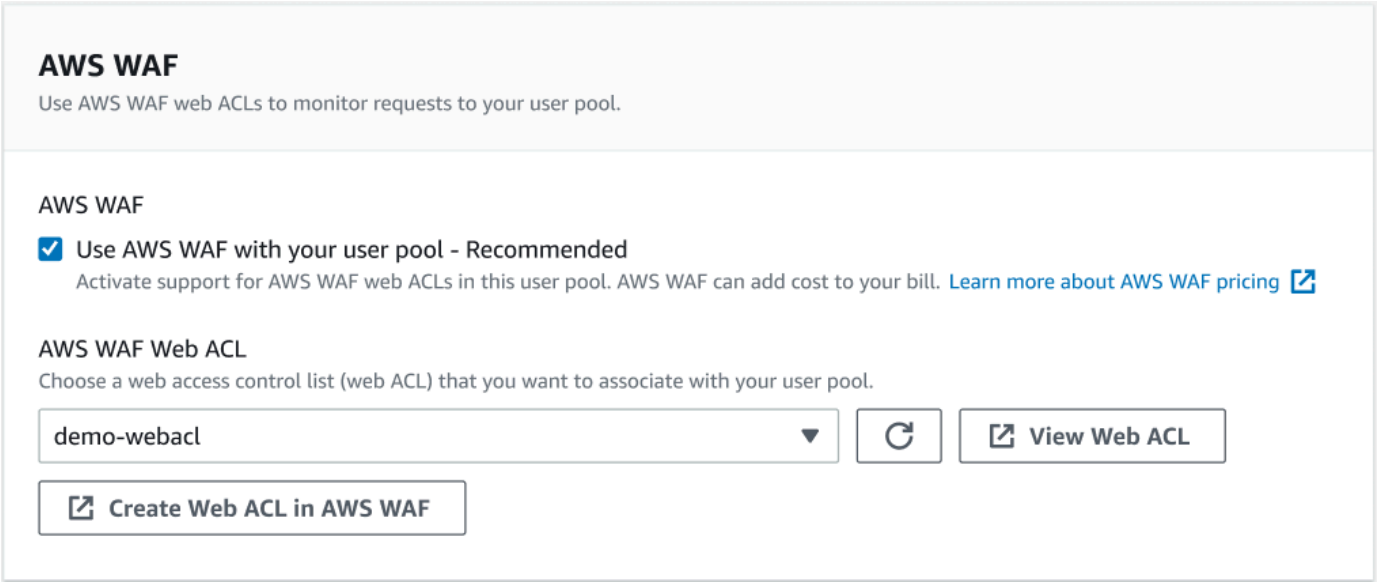
```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "AllowWebACLUserPool",
   "Effect": "Allow",
   "Action": [
    "cognito-idp:ListResourcesForWebACL",
    "cognito-idp:GetWebACLForResource",
    "cognito-idp:AssociateWebACL"
   ],
   "Resource": [
    "arn:aws:cognito-idp:*:123456789012:userpool/*"
   ]
  },
  {
   "Sid": "AllowWebACLUserPoolWAFv2",
   "Effect": "Allow",
   "Action": [
    "wafv2:ListResourcesForWebACL",
    "wafv2:AssociateWebACL",
    "wafv2:DisassociateWebACL",
    "wafv2:GetWebACLForResource"
```

```
    ],
    "Resource": "arn:aws:wafv2:*:123456789012:*/webacl/*/*"
   },
   {
    "Sid": "DisassociateWebACL1",
    "Effect": "Allow",
    "Action": "wafv2:DisassociateWebACL",
    "Resource": "*"
   },
   {
    "Sid": "DisassociateWebACL2",
    "Effect": "Allow",
    "Action": [
     "cognito-idp:DisassociateWebACL"
    ],
    "Resource": [
     "arn:aws:cognito-idp:*:123456789012:userpool/*"
    ]
   }
  ]
}
```

Sebbene sia necessario concedere le autorizzazioni IAM, le azioni elencate riguardano solo le autorizzazioni e non corrispondono a nessuna operazione API.

Per attivare AWS WAF per il tuo pool di utenti e associare un ACL web

1. Accedi alla console Amazon Cognito.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegli la AWS WAFscheda nella sezione Sicurezza.
4. Scegli Modifica.
5. Seleziona Usa AWS WAF con il tuo pool di utenti.



6. Scegliete un ACL AWS WAF Web che avete già creato oppure scegliete Crea ACL Web in AWS WAF per crearne uno in una nuova AWS WAF sessione in. Console di gestione AWS
7. Scegli Save changes (Salva modifiche).

Per associare a livello di codice un ACL Web al tuo pool di utenti in AWS Command Line Interface o in un SDK, utilizza ACL dell'API. AssociateWeb AWS WAF Amazon Cognito non dispone di un'operazione API distinta che associ una lista ACL Web.

Test e registrazione AWS WAF ACL web

Quando imposti un'azione della regola su Count nell'ACL web, AWS WAF aggiunge la richiesta a un conteggio di richieste che corrispondono alla regola. Per testare una ACL Web con il pool di utenti, impostare le operazioni delle regole su Count (Conteggio) e considerare il volume di richieste che corrispondono a ciascuna regola. Ad esempio, se una regola che desideri impostare sull'operazione Block (Blocca) corrisponde a un gran numero di richieste che si ritiene sia un normale traffico utente, potrebbe essere necessario riconfigurare la regola. Per ulteriori informazioni, consulta Testare e ottimizzare AWS WAF le protezioni nella Guida per gli AWS WAF sviluppatori.

Puoi anche configurare AWS WAF per registrare le intestazioni delle richieste in un gruppo di log di Amazon CloudWatch Logs, un bucket Amazon Simple Storage Service (Amazon S3) o un Amazon Data Firehose. Puoi identificare le richieste Amazon Cognito effettuate con l'API dei pool di utenti tramite `x-amzn-cognito-client-id` e `x-amzn-cognito-operation-name`. Le richieste di accesso gestite includono solo l'intestazione. `x-amzn-cognito-client-id` Per ulteriori

informazioni, consulta Registrazione del traffico ACL Web nella Guida per gli sviluppatori di AWS WAF .

AWS WAF gli ACL web sono disponibili in tutti i piani di funzionalità del pool di utenti. Le funzionalità di sicurezza AWS WAF completano la protezione dalle minacce di Amazon Cognito. È possibile attivare entrambe le caratteristiche in un pool di utenti. AWS WAF fattura separatamente per l'ispezione delle richieste del pool di utenti. Per ulteriori informazioni, consulta AWS WAF Prezzi.

I dati della AWS WAF richiesta di registrazione sono soggetti a una fatturazione aggiuntiva da parte del servizio a cui vengono indirizzati i log. Per ulteriori informazioni, consulta Prezzi per la registrazione delle informazioni sul traffico ACL web nella Guida per gli sviluppatori di AWS WAF .

Distinzione tra maiuscole e minuscole del bacino d'utenza

Per impostazione predefinita, i pool di utenti di Amazon Cognito creati in non Console di gestione AWS fanno distinzione tra maiuscole e minuscole. Quando un bacino d'utenza è senza distinzione tra maiuscole e minuscole, user@example.com e User@example.com si riferiscono allo stesso utente. Quando i nomi utente in un pool di utenti sono senza distinzione tra maiuscole e minuscole, anche gli attributi `preferred_username` e `email` sono senza distinzione tra maiuscole e minuscole.

La distinzione tra maiuscole e minuscole non si applica solo agli input degli attributi, ma anche agli output. Mixed-casei valori degli attributi nei pool di utenti senza distinzione tra maiuscole e minuscole vengono appiattiti in minuscolo nell'output di testo del pool di utenti. Esempi di output testuale del pool di utenti sono le risposte UserInfo, le risposte alle interrogazioni degli utenti come l'output di GetUsere gli eventi di input nei trigger Lambda.

Per tenere conto delle impostazioni di sensibilità alle maiuscole e alle minuscole del bacino d'utenza, identifica gli utenti nel codice dell'app in base a un attributo utente alternativo. Dal momento che il caso di un attributo nome utente, nome utente preferito o indirizzo e-mail può variare a seconda dei profili utente, fai riferimento all'attributo `sub`. È inoltre possibile creare un attributo personalizzato immutabile nel bacino d'utenza e assegnare il proprio valore identificativo univoco all'attributo in ogni nuovo profilo utente. Quando crei per la prima volta un utente, puoi scrivere un valore in un attributo personalizzato immutabile.

> **Note**
>
> Indipendentemente dalle impostazioni di distinzione tra maiuscole e minuscole del bacino d'utenza, Amazon Cognito richiede che un utente federato di un provider di identità SAML oppure OIDC passi un'attestazione `NameId` o `sub` univoca e con distinzione tra maiuscole e

minuscole. Per ulteriori informazioni sull'identificatore univoco, sulla distinzione tra maiuscole e minuscole e SAML, consulta. IdPs Implementa l' SP-initated accesso SAML

Creazione di un bacino d'utenza con distinzione tra maiuscole e minuscole

Se create risorse con le operazioni AWS Command Line Interface (AWS CLI) e API, ad esempio CreateUserPool, dovete impostare il parametro `CaseSensitive` Boolean su. `false` Questa impostazione crea un bacino d'utenza senza distinzione tra maiuscole e minuscole. Se non specifichi un valore, il `CaseSensitive` viene impostato di default su `true`. I pool di utenti creati nella console Amazon Cognito non fanno distinzione tra maiuscole e minuscole. Per creare un pool di utenti con distinzione tra maiuscole e minuscole, devi utilizzare l'operazione. `CreateUserPool` Prima del 12 febbraio 2020, i bacini d'utenza sono stati impostati di default con distinzione tra maiuscole e minuscole, a prescindere dalla piattaforma.

Nel Sign-inmenu di Console di gestione AWS e nella `UsernameConfiguration` proprietà di DescribeUserPool, puoi rivedere le impostazioni relative alla distinzione tra maiuscole e minuscole per ogni pool di utenti del tuo account.

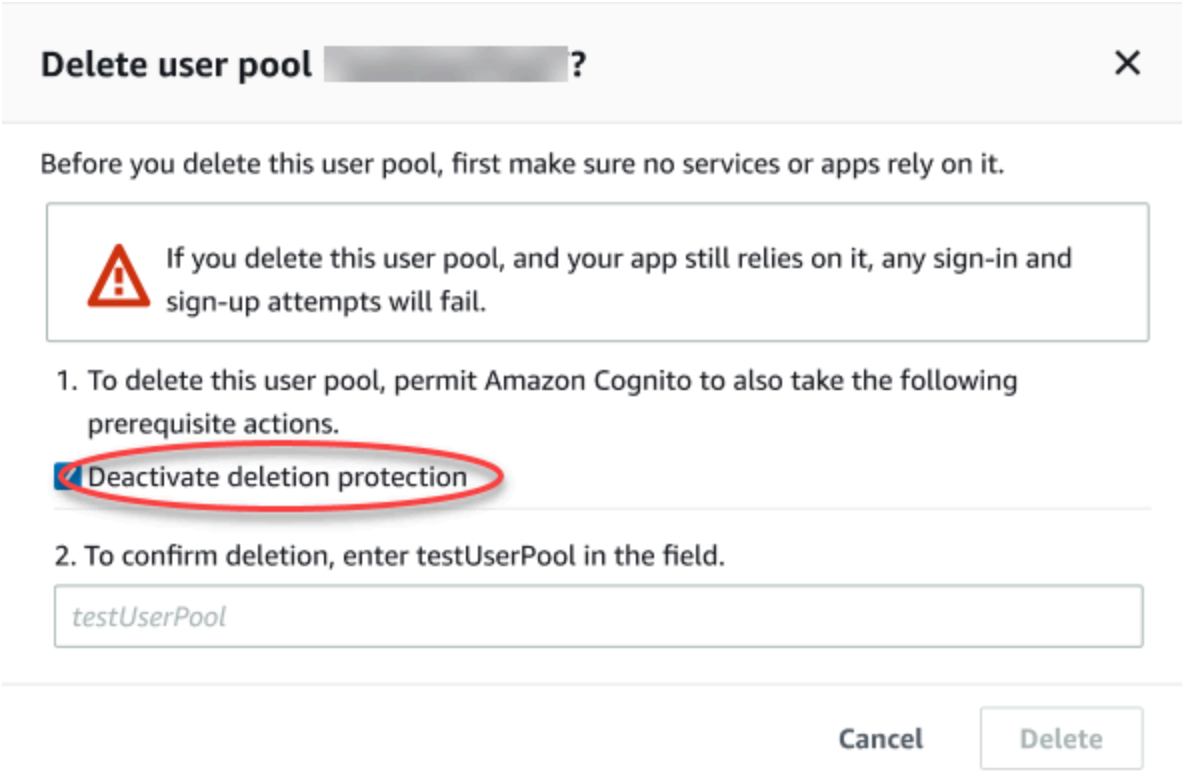
Migrazione a un nuovo bacino d'utenza

A causa di potenziali conflitti tra i profili utente, non puoi modificare un bacino d'utenza Amazon Cognito da sensibille alle maiuscole e alle minuscole a insensibile alle maiuscole e alle minuscole. Invece, migra gli utenti in un nuovo bacino d'utenza. È necessario creare un codice di migrazione per risolvere i conflitti correlati ai casi. Questo codice deve restituire un nuovo utente univoco o rifiutare il tentativo di accesso quando rileva un conflitto. In un nuovo bacino d'utenza senza distinzione tra maiuscole e minuscole, assegna un Trigger Lambda di migrazione utenti. La AWS Lambda funzione può creare utenti nel nuovo pool di utenti senza distinzione tra maiuscole e minuscole. Quando l'utente non riesce ad accedere con il bacino d'utenza senza distinzione tra maiuscole e minuscole, la funzione Lambda trova e duplica l'utente dal bacino d'utenza con distinzione tra maiuscole e minuscole. Puoi anche attivare un trigger ForgotPasswordLambda per la migrazione degli utenti sugli eventi. Amazon Cognito trasmette le informazioni sull'utente e i metadati degli eventi dall'azione di accesso o ripristino della password alla funzione Lambda. Puoi utilizzare i dati dell'evento per gestire conflitti tra nomi utente e indirizzi e-mail quando la funzione crea il nuovo utente nel pool di utenti senza distinzione tra maiuscole e minuscole. Questi conflitti sono tra nomi utente e indirizzi e-mail che sarebbero unici in un pool di utenti con distinzione tra maiuscole e minuscole, ma identici in un pool di utenti senza distinzione tra maiuscole e minuscole.

Per ulteriori informazioni su come utilizzare un trigger Lambda di utenti di migrazione tra pool di utenti Amazon Cognito, consulta Migrazione degli utenti ai pool di utenti di Amazon Cognito nel blog. AWS

Protezione da eliminazione del bacino d'utenza

Per evitare che gli amministratori eliminino accidentalmente il pool di utenti, attiva la protezione dall'eliminazione. Con la protezione dall'eliminazione attiva, puoi eliminare il tuo pool di utenti solo dopo aver confermato di volerlo eliminare. Quando si elimina un pool di utenti inConsole di gestione AWS, è possibile disattivare contemporaneamente la protezione dall'eliminazione. Quando accetti la richiesta di disattivare la protezione dall'eliminazione e confermi l'intenzione di eliminare, come mostrato nell'immagine seguente, Amazon Cognito elimina il tuo pool di utenti.



Se desideri eliminare un pool di utenti con una richiesta API Amazon Cognito, devi prima passare `DeletionProtection` a `Inactive` in una UpdateUserPoolrichiesta. Se non disattivi la protezione dall'eliminazione, Amazon Cognito restituisce un errore `InvalidParameterException`. Dopo aver disattivato la protezione dall'eliminazione, puoi eliminare il pool di utenti in una DeleteUserPoolrichiesta.

Amazon Cognito attiva Deletion protection (Protezione dall'eliminazione) per impostazione predefinita quando crei un nuovo pool di utenti nella Console di gestione AWS. Quando crei un pool di utenti con l'API `CreateUserPool`, la protezione dall'eliminazione è inattiva per impostazione predefinita. Per utilizzare questa funzionalità nei pool di utenti creati con AWS CLI o con un AWS SDK, imposta il `DeletionProtection` parametro su. `True`

Puoi attivare o disattivare lo stato di protezione dall'eliminazione nel contenitore di protezione dall'eliminazione nel menu Impostazioni della console Amazon Cognito.

Per configurare la protezione da eliminazione

1. Passa alla console Amazon Cognito. È possibile che ti vengano richieste le credenziali. AWS
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Impostazioni, vai alla scheda Sicurezza del pool di utenti e individua Deletion Protection. Seleziona Attiva o Disattiva.
5. Conferma la tua scelta nella finestra di dialogo successiva.

Gestione delle risposte agli errori relativi all'esistenza degli utenti

Amazon Cognito supporta la personalizzazione delle risposte di errore restituite dai pool di utenti. Sono disponibili risposte di errore personalizzate per le operazioni di creazione e autenticazione utente, ripristino password e conferma.

Utilizza l'impostazione `PreventUserExistenceErrors` di un client di applicazioni di un bacino d'utenza per abilitare o disabilitare errori relativi all'esistenza degli utenti. Quando crei una nuova app, il client con l'API dei pool di utenti di Amazon Cognito `PreventUserExistenceErrors` è `LEGACY` o disabilitato per impostazione predefinita. Nella console Amazon Cognito, l'opzione Previeni gli errori di esistenza degli utenti, un'impostazione di `ENABLED` for, `PreventUserExistenceErrors` è selezionata per impostazione predefinita. Per aggiornare la `PreventUserExistenceErrors` configurazione, esegui una delle seguenti operazioni:

- Modifica il valore `PreventUserExistenceErrors` tra `ENABLED` e `LEGACY` in una richiesta UpdateUserPoolClientAPI.
- Modifica il client dell'app nella console Amazon Cognito e modifica lo stato di Previeni gli errori di esistenza degli utenti tra selezionato (`ENABLED`) e deselezionato (). `LEGACY`

Quando questa proprietà ha un valore di`LEGACY`, il client dell'app restituisce una risposta di `UserNotFoundException` errore quando un utente tenta di accedere con un nome utente che non esiste nel tuo pool di utenti.

Quando questa proprietà ha un valore di`ENABLED`, il client dell'app non rivela l'inesistenza di un account utente nel pool di utenti con un `UserNotFoundException` errore. Una `PreventUserExistenceErrors` configurazione di `ENABLED` ha i seguenti effetti quando invii una richiesta per un nome utente che non esiste:

- Amazon Cognito risponde con informazioni non specifiche alle richieste API laddove la sua risposta potrebbe altrimenti rivelare l'esistenza di un utente valido.
- Amazon Cognito restituisce una risposta generica di errore di autenticazione alle richieste di password dimenticata e alle richieste di autenticazione con flussi di autenticazione ad eccezione dell'autenticazione basata sulla scelta (), ad esempio o. `USER_AUTH` `USER_SRP_AUTH` `CUSTOM_AUTH` La risposta all'errore indica che il nome utente o la password non sono corretti.
- Amazon Cognito risponde alle richieste di autenticazione basata sulla scelta con una selezione casuale tra i tipi di sfida consentiti per il pool di utenti. Il tuo pool di utenti potrebbe restituire una passkey, una password monouso o una password di verifica.
- Il comportamento delle API di conferma dell'account e recupero della password di Amazon Cognito alterna la restituzione di una risposta che indica che un codice è stato inviato a un supporto di consegna simulato e la restituzione di un errore. `InvalidParameterException`

Le seguenti informazioni descrivono in dettaglio i comportamenti delle operazioni del pool di utenti quando è impostato su. `PreventUserExistenceErrors` `ENABLED`

Operazioni di autenticazione e creazione degli utenti

È possibile configurare le risposte di errore in termini di nome utente-password e autenticazione Secure Remote Password (SRP). Puoi anche personalizzare gli errori restituiti con un'autenticazione personalizzata. Choice-based l'autenticazione non è influenzata dalla `PreventUserExistenceErrors` configurazione.

User-existence dettagli di divulgazione nei flussi di autenticazione

Choice-based authentication

Nel flusso di `USER_AUTH` autenticazione basato sulla scelta, Amazon Cognito pone una sfida rispetto ai principali fattori di autenticazione disponibili, a seconda della configurazione del pool

di utenti e degli attributi degli utenti. Questo flusso di autenticazione può restituire problemi relativi a password, password remota sicura (SRP), WebAuthn (passkey), password monouso (OTP) per SMS o e-mail. Con `PreventUserExistenceErrors` active, Amazon Cognito sfida gli utenti inesistenti a completare una o più delle forme di autenticazione disponibili. Se è `PreventUserExistenceErrors` inattivo, Amazon Cognito restituisce `UserNotFound` un'eccezione.

Autenticazione con nome utente e password

I flussi `ADMIN_USER_PASSWORD_AUTH` di autenticazione e `USER_PASSWORD_AUTH` il `PASSWORD` flusso di `USER_AUTH` ritorno a `NotAuthorizedException` con il messaggio `Incorrect username or password` quando `PreventUserExistenceErrors` è attivo. Quando `PreventUserExistenceErrors` è inattivo, questi flussi ritornano`UserNotFoundException`.

Autenticazione basata su password remota protetta (SRP)

Come procedura consigliata, implementa solo `PreventUserExistenceErrors` con `USER_SRP_AUTH` o con il `PASSWORD_SRP` flusso di pool di `USER_AUTH` utenti interni senza gli attributi di indirizzo e-mail, numero di telefono o alias del nome utente preferito. Gli utenti con attributi di alias potrebbero non essere soggetti alla soppressione dell'esistenza degli utenti nel flusso di autenticazione SRP. Username-password i flussi di autenticazione`ADMIN_USER_PASSWORD_AUTH`, e la `USER_AUTH` `PASSWORD` sfida`USER_PASSWORD_AUTH`, eliminano completamente l'esistenza degli utenti dagli attributi degli alias.

Quando qualcuno tenta l'accesso SRP con un nome utente sconosciuto al client dell'app, Amazon Cognito restituisce una risposta simulata nel primo passaggio, come descritto nella RFC 5054. Amazon Cognito restituisce lo stesso codice e un ID utente interno in formato UUID per la stessa combinazione di nome utente e pool di utenti. Quando si invia una richiesta API `RespondToAuthChallenge` con verifica della password, Amazon Cognito restituisce un errore `NotAuthorizedException` generico quando il nome utente o la password non sono corretti. Per ulteriori informazioni sull'implementazione dell'autenticazione SRP, consulta. Sign-in con password persistenti e payload sicuro

> **Note**
>
> Puoi simulare una risposta generica con l'autenticazione di nome utente e password se utilizzi attributi di alias basati sulla verifica e il nome utente immutabile non è formattato come UUID.

Sfida di autenticazione personalizzata Trigger Lambda

Amazon Cognito richiama la sfida di autenticazione personalizzata che Lambda attiva quando gli utenti tentano di accedere con il flusso di `CUSTOM_AUTH` autenticazione, ma il loro nome utente non viene trovato. L'evento di input include un parametro booleano denominato `UserNotFound` con un valore per ogni utente inesistente. `true` Questo parametro viene visualizzato negli eventi di richiesta che il pool di utenti invia alle funzioni Lambda di creazione, definizione e verifica dell'autenticazione che costituiscono l'architettura di autenticazione personalizzata. Quando esamini questo indicatore nella logica della tua funzione Lambda, puoi simulare sfide di autenticazione personalizzate per un utente che non esiste.

Trigger Lambda di pre-autenticazione

Amazon Cognito richiama il trigger di pre-autenticazione quando gli utenti tentano di accedere ma il loro nome utente non viene trovato. L'evento di input include un `UserNotFound` parametro con un valore pari a `true` per ogni utente inesistente.

L'elenco seguente descrive l'effetto della creazione `PreventUserExistenceErrors` di account utente.

User-existence dettagli di divulgazione nei flussi di creazione degli utenti

SignUp

L'`SignUp`operazione ritorna sempre `UsernameExistsException` quando è già stato utilizzato un nome utente. Per evitare che Amazon Cognito restituisca un errore `UsernameExistsException` per indirizzi e-mail e numeri di telefono durante la registrazione di utenti nell'app, utilizza gli attributi alias basati sulla verifica. Per ulteriori informazioni sugli alias, consultare Personalizzazione degli attributi di accesso.

Per un esempio di come Amazon Cognito può impedire l'uso delle richieste API `SignUp` per scoprire gli utenti del pool di utenti in uso, consulta Prevenzione degli errori `UsernameExistsException` relativi a indirizzi e-mail e numeri di telefono durante la registrazione.

Utenti importati

Se `PreventUserExistenceErrors` è abilitato, durante l'autenticazione degli utenti importati viene restituito un errore `NotAuthorizedException` generico che indica che il nome utente o la password non erano corretti, anziché restituire `PasswordResetRequiredException`. Per ulteriori informazioni, consulta Necessità degli utenti importati di ripristinare le password.

Trigger Lambda di migrazione utenti

Amazon Cognito restituirà una risposta simulata per gli utenti inesistenti quando il trigger Lambda imposta una risposta vuota nel contesto dell'evento originale. Per ulteriori informazioni, consulta Importazione di utenti con un trigger Lambda per la migrazione di utenti.

Prevenzione degli errori **`UsernameExistsException`** relativi a indirizzi e-mail e numeri di telefono durante la registrazione

Nell'esempio seguente viene illustrato come impedire che indirizzi e-mail e numeri di telefono duplicati generino errori `UsernameExistsException` in risposta alle richieste API `SignUp`, durante la configurazione di attributi alias nel pool di utenti. È necessario aver creato il pool di utenti con indirizzo e-mail o numero di telefono come un attributo alias. Per ulteriori informazioni, consulta la sezione Personalizzazione degli attributi di accesso di Attributi del pool di utenti.

1. Jie esegue la registrazione per un nuovo nome utente e fornisce anche l'indirizzo e-mail `jie@example.com`. Amazon Cognito invia un codice all'indirizzo e-mail.

   AWS CLI Comando di esempio

   ```
   aws cognito-idp sign-up --client-id 1234567890abcdef0 --username jie --password
    PASSWORD --user-attributes Name="email",Value="jie@example.com"
   ```

   Esempio di risposta

   ```
   {
       "UserConfirmed": false,
       "UserSub": "<subId>",
       "CodeDeliveryDetails": {
           "AttributeName": "email",
           "Destination": "j****@e****",
           "DeliveryMedium": "EMAIL"
       }
   }
   ```

2. Jie fornisce il codice inviato per confermare di essere il titolare dell'indirizzo email. Questo completa la registrazione come un utente.

   AWS CLI Comando di esempio

```
aws cognito-idp confirm-sign-up --client-id 1234567890abcdef0 --username=jie --
confirmation-code xxxxxx
```

3. Shirley registra un nuovo account utente e fornisce l'indirizzo e-mail `jie@example.com`. Amazon Cognito non restituisce un errore `UsernameExistsException` e invia un codice di conferma all'indirizzo e-mail di Jie.

   AWS CLI Comando di esempio

   ```
   aws cognito-idp sign-up --client-id 1234567890abcdef0 --username shirley --password
    PASSWORD --user-attributes Name="email",Value="jie@example.com"
   ```

   Esempio di risposta

   ```
   {
       "UserConfirmed": false,
       "UserSub": "<new subId>",
       "CodeDeliveryDetails": {
           "AttributeName": "email",
           "Destination": "j****@e****",
           "DeliveryMedium": "EMAIL"
       }
   }
   ```

4. In uno scenario diverso, Shirley è la titolare di `jie@example.com`. Shirley recupera il codice inviato da Amazon Cognito all'indirizzo e-mail di Jie e tenta di confermare l'account.

   AWS CLI Comando di esempio

   ```
   aws cognito-idp confirm-sign-up --client-id 1234567890abcdef0 --username=shirley --
   confirmation-code xxxxxx
   ```

   Esempio di risposta

   ```
   An error occurred (AliasExistsException) when calling the ConfirmSignUp operation: An
    account with the email already exists.
   ```

Amazon Cognito non restituisce un errore alla richiesta `aws cognito-idp sign-up` di Shirley, nonostante `jie@example.com` sia stato assegnato a un utente esistente. Shirley deve dimostrare di essere titolare dell'indirizzo e-mail prima che Amazon Cognito restituisca una risposta di errore. In un pool di utenti con attributi alias, questo comportamento impedisce l'uso dell'API `SignUp` pubblica per verificare se esiste un utente con un determinato indirizzo e-mail o numero di telefono.

Questo comportamento è diverso dalla risposta restituita da Amazon Cognito alla richiesta `SignUp` con un nome utente esistente, come illustrato nell'esempio seguente. Sebbene Shirley apprenda da questa risposta che esiste già un utente con il nome utente `jie`, non viene informata di alcun indirizzo e-mail o numero di telefono associato all'utente.

Comando della CLI di esempio

```
aws cognito-idp sign-up --client-id 1example23456789 --username jie --password PASSWORD
      --user-attributes Name="email",Value="shirley@example.com"
```

Esempio di risposta

```
An error occurred (UsernameExistsException) when calling the SignUp operation: User
 already exists
```

Operazioni di ripristino della password

Amazon Cognito restituisce le seguenti risposte alle operazioni di reimpostazione della password utente quando si evitano errori relativi all’esistenza dell'utente.

ForgotPassword

Se un utente non viene trovato, è disattivato o non dispone di un meccanismo di consegna verificato per recuperare la password, Amazon Cognito restituisce `CodeDeliveryDetails` con un supporto di recapito simulato per un utente. Il supporto di recapito simulato è determinato dal formato del nome utente di input e dalle impostazioni di verifica del pool di utenti.

ConfirmForgotPassword

Amazon Cognito restituisce l'errore `CodeMismatchException` per gli utenti che non esistono o sono disabilitati. Se non viene richiesto un codice quando si utilizza `ForgotPassword`, Amazon Cognito restituisce l'errore `ExpiredCodeException`.

Operazioni di conferma

Amazon Cognito restituisce le seguenti risposte alle operazioni di conferma e verifica quando si evitano errori relativi all’esistenza dell'utente.

ResendConfirmationCode

Amazon Cognito restituisce l'errore `CodeDeliveryDetails` per un utente disabilitato o uno che non esiste. Per un utente esistente, Amazon Cognito invia un codice di conferma all'indirizzo e-mail o al telefono dell'utente.

ConfirmSignUp

`ExpiredCodeException` restituisce se un codice è scaduto. Amazon Cognito restituisce `NotAuthorizedException` quando un utente non è autorizzato. Se il codice non corrisponde a quello previsto dal server, Amazon Cognito restituisce `CodeMismatchException`.

Creazione di applicazioni multi-tenant B2B

Questa sezione descrive i metodi che puoi implementare per separare i tenant tra le risorse di Amazon Cognito all'interno della stessa regione e. Account AWS Puoi anche suddividere i tuoi inquilini in più di una Account AWS regione e assegnare a ciascuno di loro la propria quota. Altri vantaggi della multi-tenancy multiregionale includono il massimo livello di isolamento possibile, il tempo di transito di rete più breve per gli utenti distribuiti a livello globale e l'aderenza ai modelli di distribuzione esistenti nell'organizzazione.

I pool di utenti di Amazon Cognito operano con applicazioni multi-tenant che generano un volume di richieste che deve rimanere entro le quote di Amazon Cognito. Per aumentare questa capacità man mano che la tua base di clienti cresce, puoi acquistare quote di capacità aggiuntive.

> **Note**
>
> Le quote di Amazon Cognito vengono applicate singolarmente. Account AWS Regione AWS Queste quote vengono condivise tra tutti i tenant dell'applicazione. Controlla le quote del servizio Amazon Cognito e assicurati che la quota soddisfi il volume previsto e il numero previsto di tenant nella tua applicazione.

Single-Region la multi-tenancy può avere vantaggi anche per i tuoi clienti e amministratori.

L'elenco seguente illustra alcuni dei vantaggi della multi-tenancy con risorse condivise.

Vantaggi della multi-tenancy

Elenco utenti comune

Multi-tenancy supporta modelli in cui i clienti dispongono di account in più di un'applicazione. È possibile collegare le identità di provider di terze parti in un unico profilo di pool di utenti coerente. Nei casi in cui i profili utente sono unici per il relativo tenant, qualsiasi strategia multi-tenancy con un unico pool di utenti prevede un unico punto di accesso per l'amministrazione degli utenti.

Sicurezza comune

In un pool di utenti condiviso, puoi creare un unico standard di sicurezza e applicare la stessa protezione dalle minacce, l'autenticazione a più fattori (MFA) AWS WAFe gli stessi standard

a tutti i tenant. Poiché un ACL AWS WAF Web deve coincidere con la risorsa a cui Regione AWS lo si associa, la multi-tenancy offre l'accesso condiviso a una risorsa complessa. Se desideri mantenere una configurazione di sicurezza coerente in applicazioni Amazon Cognito multiregionali, devi applicare standard operativi che replichino la configurazione tra le risorse.

Personalizzazione comune

Puoi personalizzare i pool di utenti e i pool di identità con AWS Lambda. La configurazione dei trigger Lambda nei pool di utenti e degli eventi di Amazon Cognito nei pool di identità può diventare complessa. Le funzioni Lambda devono trovarsi nello stesso pool Regione AWS di utenti o pool di identità. Le funzioni Lambda condivise possono applicare gli standard per i flussi di autenticazione personalizzati, la migrazione degli utenti, la generazione di token e altre funzioni all'interno di una regione.

Messaggistica comune

Amazon Simple Notification Service (Amazon SNS) richiede una configurazione aggiuntiva in una regione prima di poter inviare messaggi SMS ai tuoi utenti. Puoi inviare messaggi e-mail con identità e domini verificati di Amazon Simple Email Service (Amazon SES) contenuti in una regione.

Con la multi-tenancy, puoi condividere questo sovraccarico di configurazione e manutenzione tra tutti i tuoi tenant. Poiché Amazon SNS e Amazon SES non sono tutti disponibili Regioni AWS, la suddivisione delle risorse tra le regioni richiede ulteriori considerazioni.

Quando utilizzi provider di messaggistica personalizzati, ottieni la personalizzazione comune di una singola funzione Lambda per gestire la consegna dei messaggi.

L'accesso gestito imposta un cookie di sessione nel browser in modo che riconosca un utente che si è già autenticato. Quando autentichi gli utenti locali in un pool di utenti, il relativo cookie di sessione li autentica per tutti i client dell'app nello stesso pool di utenti. Un utente locale esiste esclusivamente nella directory del pool di utenti senza federazione tramite un IdP esterno. Il cookie di sessione è valido per un'ora. Non è possibile modificare la durata del cookie di sessione.

Esistono due modi per impedire l'accesso tra client di app con un cookie di sessione dell'interfaccia utente ospitato.

- Separa i tuoi utenti in pool di utenti per tenant.
- Sostituisci l'accesso all'interfaccia utente ospitata con l'accesso all'API dei pool di utenti di Amazon Cognito.

Argomenti

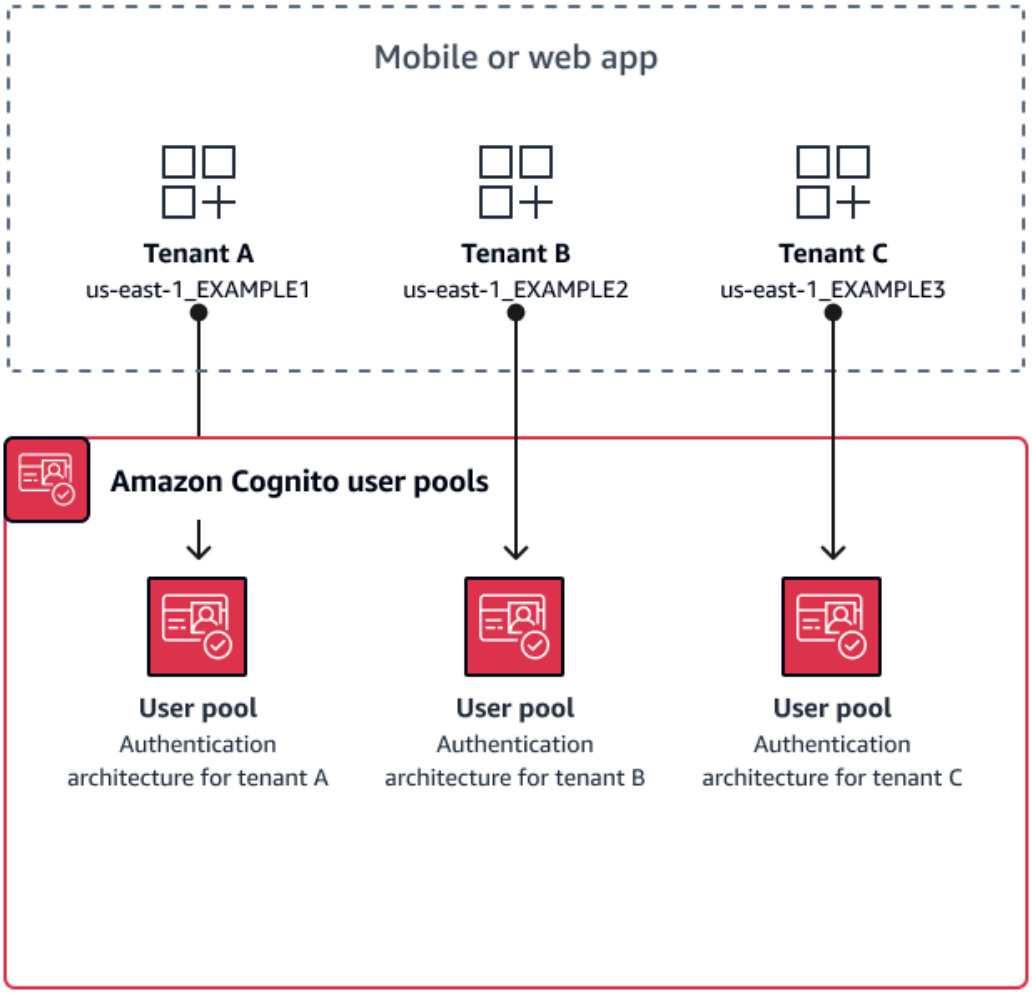
- User-pool migliori pratiche di multi-tenancy
- App-client migliori pratiche di multi-tenancy
- Migliori pratiche di multi-tenancy per gruppi di utenti
- Custom-attribute migliori pratiche di multi-tenancy
- Best practice di multi-tenancy con ambito personalizzato
- Multi-tenancy raccomandazioni sulla sicurezza

User-pool migliori pratiche di multi-tenancy

Crea un pool di utenti per ogni tenant nella tua app. Questo approccio fornisce il massimo isolamento per ogni tenant. È possibile implementare configurazioni diverse per ciascuno di essi. L'isolamento dei tenant in base al bacino d'utenza fornisce flessibilità nella mappatura da utente a tenant. È possibile creare più profili per lo stesso utente. Tuttavia, ogni utente deve registrarsi individualmente per ogni tenant a cui può accedere.

Utilizzando questo approccio, puoi configurare un'interfaccia utente ospitata per ogni tenant in modo indipendente e reindirizzare gli utenti all'istanza della tua applicazione specifica per il tenant. Puoi anche utilizzare questo approccio per l'integrazione con servizi di back-end come Amazon API Gateway.

Il diagramma seguente mostra ogni tenant con un pool di utenti dedicato.



Quando implementare la modalità multi-tenancy con pool di utenti

Quando l'isolamento e la personalizzazione sono le tue preoccupazioni principali. La relazione tra utenti e tenant potrebbe essere complessa in un'architettura con più pool di utenti. Considerate un esempio in cui avete due inquilini che frequentano un istituto di istruzione. Lo stesso utente potrebbe essere uno studente con accesso limitato in un'app e un insegnante con un livello elevato di autorizzazioni in un'altra. Potresti richiedere la MFA in un'app ma non in un'altra o avere una politica di password diversa. Poiché gli utenti locali possono accedere a più client di app in pool di utenti con accesso gestito, la modalità multi-tenancy con pool di utenti è ideale anche quando desideri che più di uno dei tuoi tenant acceda con l'accesso gestito.

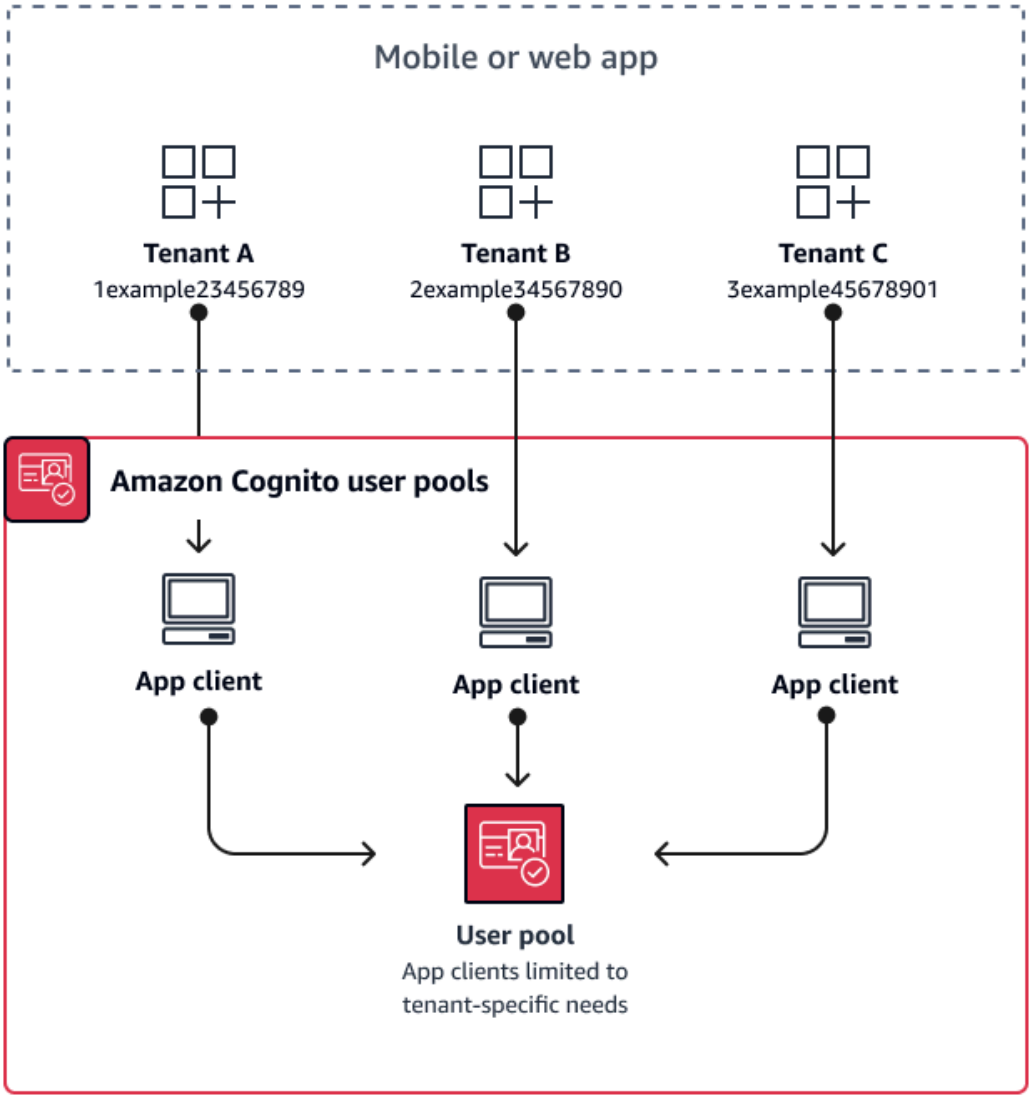
Livello di impegno

Lo sforzo di sviluppo e di funzionamento per utilizzare questo approccio è elevato. Per garantire risultati coerenti e prevedibili per la tua famiglia di app, devi integrare le risorse di Amazon Cognito con i tuoi strumenti di automazione e mantenere le tue linee di base man mano che l'architettura di autenticazione diventa più complessa. Quando vuoi creare un unico punto di partenza per le tue app, devi creare gli elementi dell'interfaccia utente (UI) per acquisire la decisione iniziale che indirizza gli utenti alla risorsa corretta.

App-client migliori pratiche di multi-tenancy

Crea un app client per ogni inquilino della tua app. Con app-client multi-tenancy, puoi assegnare qualsiasi utente a client di app collegati al tenant e mantenere un unico profilo utente. Poiché puoi assegnare uno o tutti gli identity provider (IdPs) del tuo pool di utenti a un app client, un client di app tenant può consentire l'accesso con un IdP specifico del tenant. Quando gli utenti esistono in più tenant, puoi collegare i loro profili a più tenant per un'esperienza utente coerente. IdPs

Il diagramma seguente mostra ogni tenant con un client di app dedicato in un pool di utenti condiviso.



Quando implementare la modalità multi-tenancy tra app e client

Quando puoi scegliere una configurazione universale per le impostazioni a livello di pool di utenti, come i trigger Lambda, la politica delle password e il contenuto e i metodi di consegna di e-mail e SMS. Poiché gli utenti di un pool di utenti condiviso possono accedere a qualsiasi client di app, il multi-tenancy app-client è ideale per l'accesso con l'API dei pool di utenti di Amazon Cognito IdPs o con l'API dei pool di utenti di Amazon Cognito. App-client La multi-tenancy è ideale anche per ambienti one-to-many in cui si desidera consentire agli utenti di passare da un'applicazione all'altra.

Livello di impegno

App-client la locazione multipla richiede uno sforzo moderato. Una delle principali sfide del multi-tenancy tra app e client è la possibilità per i tenant di presentare un cookie dell'interfaccia utente ospitato e passare da un'app all'altra. In un'architettura multi-tenancy tra app e client, evita l'accesso all'interfaccia utente ospitata dove è necessario l'isolamento. Puoi distribuire la tua app per dispositivi mobili o i link alla tua app web utilizzando la logica del client dell'app integrata, oppure puoi creare elementi iniziali dell'interfaccia utente che determinano la locazione degli utenti. Il livello di impegno è inferiore perché non è necessario standardizzare e mantenere la configurazione su più pool di utenti e pool di identità.

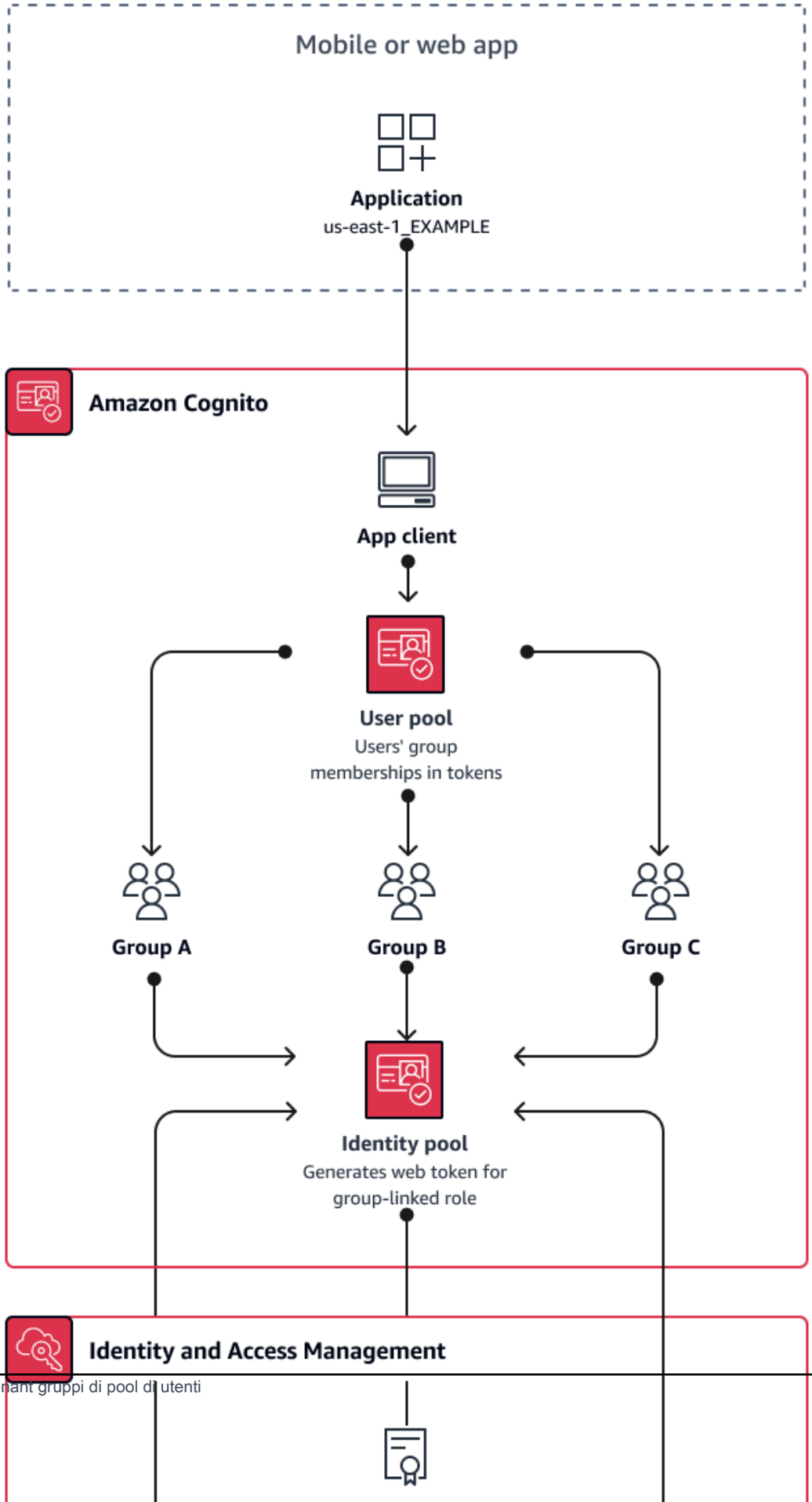
Migliori pratiche di multi-tenancy per gruppi di utenti

Group-based la multi-tenancy funziona al meglio quando la tua architettura richiede pool di utenti Amazon Cognito con pool di identità.

L'ID del pool di utenti e i token di accesso contengono un claim. `cognito:groups` Inoltre, i token ID contengono `cognito:roles` e `cognito:preferred_role` rivendicazioni. Quando il risultato principale dell'autenticazione nella tua app sono AWS credenziali temporanee provenienti da un pool di identità, l'appartenenza ai gruppi degli utenti può determinare il ruolo IAM e le autorizzazioni che ricevono.

Ad esempio, prendiamo in considerazione tre tenant, ciascuno dei quali archivia gli asset applicativi nel proprio bucket Amazon S3. Assegna gli utenti di ogni tenant a un gruppo associato, configura un ruolo preferito per il gruppo e concedi a quel ruolo l'accesso in lettura al relativo bucket.

Il diagramma seguente mostra i tenant che condividono un client di app e un pool di utenti, con gruppi dedicati nel pool di utenti che determinano la loro idoneità per un ruolo IAM.


Mobile or web app
Application
us-east-1_EXAMPLE
Amazon Cognito
App client
User pool
Users' group memberships in tokens
Group A
Group B
Group C
Identity pool
Generates web token for group-linked role
Identity and Access Management

Quando implementare la multi-tenancy di gruppo

Quando l'accesso alle AWS risorse è la tua preoccupazione principale. I gruppi nei pool di utenti di Amazon Cognito sono un meccanismo per il controllo degli accessi basato sui ruoli (RBAC). Puoi configurare molti gruppi in un pool di utenti e prendere decisioni RBAC complesse con priorità di gruppo. I pool di identità possono assegnare credenziali per il ruolo con la priorità più alta, per qualsiasi ruolo nel claim di gruppo o per altre attestazioni nei token di un utente.

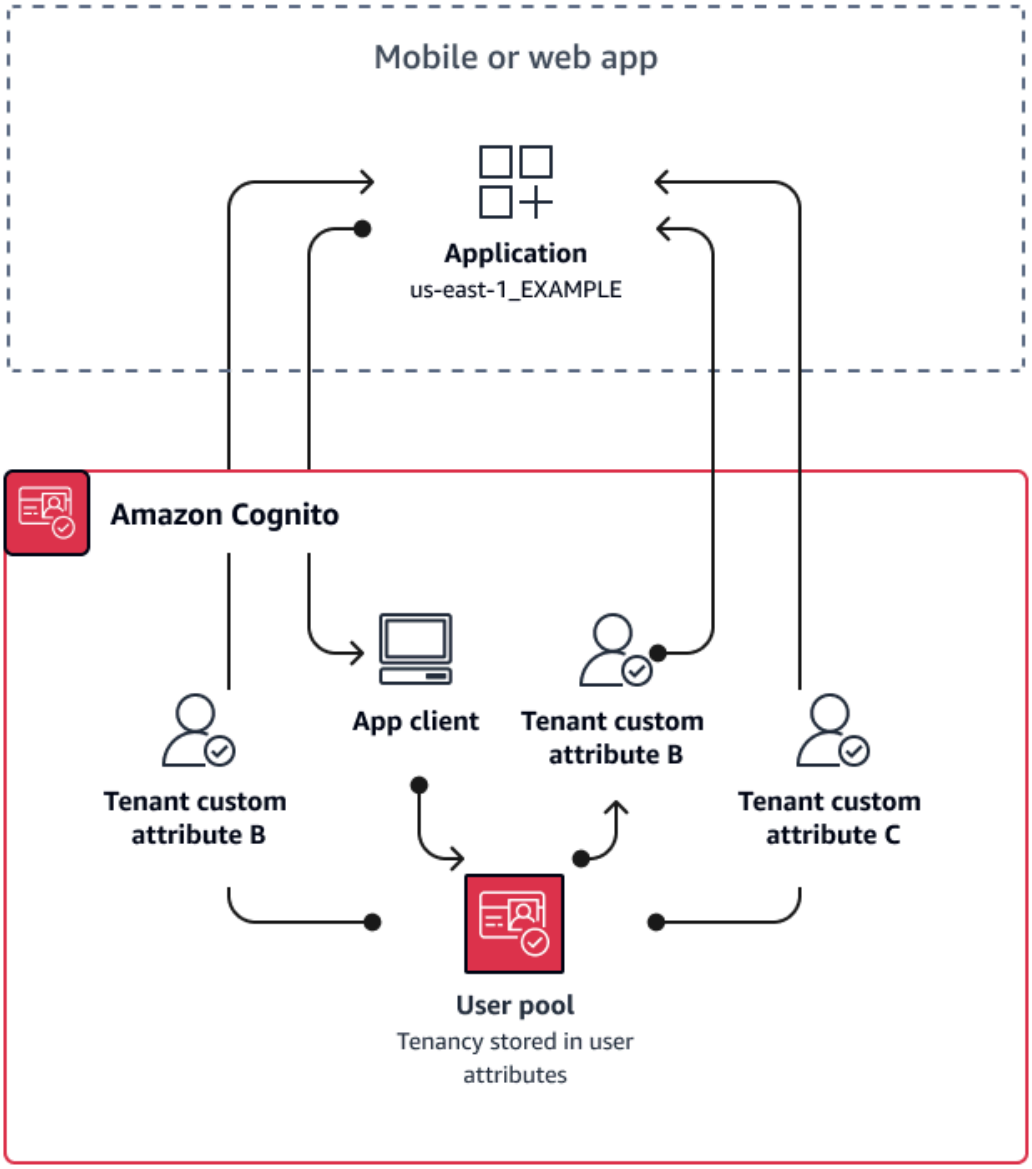
Livello di impegno

Il livello di impegno per mantenere la multi-tenancy con la sola appartenenza al gruppo è basso. Tuttavia, per ampliare il ruolo dei gruppi di pool di utenti oltre alla capacità integrata di selezione dei ruoli IAM, è necessario creare una logica applicativa che elabori l'appartenenza ai gruppi nei token degli utenti e determinare cosa fare nel client. Puoi integrare Amazon Verified Permissions con le tue app per prendere decisioni di autorizzazione lato client. Gli identificatori di gruppo non vengono attualmente elaborati nelle operazioni dell'IsAuthorizedWithTokenAPI Verified Permissions, ma puoi sviluppare codice personalizzato che analizzi il contenuto dei token, comprese le richieste di appartenenza ai gruppi.

Custom-attribute migliori pratiche di multi-tenancy

Amazon Cognito supporta attributi personalizzati con nomi a tua scelta. Uno scenario in cui gli attributi personalizzati sono utili è quando distinguono la locazione degli utenti in un pool di utenti condiviso. Quando assegni agli utenti un valore per un attributo come`custom:tenantID`, ad esempio, la tua app può assegnare di conseguenza l'accesso alle risorse specifiche del tenant. Un attributo personalizzato che definisce un ID tenant deve essere immutabile o di sola lettura per il client dell'app.

Il diagramma seguente mostra i tenant che condividono un client di app e un pool di utenti, con attributi personalizzati nel pool di utenti che indica il tenant a cui appartengono.



Quando gli attributi personalizzati determinano la locazione, è possibile distribuire una singola applicazione o un URL di accesso. Dopo che l'utente ha effettuato l'accesso, l'app può elaborare il `custom:tenantID` reclamo per determinare quali risorse caricare, il marchio da applicare e le funzionalità da visualizzare. Per decisioni avanzate sul controllo degli accessi in base agli attributi degli utenti, configura il tuo pool di utenti come provider di identità in Amazon Verified Permissions e genera decisioni di accesso dai contenuti di ID o token di accesso.

Quando implementare la multi-tenancy con attributi personalizzati

Quando la locazione è a livello di superficie. Un attributo tenant può contribuire ai risultati di branding e layout. Quando si desidera ottenere un isolamento significativo tra i tenant, gli attributi personalizzati non sono la scelta migliore. Qualsiasi differenza tra i tenant che devono essere configurati a livello di pool di utenti o app-client, come MFA o il branding dell'interfaccia utente ospitata, richiede la creazione di distinzioni tra i tenant in un modo che gli attributi personalizzati non offrono. Con i pool di identità, puoi persino scegliere il ruolo IAM tra i tuoi utenti dall'indicazione degli attributi personalizzati contenuta nel loro token ID.

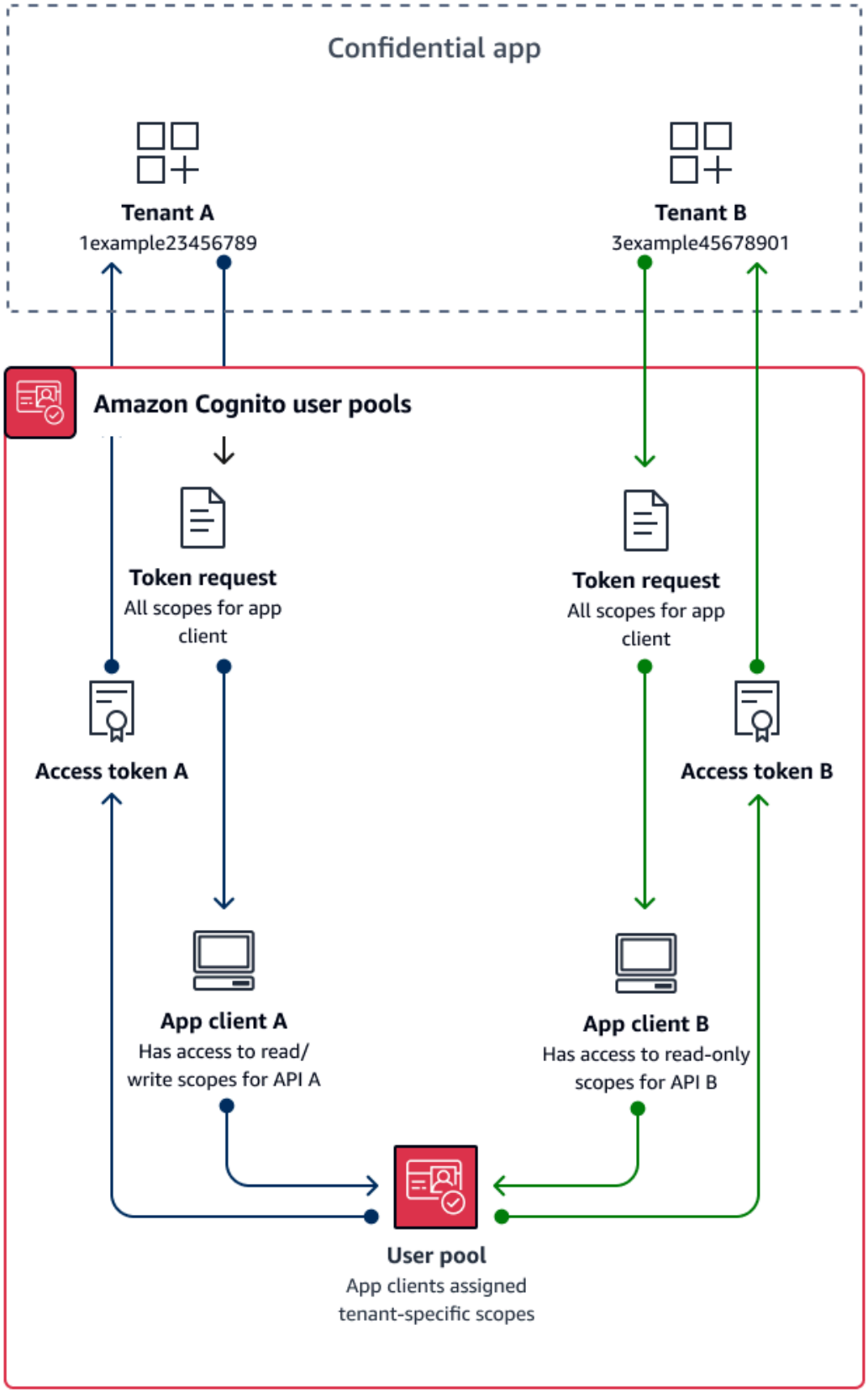
Livello di impegno

Poiché la multi-tenancy con attributi personalizzati trasferisce all'app l'obbligo di prendere decisioni di autorizzazione basate sul tenant, il livello di impegno tende ad essere elevato. Se sei già esperto in una configurazione client che analizza le dichiarazioni OIDC o in Amazon Verified Permissions, questo approccio potrebbe richiedere il minimo sforzo.

Best practice di multi-tenancy con ambito personalizzato

Amazon Cognito supporta ambiti OAuth 2.0 personalizzati per server di risorse. Puoi implementare la multi-tenancy di app client in pool di utenti per modelli di autorizzazione da macchina a macchina (M2M) con ambiti personalizzati. Scope-based la multi-tenancy riduce lo sforzo richiesto per implementare la multi-tenancy M2M definendo l'accesso nel client dell'app o nella configurazione dell'applicazione.

Il diagramma seguente illustra un'opzione per la multi-tenancy con ambito personalizzato. Mostra ogni tenant con un client di app dedicato che ha accesso agli ambiti pertinenti in un pool di utenti.


Confidential app
Tenant A
1example23456789
Tenant B
3example45678901
Amazon Cognito user pools
Token request
All scopes for app client
Token request
All scopes for app client
Access token A
Access token B
App client A
Has access to read/ write scopes for API A
App client B
Has access to read-only scopes for API B
User pool
App clients assigned tenant-specific scopes

Quando implementare la multi-tenancy con ambito personalizzato

Quando si utilizza l'autorizzazione M2M con credenziali client in un client riservato. Come best practice, create server di risorse esclusivi per un client di app. La multi-tenancy con ambito personalizzato può dipendere dalla richiesta o dal client.

Request-dependent

Implementa la logica dell'applicazione per richiedere solo gli ambiti che soddisfano i requisiti del tuo tenant. Ad esempio, un client di app potrebbe essere in grado di fornire l'accesso in lettura e scrittura all'API A e all'API B, ma l'applicazione tenant A richiede solo l'ambito di lettura per l'API A e l'ambito che indica la tenancy. Questo modello consente combinazioni più complesse di ambiti condivisi tra tenant.

Client-dependent

Richiedi tutti gli ambiti assegnati a un client dell'app nelle tue richieste di autorizzazione. Per fare ciò, ometti il parametro `scope` request nella tua richiesta in. Endpoint Token Se ottenete token di accesso M2M con l'operazione GetClientTokenAPI, omettete il `Scopes` parametro per lo stesso risultato. Questo modello consente ai client delle app di memorizzare gli indicatori di accesso che desideri aggiungere agli ambiti personalizzati.

In entrambi i casi, le applicazioni ricevono token di accesso con ambiti che indicano i loro privilegi per le fonti di dati da cui dipendono. Gli ambiti possono anche presentare altre informazioni all'applicazione:

- Designare la locazione
- Contribuisci alla registrazione delle richieste
- Indicate le API che l'applicazione è autorizzata a interrogare
- Informa i controlli iniziali per i clienti attivi.

Livello di impegno

Custom-scope la multi-tenancy richiede un livello di impegno variabile rispetto alla scala dell'applicazione. È necessario elaborare una logica applicativa che consenta alle applicazioni di analizzare i token di accesso ed effettuare le richieste API appropriate.

Ad esempio, l'ambito di un server di risorse è disponibile nel formato. `[resource server identifier]/[name]` È improbabile che l'identificatore del server di risorse sia rilevante per la

decisione di autorizzazione presa dall'ambito del tenant, poiché richiede un'analisi coerente del nome dell'ambito.

Risorsa di esempio

Il AWS CloudFormation modello seguente crea un pool di utenti per la multi-tenancy con ambito personalizzato con un server di risorse e un client di app.

```
AWSTemplateFormatVersion: "2010-09-09"
Description: A sample template illustrating scope-based multi-tenancy
Resources:
  MyUserPool:
    Type: "AWS::Cognito::UserPool"
  MyUserPoolDomain:
    Type: AWS::Cognito::UserPoolDomain
    Properties:
      UserPoolId: !Ref MyUserPool
      # Note that the value for "Domain" must be unique across all of AWS.
      # In production, you may want to consider using a custom domain.
      # See: https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-
pools-add-custom-domain.html#cognito-user-pools-add-custom-domain-adding
      Domain: !Sub "example-userpool-domain-${AWS::AccountId}"
  MyUserPoolResourceServer:
    Type: "AWS::Cognito::UserPoolResourceServer"
    Properties:
      Identifier: resource1
      Name: resource1
      Scopes:
        - ScopeDescription: Read-only access
          ScopeName: readScope
      UserPoolId: !Ref MyUserPool
  MyUserPoolTenantBatch1ResourceServer:
    Type: "AWS::Cognito::UserPoolResourceServer"
    Properties:
      Identifier: TenantBatch1
      Name: TenantBatch1
      Scopes:
        - ScopeDescription: tenant1 identifier
          ScopeName: tenant1
        - ScopeDescription: tenant2 identifier
          ScopeName: tenant2
      UserPoolId: !Ref MyUserPool
  MyUserPoolClientTenant1:
```

```
    Type: "AWS::Cognito::UserPoolClient"
    Properties:
      AllowedOAuthFlows:
        - client_credentials
      AllowedOAuthFlowsUserPoolClient: true
      AllowedOAuthScopes:
        - !Sub "${MyUserPoolTenantBatch1ResourceServer}/tenant1"
        - !Sub "${MyUserPoolResourceServer}/readScope"
      GenerateSecret: true
      UserPoolId: !Ref MyUserPool
Outputs:
  UserPoolClientId:
    Description: User pool client ID
    Value: !Ref MyUserPoolClientTenant1
  UserPoolDomain:
    Description: User pool domain
    Value: !Sub "https://${MyUserPoolDomain}.auth.${AWS::Region}.amazoncognito.com"
```

Multi-tenancy raccomandazioni sulla sicurezza

Per rendere la tua applicazione più sicura, ti consigliamo di seguire questi suggerimenti:

- Convalida la locazione nella tua app con Amazon Verified Permissions. Crea politiche che esaminino l'autorizzazione del pool di utenti, del client dell'app, del gruppo o degli attributi personalizzati prima di consentire la richiesta di un utente nella tua applicazione. AWS ha creato fonti di identità Verified Permissions pensando ai pool di utenti di Amazon Cognito. Verified Permissions offre ulteriori linee guida per la gestione multi-tenancy.
- Usa solo un indirizzo e-mail verificato per autorizzare l'accesso degli utenti a un tenant in base alla corrispondenza del dominio. Non fidarti degli indirizzi e-mail e dei numeri di telefono a meno che la tua app non li verifichi o che l'IdP esterno fornisca una prova di verifica. Per maggiori dettagli sull'impostazione di queste autorizzazioni, consulta la sezione Autorizzazioni e ambiti degli attributi.
- Utilizza attributi personalizzati immutabili o di sola lettura per gli attributi del profilo utente che identificano i tenant. Puoi impostare il valore degli attributi immutabili solo quando crei un utente o quando un utente si iscrive al tuo pool di utenti. Fornisci inoltre ai client di app l'accesso in sola lettura agli attributi.
- Utilizza la mappatura 1:1 tra l'IdP esterno di un tenant e il client dell'applicazione per impedire l'accesso non autorizzato tra tenant. Un utente autenticato da un IdP esterno e dotato di un cookie di sessione Amazon Cognito valido, può accedere ad altre App tenant che considerano lo stesso IdP come affidabile.

- Quando implementi la logica di autorizzazione e corrispondenza tenant nell'applicazione, imposta delle limitazioni affinché le autorizzazioni di accesso degli utenti ai tenant non possano essere modificate dagli utenti stessi. Inoltre, se per la federazione viene utilizzato un IdP esterno, limita gli amministratori del provider di identità del tenant in modo che non possano modificare l'accesso degli utenti.

Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili

L'implementazione di Amazon Cognito è una combinazione dei nostri strumenti Console di gestione AWS amministrativi AWS SDK e delle librerie SDK nelle applicazioni. La console Amazon Cognito è l'interfaccia visiva per la configurazione e la gestione dei pool di utenti e dei pool di identità di Amazon Cognito.

L'integrazione con il minimo sforzo che puoi creare con i pool di utenti di Amazon Cognito è l'accesso gestito. Managed login è un'applicazione di accesso basata sul Web pronta all'uso per il test e la distribuzione rapidi dei pool di utenti di Amazon Cognito. L'autenticazione del pool di utenti con accesso gestito richiede librerie OpenID Connect (OIDC) che indirizzano gli utenti alle pagine di accesso ospitate. In questa serie di endpoint Web interattivi per l'utente e di reindirizzamento, Amazon Cognito gestisce il flusso di autenticazione, tra cui l'accesso di terze parti, l'autenticazione a più fattori (MFA) e la scelta di un flusso di autenticazione. L'applicazione deve solo elaborare il risultato dell'autenticazione restituito da Amazon Cognito nella risposta.

Puoi anche aggiungere un AWS SDK alla tua applicazione, creare interfacce di autenticazione personalizzate e richiamare operazioni API per l'autenticazione e l'autorizzazione dei tuoi utenti. AWS Amplifyè un programma Servizio AWS per la creazione di applicazioni complete, con l'autenticazione Amazon Cognito nel back-end.

Ad esempio, l'app potrebbe richiamare l'accesso gestito per l'accesso dell'utente, quindi richiamare l'endpoint del token dal codice dell'app per scambiare il codice di autorizzazione dell'utente con token. Quindi, l'app deve interpretare e archiviare i token dell'utente e presentarli nel contesto appropriato per l'autenticazione e l'autorizzazione. Amplify aggiunge strumenti di integrazione guidata con funzioni integrate per tali processi.

Puoi anche creare le risorse Amazon Cognito interamente nel codice. I pool di identità non hanno le stesse opzioni di autenticazione gestita dei pool di utenti: per accedere alle AWS credenziali nelle applicazioni, implementa le operazioni dei pool di identità nei moduli SDK importati. Per iniziare a usare il tuo codice applicativo personalizzato, consulta gli esempi di codice di Amazon Cognito per gli SDK.AWS Per l'integrazione con Amazon Cognito come provider di identità OpenID Connect, usa Strumenti per sviluppatori OpenID Connect.

Prima di utilizzare l'autenticazione e l'autorizzazione Amazon Cognito, scegli una piattaforma app e prepara il tuo codice per l'integrazione con il servizio. Per le piattaforme disponibili per gli SDK,

consulta AWS . Autenticazione con AWS SDK AWS CLI È un SDK a riga di comando per Amazon Cognito e altri Servizi AWS ed è un ottimo punto di partenza per iniziare a familiarizzare con le operazioni dell'API Amazon Cognito e la loro sintassi.

> **Note**
>
> Puoi configurare alcuni componenti di Amazon Cognito solo con l'API. Ad esempio, puoi impostare un trigger Lambda personalizzato per SMS o mittente e-mail del pool di utenti solo con una richiesta che aggiorna `LambdaConfig` la proprietà UserPooldella classe in CreateUserPooluna richiesta UpdateUserPoolo API.

L'API dei pool di utenti di Amazon Cognito condivide lo spazio dei nomi con diverse classi di operazioni API. Una classe configura i pool di utenti e i relativi processi, i provider di identità e gli utenti. Un'altra include operazioni non autenticate che consentono agli utenti di un client pubblico di accedere, disconnettersi e gestire i propri profili. L'ultima classe di operazioni API esegue operazioni utente autorizzate con le proprie AWS credenziali in un client riservato lato server. È necessario conoscere l'architettura dell'app desiderata prima di iniziare a implementare il codice dell'app. Per ulteriori informazioni, consulta Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite.

Argomenti

- Autenticazione con AWS Amplify
- Autenticazione con AWS SDK
- Come funziona l'autenticazione con Amazon Cognito
- Utilizzo di questo servizio con un AWS SDK
- Autorizzazione con Amazon Verified Permissions

Autenticazione con AWS Amplify

AWS Amplify è una soluzione completa per la creazione di applicazioni web e mobili. Con Amplify, puoi connetterti alle risorse esistenti con le librerie Amplify oppure puoi creare e configurare nuove risorse con l'interfaccia a riga di comando (CLI) di Amplify. Amplify dispone anche di componenti dell'interfaccia utente collegati come Autenticatore per configurare e personalizzare l'esperienza di accesso e registrazione nella tua app.

Per utilizzare le funzionalità di autenticazione di Amplify nella tua app front-end, consulta la seguente documentazione in base alla piattaforma.

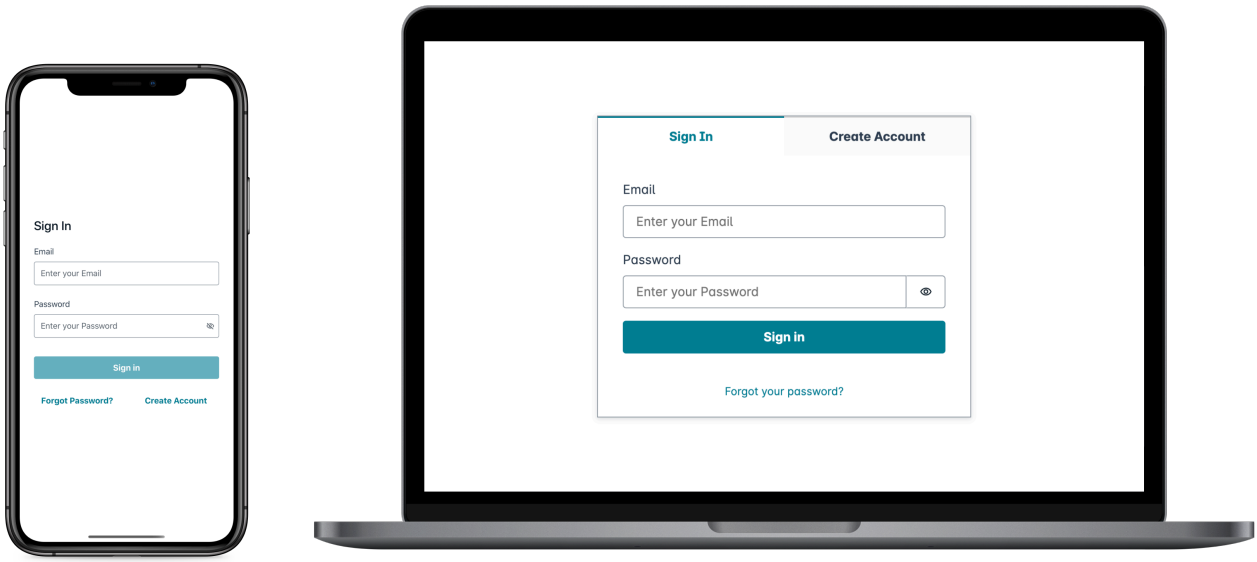
- Autenticazione Amplify per React
- Autenticazione Amplify per React Native
- Autenticazione Amplify per Swift (iOS)
- Autenticazione Amplify per Android
- Amplifica l'autenticazione per Flutter

Le librerie Amplify sono open source e sono disponibili su. GitHub Per ulteriori informazioni su come Amplify Auth implementa l'autenticazione Amazon Cognito, visita le seguenti librerie.

- amplifica-js
- amplify-swift
- amplify-flutter
- amplify-android

Creazione di un'interfaccia utente con Amplify

Accesso gestito dal pool di utentipuò soddisfare le esigenze essenziali di un front-end di autenticazione per un'app web o mobile. Per personalizzare l'interfaccia utente (UI) oltre ai parametri consentiti dall'accesso gestito, crea un'applicazione personalizzata. Amplify UI è una raccolta personalizzabile di componenti front-end in un'ampia gamma di lingue.



Per iniziare a utilizzate il tuo componente di autenticazione personalizzato, consulta la seguente documentazione per il componente Authenticator.

- Authenticator per Android
- Authenticator per Angular
- Authenticator per Angular
- Authenticator per React
- Authenticator per React Native
- Authenticator per Swift
- Authenticator per Vue

Autenticazione con AWS SDK

Per utilizzare un backend sicuro per creare il tuo microservizio di identità che interagisce con Amazon Cognito, connettiti ai pool di utenti di Amazon Cognito e all'API dei pool di identità di Amazon Cognito con AWS un SDK nella lingua che preferisci.

Per informazioni dettagliate su tutte le operazioni API, consulta la Documentazione di riferimento dell'API di Amazon Cognito e la Documentazione di riferimento dell'API Amazon Cognito. Questi

documenti contengono sezioni vedi anche, che includono risorse per l'utilizzo di una varietà di SDK nelle piattaforme supportate.

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK for Go
- AWS SDK per Java V2
- AWS SDK per JavaScript
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

Come funziona l'autenticazione con Amazon Cognito

Quando il cliente accede a un pool di utenti Amazon Cognito, l'applicazione riceve token web JSON (JWT).

Quando il cliente accede a un pool di identità, con un token del pool di utenti o con un altro provider, l'applicazione riceve credenziali temporanee. AWS

Con l'accesso al pool di utenti, puoi implementare l'autenticazione e l'autorizzazione interamente con un AWS SDK. Se non desideri creare componenti di interfaccia utente (UI) personalizzati, puoi richiamare un'interfaccia utente Web predefinita (accesso gestito) o la pagina di accesso per il tuo provider di identità (IdP) di terze parti.

Questo argomento offre una panoramica di alcuni modi in cui l'applicazione può interagire con Amazon Cognito per autenticarsi con token ID, autorizzare con token di accesso e accedere con credenziali di pool di identità. Servizi AWS

Argomenti

- Autenticazione del pool di utenti con accesso gestito
- Autenticazione e autorizzazione dell'API del pool di utenti con un AWS SDK
- Autenticazione del pool di utenti con un provider di identità di terze parti
- Autenticazione del pool di identità

Autenticazione del pool di utenti con accesso gestito

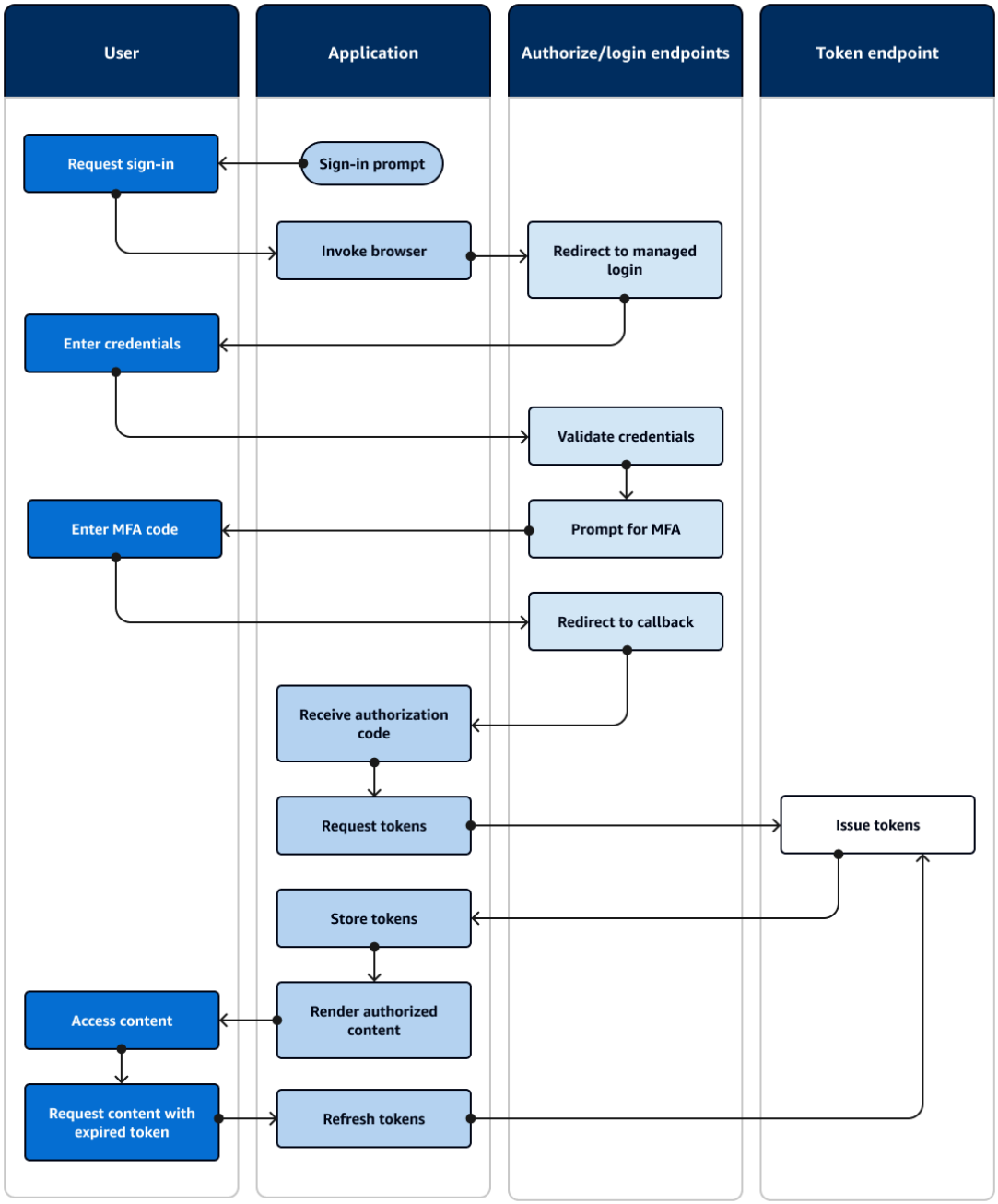
L'accesso gestito è un sito Web collegato al pool di utenti e al client dell'app. Può eseguire operazioni di accesso, registrazione e reimpostazione della password per i tuoi utenti. Un'applicazione con un componente di accesso gestito per l'autenticazione può richiedere meno sforzi da parte degli sviluppatori per l'implementazione. Un'applicazione può ignorare i componenti dell'interfaccia utente per l'autenticazione e richiamare pagine Web di accesso gestito nel browser dell'utente.

Le applicazioni raccolgono i JWT degli utenti con una posizione di reindirizzamento Web o app. Le applicazioni che implementano l'accesso gestito possono connettersi ai pool di utenti per l'autenticazione come se fossero un IdP OpenID Connect (OIDC).

L'accesso gestito si adatta al modello in cui le applicazioni richiedono i servizi di autenticazione di un server di autorizzazione OIDC, ma non richiedono immediatamente funzionalità come l'autenticazione personalizzata, l'integrazione dei pool di identità o il self-service degli attributi utente. Se desideri utilizzare alcune di queste opzioni avanzate, puoi implementarle con un componente di pool di utenti per un SDK.

L'accesso gestito e i modelli di autenticazione IdP di terze parti, che si basano principalmente sull'implementazione OIDC, sono ideali per i modelli di autorizzazione avanzati con ambiti OAuth 2.0.

Il diagramma seguente illustra una tipica sessione di accesso per l'autenticazione gestita dell'accesso.



Flusso di autenticazione dell'accesso gestito

1. Un utente accede all'applicazione.
2. Selezionano un link «Accedi».
3. L'applicazione indirizza l'utente a una richiesta di accesso nelle pagine di accesso gestite del dominio del pool di utenti.
4. Inseriscono nome utente e password.

5. Il pool di utenti convalida le credenziali dell'utente e determina che l'utente ha attivato l'autenticazione a più fattori (MFA).
6. La pagina di accesso gestito richiede all'utente di inserire un codice MFA.
7. L'utente inserisce il proprio codice MFA.
8. Il pool di utenti reindirizza l'utente all'URL dell'applicazione.
9. L'applicazione raccoglie il codice di autorizzazione dal parametro di richiesta URL che ha gestito l'accesso aggiunto all'URL di callback.
10. L'applicazione richiede i token con il codice di autorizzazione.
11. L'endpoint del token restituisce JWT all'applicazione.
12. L'applicazione decodifica, convalida e archivia o memorizza nella cache i JWT dell'utente.
13. L'applicazione visualizza il componente con accesso controllato richiesto.
14. L'utente ne visualizza il contenuto.
15. Successivamente, il token di accesso dell'utente è scaduto e richiede di visualizzare un componente con accesso controllato.
16. L'applicazione determina che la sessione dell'utente debba persistere. Richiede nuovi token dall'endpoint del token con il token di aggiornamento.

Varianti e personalizzazioni

Puoi personalizzare l'aspetto delle tue pagine di accesso gestite con l'editor di branding per l'intero pool di utenti o a livello di qualsiasi client di app. Puoi anche configurare i client delle app con i propri provider di identità, ambiti, accesso agli attributi utente e configurazioni di sicurezza avanzate.

Risorse correlate

- Accesso gestito dal pool di utenti
- Ambiti, M2M e server di risorse
- Endpoint del pool di utenti e riferimento per l'accesso gestito

Autenticazione e autorizzazione dell'API del pool di utenti con un AWS SDK

AWS ha sviluppato componenti per i pool di utenti di Amazon Cognito, o provider di identità Amazon Cognito, in una varietà di framework di sviluppo. I metodi integrati in questi SDK chiamano l'API dei pool di utenti di Amazon Cognito. Lo stesso spazio dei nomi API dei pool di utenti include operazioni

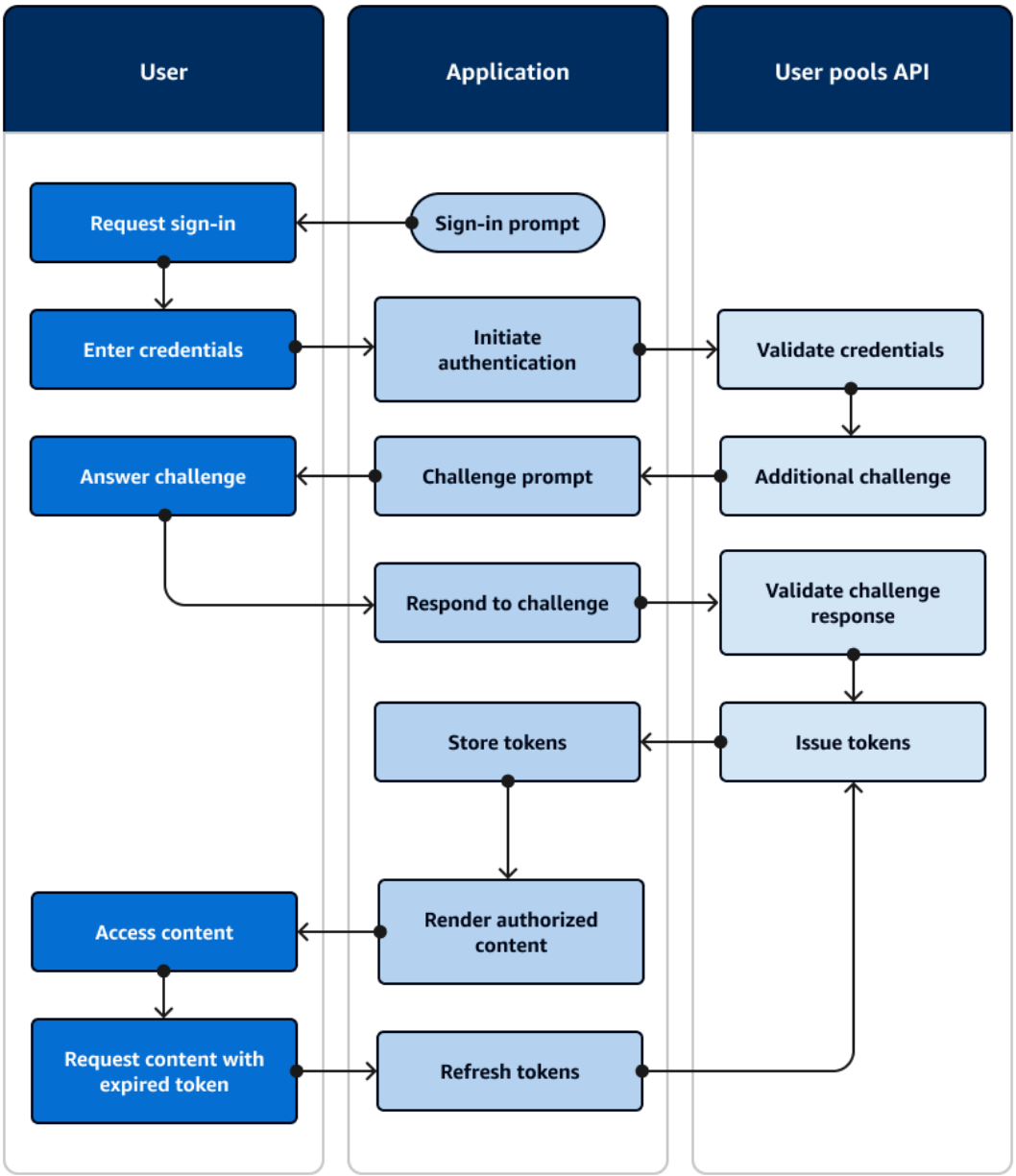
per la configurazione dei pool di utenti e per l'autenticazione degli utenti. Per una panoramica più completa, vedere. Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite

L'autenticazione API si adatta al modello in cui le applicazioni dispongono di componenti dell'interfaccia utente esistenti e si basano principalmente sul pool di utenti come directory utente. Questo design aggiunge Amazon Cognito come componente all'interno di un'applicazione più grande. Richiede una logica programmatica per gestire catene complesse di sfide e risposte.

Questa applicazione non ha bisogno di implementare un'implementazione completa del relying party OpenID Connect (OIDC). Invece, ha la capacità di decodificare e utilizzare JWT. Se desideri accedere al set completo di funzionalità del pool di utenti per gli utenti locali, crea la tua autenticazione con l'SDK Amazon Cognito nel tuo ambiente di sviluppo.

L'autenticazione API con ambiti OAuth personalizzati è meno orientata all'autorizzazione API esterna. Per aggiungere ambiti personalizzati a un token di accesso dall'autenticazione API, modifica il token in fase di esecuzione con un. Trigger Lambda di pre-generazione del token

Il diagramma seguente illustra una tipica sessione di accesso per l'autenticazione tramite API.



Flusso di autenticazione dell'API

1. Un utente accede alla tua applicazione.
2. Selezionano un link «Accedi».
3. Inseriscono nome utente e password.
4. L'applicazione richiama il metodo che effettua una richiesta InitiateAuthAPI. La richiesta passa le credenziali dell'utente a un pool di utenti.

5. Il pool di utenti convalida le credenziali dell'utente e determina che l'utente ha attivato l'autenticazione a più fattori (MFA).
6. Il pool di utenti risponde con una sfida che richiede un codice MFA.
7. L'applicazione genera un prompt che raccoglie il codice MFA dall'utente.
8. L'applicazione richiama il metodo che effettua una richiesta API. RespondToAuthChallenge La richiesta passa il codice MFA dell'utente.
9. Il pool di utenti convalida il codice MFA dell'utente.
10. Il pool di utenti risponde con i JWT dell'utente.
11. L'applicazione decodifica, convalida e archivia o memorizza nella cache i JWT dell'utente.
12. L'applicazione visualizza il componente con accesso controllato richiesto.
13. L'utente ne visualizza il contenuto.
14. Successivamente, il token di accesso dell'utente è scaduto e richiede di visualizzare un componente con accesso controllato.
15. L'applicazione determina che la sessione dell'utente debba persistere. Richiama nuovamente il InitiateAuthmetodo con il token di aggiornamento e recupera nuovi token.

Varianti e personalizzazioni

Puoi ampliare questo flusso con sfide aggiuntive, ad esempio sfide di autenticazione personalizzate. Puoi limitare automaticamente l'accesso agli utenti le cui password sono state compromesse o le cui caratteristiche di accesso impreviste potrebbero indicare un tentativo di accesso malevolo. Questo flusso ha lo stesso aspetto per le operazioni di registrazione, aggiornamento degli attributi utente e reimpostazione delle password. La maggior parte di questi flussi prevede operazioni API pubbliche (lato client) e riservate (lato server) duplicate.

Risorse correlate

- API dei pool di utenti di Amazon Cognito
- Crea una nuova applicazione nella console Amazon Cognito
- Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili
- Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite

Autenticazione del pool di utenti con un provider di identità di terze parti

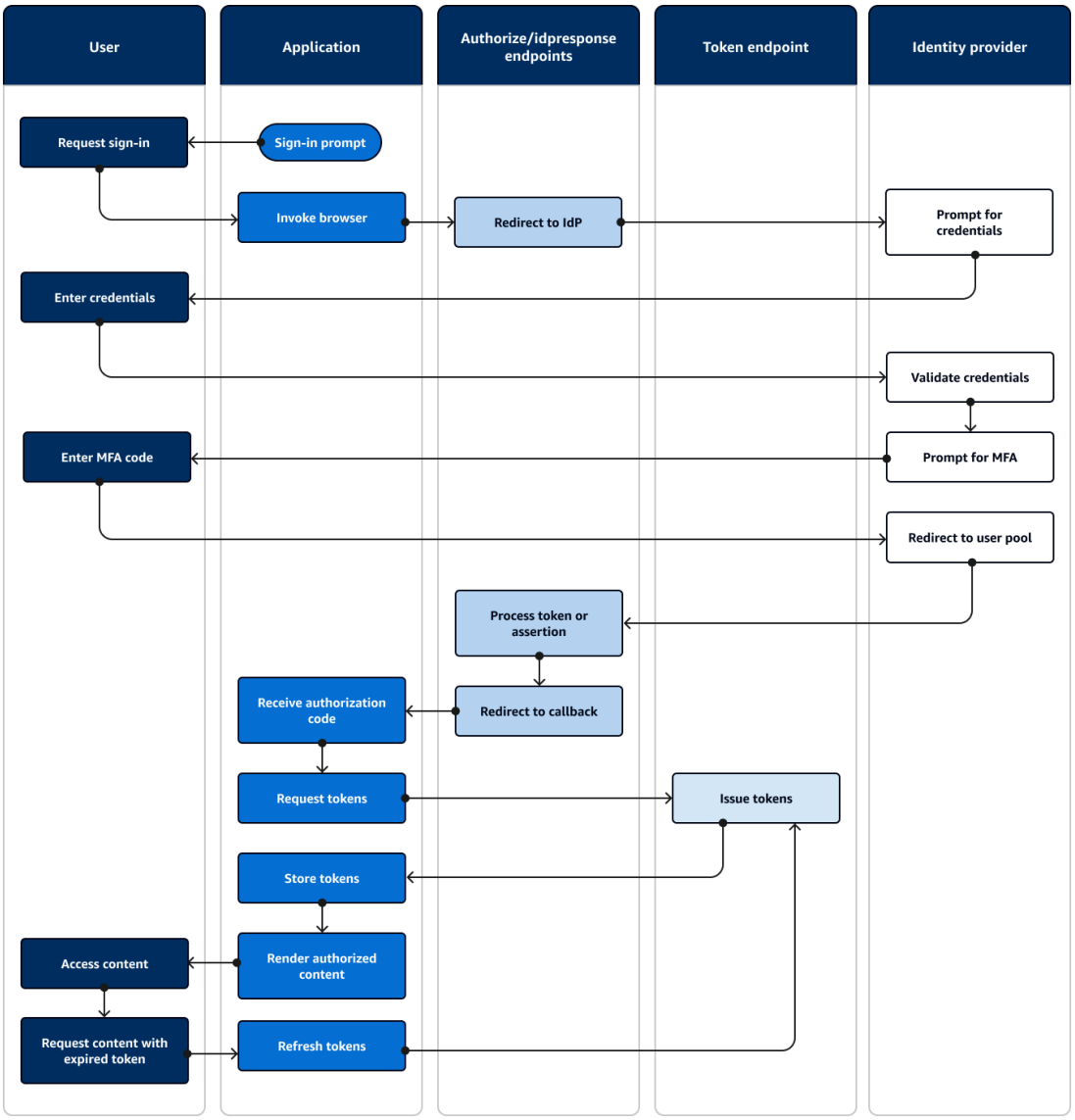
Sign-in con un provider di identità esterno (IdP), o autenticazione federata, è un modello simile all'accesso gestito. La tua applicazione è un relying party OIDC per il tuo pool di utenti, mentre il pool di utenti funge da passthrough per un IdP. L'IdP può essere una directory di utenti consumer come Facebook o Google, oppure può essere una directory aziendale SAML 2.0 o OIDC come Azure.

Invece di gestire l'accesso nel browser dell'utente, l'applicazione richiama un endpoint di reindirizzamento sul server di autorizzazione del pool di utenti. Dal punto di vista dell'utente, sceglie il pulsante di accesso nell'applicazione. Quindi il loro IdP li richiede di accedere. Come con l'autenticazione di accesso gestita, un'applicazione raccoglie i JWT in una posizione di reindirizzamento all'interno dell'app.

L'autenticazione con un IdP di terze parti si adatta a un modello in cui gli utenti potrebbero non voler inserire una nuova password quando si iscrivono alla tua applicazione. Third-party l'autenticazione può essere aggiunta con il minimo sforzo a un'applicazione che implementa l'autenticazione con accesso gestito. In effetti, l'accesso gestito e quello di terze parti IdPs producono un risultato di autenticazione coerente a partire da piccole variazioni di ciò che viene richiamato nei browser degli utenti.

Come l'autenticazione di accesso gestito, l'autenticazione federata è ideale per i modelli di autorizzazione avanzati con ambiti OAuth 2.0.

Il diagramma seguente illustra una tipica sessione di accesso per l'autenticazione federata.



Flusso di autenticazione federato

1. Un utente accede alla tua applicazione.
2. Selezionano un link «Accedi».
3. L'applicazione indirizza l'utente a una richiesta di accesso con il proprio IdP.
4. Inseriscono nome utente e password.
5. L'IdP convalida le credenziali dell'utente e determina che l'utente ha attivato l'autenticazione a più fattori (MFA).
6. L'IdP richiede all'utente di inserire un codice MFA.
7. L'utente inserisce il proprio codice MFA.

8. L'IdP reindirizza l'utente al pool di utenti con una risposta SAML o un codice di autorizzazione.
9. Se l'utente ha passato un codice di autorizzazione, il pool di utenti scambia silenziosamente il codice con token IdP. Il pool di utenti convalida i token IdP e reindirizza l'utente all'applicazione con un nuovo codice di autorizzazione.
10. L'applicazione raccoglie il codice di autorizzazione dal parametro di richiesta URL che il pool di utenti ha aggiunto all'URL di callback.
11. L'applicazione richiede i token con il codice di autorizzazione.
12. L'endpoint del token restituisce JWT all'applicazione.
13. L'applicazione decodifica, convalida e archivia o memorizza nella cache i JWT dell'utente.
14. L'applicazione visualizza il componente con accesso controllato richiesto.
15. L'utente ne visualizza il contenuto.
16. Successivamente, il token di accesso dell'utente è scaduto e richiede di visualizzare un componente con accesso controllato.
17. L'applicazione determina che la sessione dell'utente debba persistere. Richiede nuovi token dall'endpoint del token con il token di aggiornamento.

Varianti e personalizzazioni

Puoi avviare l'autenticazione federata nell'accesso gestito, dove gli utenti possono scegliere da un elenco di quelle IdPs che hai assegnato al client dell'app. L'accesso gestito può anche richiedere un indirizzo e-mail e indirizzare automaticamente la richiesta di un utente all'IdP SAML corrispondente. L'autenticazione con un provider di identità di terze parti non richiede l'interazione dell'utente con l'accesso gestito. L'applicazione può aggiungere un parametro di richiesta alla richiesta del server di autorizzazione dell'utente e fare in modo che l'utente reindirizzi silenziosamente alla pagina di accesso dell'IdP.

Risorse correlate

- Accesso al pool di utenti con provider di identità di terze parti
- Ambiti, M2M e server di risorse
- Endpoint del pool di utenti e riferimento per l'accesso gestito

Autenticazione del pool di identità

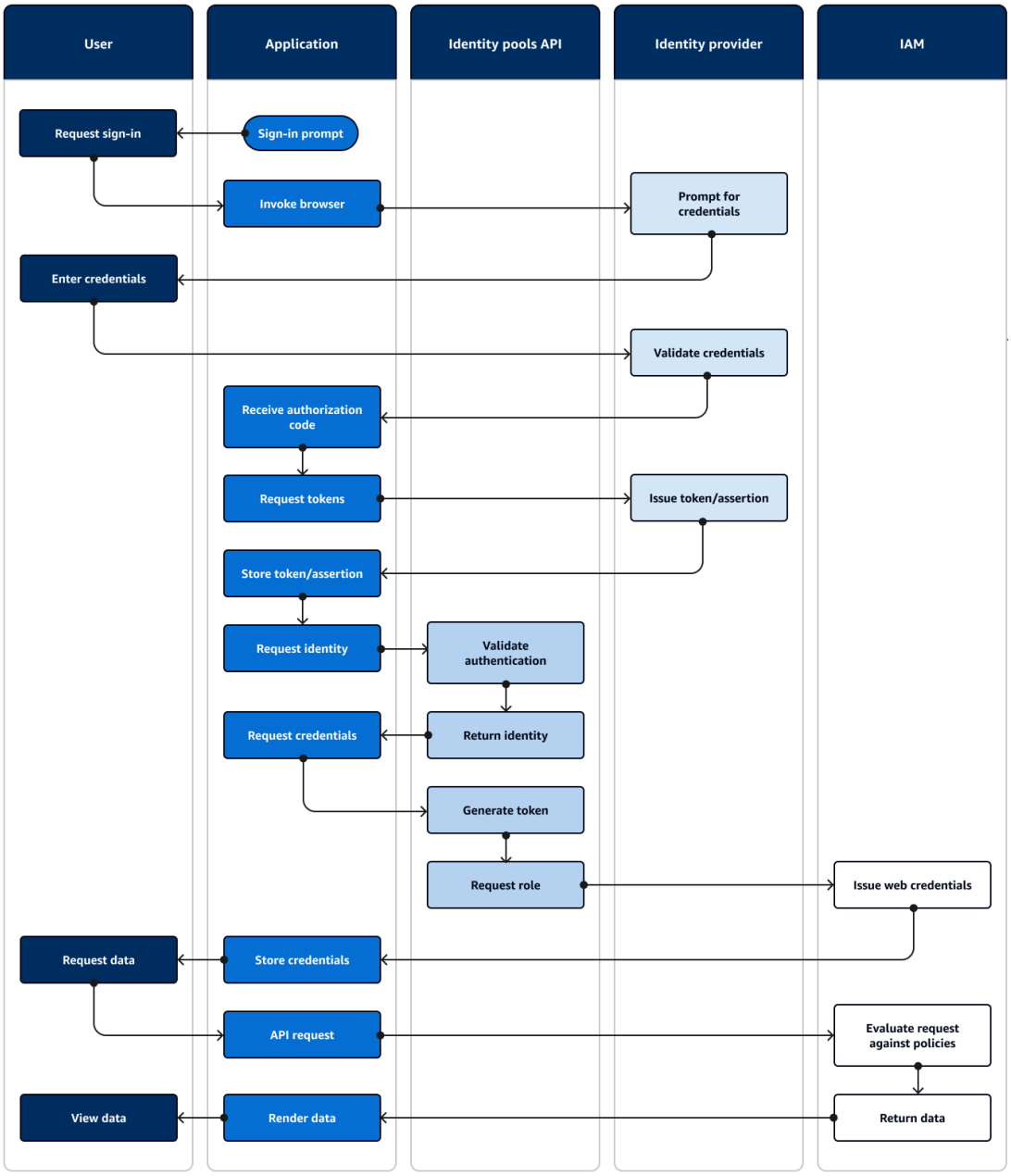
Un pool di identità è un componente dell'applicazione che si distingue da un pool di utenti in termini di funzioni, namespace API e modello SDK. Laddove i pool di utenti offrono l'autenticazione e l'autorizzazione basate su token, i pool di identità offrono l'autorizzazione per (IAM). AWS Identity and Access Management

Puoi assegnarne un set IdPs ai pool di identità e accedere agli utenti con essi. I pool di utenti sono strettamente integrati come pool di identità IdPs e offrono ai pool di identità la maggior parte delle opzioni per il controllo degli accessi. Allo stesso tempo, esiste un'ampia selezione di opzioni di autenticazione per i pool di identità. I pool di utenti si uniscono alle fonti di identità SAML, OIDC, social, per sviluppatori e ospiti come percorsi verso le AWS credenziali temporanee dai pool di identità.

L'autenticazione con un pool di identità è esterna: segue uno dei flussi del pool di utenti illustrati in precedenza o un flusso sviluppato indipendentemente con un altro IdP. Dopo aver eseguito l'autenticazione iniziale, l'applicazione passa la prova a un pool di identità e riceve in cambio una sessione temporanea.

L'autenticazione con un pool di identità si adatta a un modello in cui si impone il controllo degli accessi per gli asset e i dati delle applicazioni Servizi AWS con l'autorizzazione IAM. Analogamente all'autenticazione tramite API nei pool di utenti, un'applicazione di successo include AWS SDK per ciascuno dei servizi a cui desideri accedere a vantaggio degli utenti. AWS Gli SDK applicano le credenziali dell'autenticazione del pool di identità come firme alle richieste API.

Il diagramma seguente illustra una tipica sessione di accesso per l'autenticazione del pool di identità con un IdP.



Flusso di autenticazione del pool di identità

1. Un utente accede all'applicazione.
2. Selezionano un link «Accedi».
3. L'applicazione indirizza l'utente a una richiesta di accesso con il proprio IdP.
4. Inseriscono nome utente e password.
5. L'IdP convalida le credenziali dell'utente.

6. L'IdP reindirizza l'utente all'applicazione con una risposta SAML o un codice di autorizzazione.
7. Se l'utente ha passato un codice di autorizzazione, l'applicazione scambia il codice con token IdP.
8. L'applicazione decodifica, convalida e archivia o memorizza nella cache i JWT o l'asserzione dell'utente.
9. L'applicazione richiama il metodo che effettua una richiesta API. GetId Passa il token o l'asserzione dell'utente e richiede un ID di identità.
10. Il pool di identità convalida il token o l'asserzione rispetto ai provider di identità configurati.
11. Il pool di identità restituisce un ID di identità.
12. L'applicazione richiama il metodo che effettua una richiesta GetCredentialsForIdentityAPI. Passa il token o le asserzioni dell'utente e richiede un ruolo IAM.
13. Il pool di identità genera un nuovo JWT. Il nuovo JWT contiene affermazioni che richiedono un ruolo IAM. Il pool di identità determina il ruolo in base alla richiesta dell'utente e ai criteri di selezione del ruolo nella configurazione del pool di identità per l'IdP.
14. Servizio di token di sicurezza AWS (AWS STS) risponde alla AssumeRoleWithWebIdentityrichiesta del pool di identità. La risposta contiene le credenziali API per una sessione temporanea con un ruolo IAM.
15. L'applicazione memorizza le credenziali della sessione.
16. L'utente esegue un'azione nell'app che richiede risorse con accesso protetto in. AWS
17. L'applicazione applica le credenziali temporanee come firme alle richieste API per quanto richiesto. Servizi AWS
18. IAM valuta le politiche associate al ruolo nelle credenziali. Le confronta con la richiesta.
19. Servizio AWS Restituisce i dati richiesti.
20. L'applicazione esegue il rendering dei dati nell'interfaccia utente.
21. L'utente visualizza i dati.

Varianti e personalizzazioni

Per visualizzare l'autenticazione con un pool di utenti, inserisci una delle precedenti panoramiche del pool di utenti dopo la fase Issue. token/assertion L'autenticazione dello sviluppatore sostituisce tutti i passaggi precedenti a Request identity con una richiesta firmata dalle credenziali dello sviluppatore. L'autenticazione guest passa inoltre direttamente a Request identity, non convalida l'autenticazione e restituisce le credenziali per un ruolo IAM ad accesso limitato.

Risorse correlate

- Pool di identità di Amazon Cognito
- Ruoli IAM dell'utente
- Flusso di autenticazione dei pool di identità

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
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

Documentazione sugli SDK	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

> **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.

Autorizzazione con Amazon Verified Permissions

Amazon Verified Permissions è un servizio di autorizzazione per le applicazioni che crei. Quando aggiungi un pool di utenti Amazon Cognito come fonte di identità, la tua app può passare i token di accesso o di identità (ID) al pool di utenti alle Autorizzazioni verificate per consentire o negare una decisione. Verified Permissions considera le proprietà dell'utente e il contesto della richiesta in base alle politiche redatte in Cedar Policy Language. Il contesto della richiesta può includere un identificatore per il documento, l'immagine o l'altra risorsa richiesta e l'azione che l'utente desidera intraprendere sulla risorsa.

La tua app può fornire l'identità dell'utente o i token di accesso alle autorizzazioni verificate nelle IsAuthorizedWithTokenrichieste BatchIsAuthorizedWithTokenAPI. Queste operazioni API accettano gli utenti come utenti `Principal` e prendono decisioni di `Action` autorizzazione per `Resource` chi desidera accedere. Ulteriori personalizzazioni `Context` possono contribuire a una decisione di accesso dettagliata.

Quando la tua app presenta un token in una richiesta `IsAuthorizedWithToken` API, Verified Permissions esegue le seguenti convalide.

1. Il tuo pool di utenti è un'origine di identità di Verified Permissions configurata per il policy store richiesto.
2. La richiesta `client_id` o `aud`, rispettivamente, nel tuo token di accesso o di identità, corrisponde all'ID client dell'app pool di utenti che hai fornito a Verified Permissions. Per verificare questa richiesta, devi configurare la convalida dell'ID cliente nella tua fonte di identità Autorizzazioni verificate.

3. Il tuo token non è scaduto.
4. Il valore dell'`token_use`attestazione nel tuo token corrisponde ai parametri a cui lo hai passato`IsAuthorizedWithToken`. L'`token_use`attestazione deve essere `access` se l'hai passata al `accessToken` parametro e `id` se l'hai passata al `identityToken` parametro.
5. La firma nel tuo token proviene dalle chiavi web JSON pubblicate (JWK) del tuo pool di utenti. Puoi visualizzare i tuoi JWK su `https://cognito-idp.`*`Region`*`.amazonaws.com/`*`your user pool ID`*`/.well-known/jwks.json`.

Token revocati e utenti eliminati

Verified Permissions convalida solo le informazioni che conosce dalla fonte della tua identità e dalla data di scadenza del token dell'utente. Verified Permissions non verifica la revoca del token o l'esistenza dell'utente. Se hai revocato il token dell'utente o hai eliminato il profilo utente dal tuo pool di utenti, Verified Permissions considera comunque il token valido fino alla scadenza.

Valutazione delle politiche

Configura il tuo pool di utenti come fonte di identità per il tuo policy store. Configura la tua app per inviare i token degli utenti nelle richieste a Verified Permissions. Per ogni richiesta, Verified Permissions confronta le affermazioni contenute nel token con una politica. Una politica di Verified Permissions è come una politica IAM in AWS. Dichiara un principio, una risorsa e un'azione. Verified Permissions risponde alla tua richiesta indicando `Allow` se corrisponde a un'azione consentita e non corrisponde a un'`Deny`azione esplicita; in caso contrario, risponde con. `Deny` Per ulteriori informazioni, consulta le politiche relative a Amazon Verified Permissions nella Guida per l'utente di Amazon Verified Permissions.

Personalizzazione dei token

Per modificare, aggiungere e rimuovere le affermazioni degli utenti che desideri presentare a Verified Permissions, personalizza il contenuto dei tuoi token di accesso e di identità con un. Trigger Lambda di pre-generazione del token Con un trigger prima della generazione di token, puoi aggiungere e modificare le richieste nei tuoi token. Ad esempio, puoi interrogare un database per gli attributi utente aggiuntivi e codificarli nel tuo token ID.

> **Note**
>
> A causa del modo in cui Verified Permissions elabora le richieste, non aggiungerle con nome `cognito` , `dev` o `custom` nella funzione di pre-generazione del token. Quando presenti

questi prefissi riservati per richieste non in un formato delimitato da due punti, ad esempio come `cognito:username`, ma come nomi completi delle richieste, le tue richieste di autorizzazione hanno esito negativo.

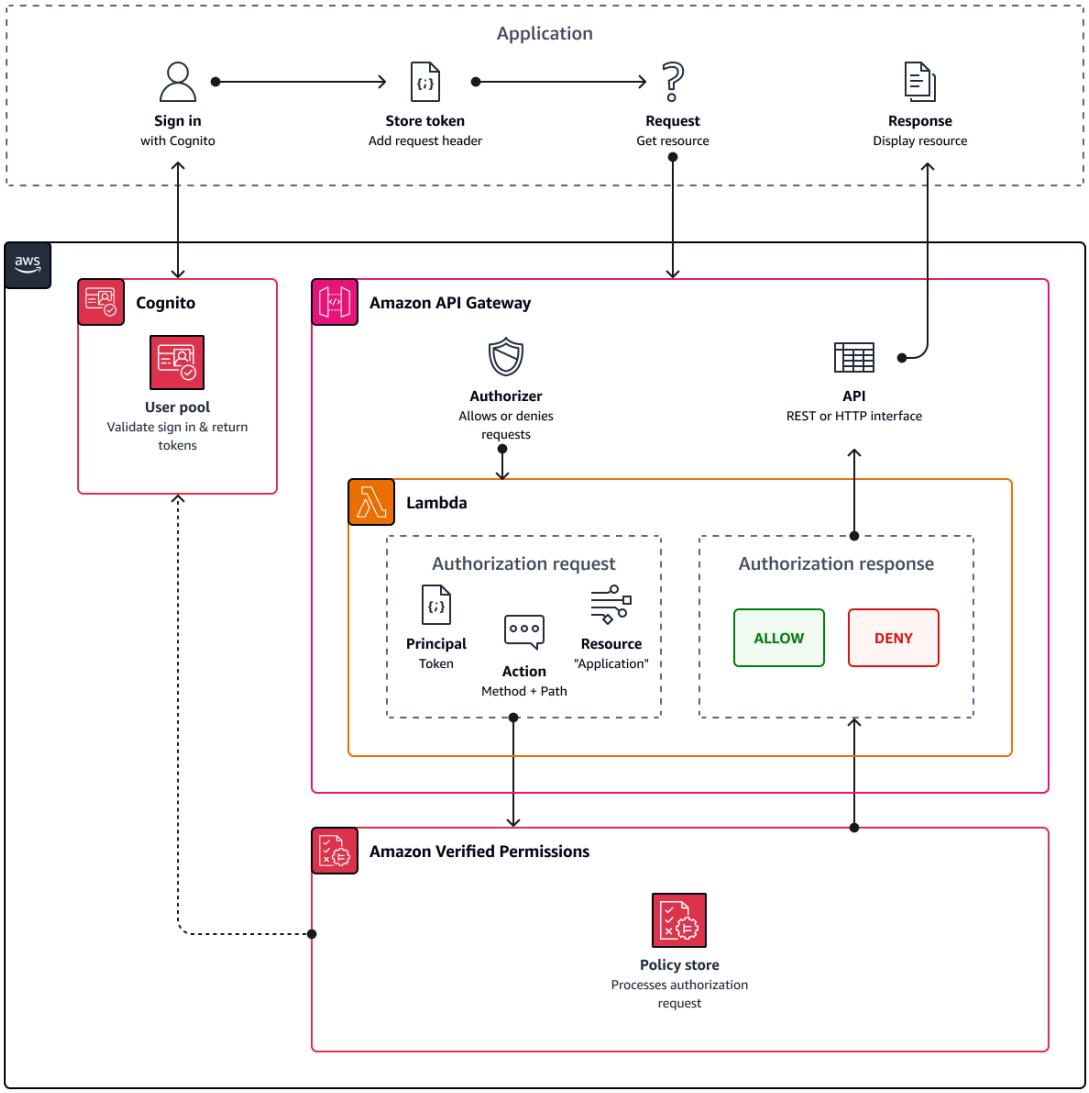
Risorse aggiuntive

- Mappatura dei token Amazon Cognito allo schema di autorizzazioni verificate
- Autorizza le API API Gateway utilizzando Amazon Verified Permissions e Amazon Cognito
- Workshop: Autenticazione e autorizzazione con Amazon Cognito e autorizzazioni verificate

Autorizzazione API con autorizzazioni verificate

Il tuo ID o i token di accesso possono autorizzare le richieste alle API REST di Amazon API Gateway di back-end con autorizzazioni verificate. Puoi creare un archivio di policy con collegamenti immediati al tuo pool di utenti e alla tua API. Con l'opzione di avvio Configurazione con API Gateway e un'origine di identità, Verified Permissions aggiunge un'origine di identità del pool di utenti al policy store e un autorizzatore Lambda all'API. Quando l'applicazione passa un token portatore del pool di utenti all'API, l'autorizzatore Lambda richiama le autorizzazioni verificate. L'autorizzatore passa il token come principale e il percorso e il metodo della richiesta come azione.

Il diagramma seguente illustra il flusso di autorizzazione per un'API API Gateway con autorizzazioni verificate. Per un'analisi dettagliata, consulta gli archivi API-linked delle politiche nella Amazon Verified Permissions User Guide.



Verified Permissions struttura l'autorizzazione delle API in base ai gruppi di pool di utenti. Poiché sia l'ID che i token di accesso includono un `cognito:groups` claim, il policy store può gestire il controllo degli accessi basato sui ruoli (RBAC) per le API in una varietà di contesti applicativi.

Scelta delle impostazioni del policy store

Quando si configura un'origine di identità in un policy store, è necessario scegliere se elaborare i token di accesso o ID. Questa decisione è importante per il modo in cui funziona il motore delle policy. I token ID contengono attributi utente. I token di accesso contengono informazioni sul controllo degli accessi degli utenti: ambiti OAuth. Sebbene entrambi i tipi di token contengano informazioni sull'appartenenza al gruppo, in genere consigliamo il token di accesso per RBAC con un archivio di policy di autorizzazioni verificate. Il token di accesso si aggiunge all'appartenenza al gruppo

con ambiti che possono contribuire alla decisione di autorizzazione. Le affermazioni in un token di accesso diventano contesto nella richiesta di autorizzazione.

È inoltre necessario configurare i tipi di entità utente e di gruppo quando si configura un pool di utenti come fonte di identità. I tipi di entità sono identificatori principali, di azioni e di risorse a cui puoi fare riferimento nelle politiche di autorizzazione verificate. Le entità negli archivi delle politiche possono avere una relazione di appartenenza, in cui un'entità può essere membro di un'entità principale. Con l'appartenenza, puoi fare riferimento a gruppi principali, gruppi di azione e gruppi di risorse. Nel caso di gruppi di pool di utenti, il tipo di entità utente specificato deve essere un membro del tipo di entità di gruppo. Quando si configura un API-linkedpolicy store o si segue la configurazione guidata nella console Verified Permissions, il policy store instaura automaticamente questa relazione genitore-membro.

Il token ID può combinare RBAC con il controllo degli accessi basato sugli attributi (ABAC). Dopo aver creato un archivio di API-linked politiche, è possibile migliorare le politiche con gli attributi utente e l'appartenenza ai gruppi. Le dichiarazioni di attributo in un token ID diventano gli attributi principali nella richiesta di autorizzazione. Le tue politiche possono prendere decisioni di autorizzazione in base agli attributi principali.

Puoi anche configurare un policy store per accettare token con un'`client_id`affermazione `aud` or che corrisponda a un elenco di app client accettabili da te fornito.

Esempio di politica per l'autorizzazione delle API basata sui ruoli

La seguente policy di esempio è stata creata configurando un archivio di policy di Verified Permissions per un'API REST di PetStoreesempio.

```
permit(
  principal in PetStore::UserGroup::"us-east-1_EXAMPLE|MyGroup",
  action in [ PetStore::Action::"get /pets", PetStore::Action::"get /pets/{petId}" ],
  resource
  );
```

Verified Permissions restituisce una `Allow` decisione sulla richiesta di autorizzazione dell'applicazione quando:

1. L'applicazione ha passato un ID o un token di accesso in un'`Authorization`intestazione come token portatore.
2. L'applicazione ha passato un token con un'`cognito:groups`affermazione che contiene la stringa. `MyGroup`

3. L'applicazione ha inviato una `HTTP GET` richiesta, ad esempio, a `https://myapi.example.com/pets` o`https://myapi.example.com/pets/scrappy`.

Esempio di policy per un utente Amazon Cognito.

Il tuo pool di utenti può anche generare richieste di autorizzazione a Verified Permissions in condizioni diverse dalle richieste API. Puoi inviare qualsiasi decisione di controllo degli accessi contenuta nella tua applicazione al tuo policy store. Ad esempio, puoi integrare la sicurezza di Amazon DynamoDB o Amazon S3 con il controllo degli accessi basato sugli attributi prima che le richieste transitino sulla rete, riducendo l'utilizzo delle quote.

L'esempio seguente utilizza il Cedar Policy Language per consentire agli utenti Finance che si autenticano con un client dell'app pool di utenti di leggere e scrivere `example_image.png`. John, un utente della tua app, riceve un token ID dal client dell'app e lo trasmette in una richiesta GET a un URL che richiede l'autorizzazione, `https://example.com/images/example_image.png`. Il token ID di John ha una richiesta `aud` dell'ID client dell'app del pool di utenti `1234567890example`. La tua funzione Lambda di prima generazione del token ha anche inserito una nuova affermazione `costCenter` con un valore, per John, di `Finance1234`.

```
permit (
   principal,
   actions in [ExampleCorp::Action::"readFile", "writeFile"],
   resource == ExampleCorp::Photo::"example_image.png"
)
when {
   principal.aud == "1234567890example" &&
   principal.custom.costCenter like "Finance*"
};
```

Il seguente corpo di richiesta restituisce una risposta `Allow`.

```
{
   "accesstoken": "[John's ID token]",
   "action": {
      "actionId": "readFile",
      "actionType": "Action"
   },
   "resource": {
      "entityId": "example_image.png",
```

```
      "entityType": "Photo"
   }
}
```

Quando desideri specificare un principale in una politica di autorizzazioni verificate, utilizza il seguente formato:

```
permit (
   principal == [Namespace]::[Entity]::"[user pool ID]|[user sub]",
   action,
   resource
);
```

Di seguito è riportato un esempio principale per un utente in un pool di utenti con ID `us-east-1_Example` con ID secondario o ID utente. `973db890-092c-49e4-a9d0-912a4c0a20c7`

```
principal == ExampleCorp::User::"us-east-1_Example|973db890-092c-49e4-
a9d0-912a4c0a20c7",
```

Quando desideri specificare un gruppo di utenti in una politica di autorizzazioni verificate, utilizza il seguente formato:

```
permit (
   principal in [Namespace]::[Group Entity]::"[Group name]",
   action,
   resource
);
```

Attribute-based controllo degli accessi

L'autorizzazione con autorizzazioni verificate per le tue app e gli attributi per la funzionalità di controllo degli accessi dei pool di identità di Amazon Cognito AWS per le credenziali sono entrambe forme di controllo degli accessi basato sugli attributi (ABAC). Di seguito è riportato un confronto tra le funzionalità di Verified Permissions e Amazon Cognito ABAC. In ABAC, un sistema esamina gli attributi di un'entità e prende una decisione di autorizzazione in base a condizioni definite dall'utente.

Servizio	Processo	Risultato
Autorizzazioni verificate da Amazon	Restituisce una `Deny` decisione `Allow` or dall'analisi di un pool di utenti JWT.	L'accesso alle risorse dell'applicazione riesce o fallisce in base alla valutazione delle politiche Cedar.
Pool di identità di Amazon Cognito (attributi per il controllo degli accessi)	Assegna i tag di sessione all'utente in base ai suoi attributi. Le condizioni delle policy IAM possono controllare i tag `Allow` o `Deny` l'accesso degli utenti a Servizi AWS.	Una sessione con tag con AWS credenziali temporanee per un ruolo IAM.

Endpoint del pool di utenti e riferimento per l'accesso gestito

Amazon Cognito offre due modelli di autenticazione dei pool di utenti: con l'API dei pool di utenti e con il server di autorizzazione OAuth 2.0. Usa l'API quando desideri recuperare i token OpenID Connect (OIDC) AWS con un SDK nel backend dell'applicazione. Utilizza il server di autorizzazione quando desideri implementare il tuo pool di utenti come provider OIDC. Il server di autorizzazione aggiunge funzionalità come l'accesso federato, l'autorizzazione API e M2M con ambiti dei token di accesso e l'accesso gestito. È possibile utilizzare i modelli API e OIDC singolarmente o insieme, configurati a livello di pool di utenti o a livello di client dell'app. Questa sezione è un riferimento per l'implementazione del modello OIDC. Per ulteriori informazioni sui due modelli di autenticazione, vedere. Comprendere l'API, l'OIDC e l'autenticazione delle pagine di accesso gestite

Amazon Cognito attiva le pagine Web pubbliche elencate qui quando si assegna un dominio al pool di utenti. Il dominio funge da punto di accesso centrale per tutti i client dell'app. Includono l'accesso gestito, in cui gli utenti possono registrarsi e accedere (Endpoint Login) e disconnettersi (Endpoint Logout). Per ulteriori informazioni su queste risorse, vedere Accesso gestito dal pool di utenti.

Queste pagine includono anche le risorse Web pubbliche che consentono al pool di utenti di comunicare con provider di identità SAML, OpenID Connect (OIDC) e OAuth 2.0 di terze parti (). IdPs Per accedere a un utente con un provider di identità federato, gli utenti devono inviare una richiesta all'accesso gestito interattivo o all'OIDC. Endpoint Login Endpoint Authorize L'endpoint Authorize reindirizza gli utenti alle pagine di accesso gestite o alla pagina di accesso IdP.

L'app può anche effettuare l'accesso come utenti locali con l'API dei pool di utenti Amazon Cognito. Un utente locale esiste esclusivamente nella directory del pool di utenti senza federazione tramite un IdP esterno.

Oltre all'accesso gestito, Amazon Cognito si integra con gli SDK per Android, JavaScript iOS e altri. Gli SDK forniscono strumenti per eseguire operazioni API del pool di utenti con endpoint del servizio dell'API Amazon Cognito. Per ulteriori informazioni su questi endpoint del servizio, consulta Endpoint e quote di Amazon Cognito Identity.

> **Warning**
>
> Non aggiungere i certificati Transport Layer Security (TLS) di entità finale o intermedi per i domini Amazon Cognito. AWS gestisce tutti i certificati per tutti gli endpoint e i domini con prefisso del pool di utenti. Le autorità di certificazione (CA) della catena di attendibilità che supporta i certificati Amazon Cognito si alternano e si rinnovano dinamicamente. Quando

aggiungi la tua app a un certificato intermedio o principale, l'app può fallire senza preavviso durante la rotazione dei certificati. AWS
Associa invece la tua applicazione a tutti i certificati root Amazon disponibili. Per ulteriori informazioni, consulta le best practice e i suggerimenti in Associazione dei certificati nella Guida per l'utente di AWS Certificate Manager .

Per un elenco degli endpoint del servizio per l'API dei pool di utenti Regione AWS, consulta Service endpoints nel General Reference.AWS

Argomenti

- User-interactive accesso gestito e endpoint dell'interfaccia utente ospitati in modalità classica
- Endpoint del provider di identità e del relying party
- Concessioni OAuth 2.0
- Utilizzo di PKCE nelle concessioni di codici di autorizzazione
- Risposte gestite agli errori di accesso e federazione

User-interactive accesso gestito e endpoint dell'interfaccia utente ospitati in modalità classica

Amazon Cognito attiva gli endpoint di accesso gestiti in questa sezione quando aggiungi un dominio al tuo pool di utenti. Sono pagine Web in cui gli utenti possono completare le operazioni di autenticazione di base di un pool di utenti. Includono pagine per la gestione delle password, l'autenticazione a più fattori (MFA) e la verifica degli attributi.

Le pagine Web che costituiscono l'accesso gestito sono un'applicazione Web front-end per sessioni utente interattive con i clienti. L'app deve richiamare l'accesso gestito nei browser degli utenti. Amazon Cognito non supporta l'accesso programmatico alle pagine Web di questo capitolo. Gli endpoint di federazione in Endpoint del provider di identità e del relying party che restituiscono una risposta JSON possono essere interrogati direttamente nel codice dell'app. I Endpoint Authorize reindirizzamenti all'accesso gestito o a una pagina di accesso IdP devono inoltre essere aperti nei browser degli utenti.

Tutti gli endpoint del pool di utenti accettano il traffico proveniente da indirizzi IP di origine IPv4 e IPv6.

Gli argomenti di questa guida descrivono in dettaglio l'accesso gestito di uso frequente e gli endpoint dell'interfaccia utente ospitati classici. La differenza tra l'accesso gestito e l'interfaccia utente ospitata è visibile, non funzionale. Ad eccezione di`/passkeys/add`, tutti i percorsi sono condivisi tra le due versioni del branding di accesso gestito.

Amazon Cognito rende disponibili le pagine Web seguenti quando si assegna un dominio al pool di utenti.

Endpoint di accesso gestiti

URL dell'endpoint	Description	Modalità di accesso
https:///login *Your user pool domain*	Accede agli utenti locali e federati del pool di utenti.	Esegui il reindirizzamento da endpoint come Endpoint Authorize, `/logout` e `/confirmforgotPassword` . Consulta Endpoint Login.
https:///logout *Your user pool domain*	Disconnette gli utenti del pool di utenti.	Collegamento diretto. Consulta Endpoint Logout.
https:///ConfirmUser *Your user pool domain*	Conferma gli utenti che hanno selezionato un collegamento e-mail per verificare il proprio account utente.	Collegamento selezionato dall'utente in un messaggio di posta elettronica.
https:///registrazione *Your user pool domain*	Registra un nuovo utente. La pagina `/login` indirizza l'utente a `/signup` quando si seleziona Sign up (Registrati).	Collegamento diretto con gli stessi parametri di `/oauth2/authorize` .
https:///conferma *Your user pool domain*	Dopo che il pool di utenti invia un codice di conferma a un utente che ha effettuato la registrazione, richiede il codice all'utente.	Redirect-only da`/signup`.
https://*Your user pool domain*/Password dimenticata	Richiede all'utente il nome utente e invia un codice	1. Dal collegamento Password dimenticata a `/login`.

URL dell'endpoint	Description	Modalità di accesso
	per la reimpostazione della password. La pagina `/login` indirizza l'utente a `/forgotPassword` quando si seleziona Forgot your password? (Hai dimenticato la password?).	2. Collegamento diretto con gli stessi parametri di `/oauth2/authorize` .
https://*Your user pool domain*/ConfirmForgotPassword	Richiede all'utente il codice di ripristino della password e una nuova password. La pagina `/forgotPassword` indirizza l'utente a `/confirmforgotPassword` quando si seleziona Reset your password (Reimposta la password). Dopo che l'utente ha inviato un codice di ripristino valido e una nuova password, lo `/confirmforgotPassword` reindirizza a `/login` all'accesso con la nuova password.	Redirect-only da`/forgotPassword` .
https://*Your user pool domain*/resendcode	Invia un nuovo codice di conferma a un utente che ha effettuato la registrazione nel pool di utenti.	Redirect-only da Invia un nuovo link di codice all'indirizzo. `/confirm`

URL dell'endpoint	Description	Modalità di accesso
https://*Your user pool domain*/passkeys/add	Registra una nuova passkey. Disponibile solo nell'accesso gestito.	• Nel flusso di registrazione dopo la conferma nei client dell'app che supportano l'autenticazione con chiave di accesso. • Collegamento diretto con gli stessi parametri di `/oauth2/authorize` .

Argomenti

- L'endpoint di accesso gestito: /login
- L'endpoint di disconnessione di accesso gestito: /logout

L'endpoint di accesso gestito: /login

L'endpoint di accesso è un server di autenticazione e una destinazione di reindirizzamento da. Endpoint Authorize È il punto di accesso all'accesso gestito quando non si specifica un provider di identità. Quando generi un reindirizzamento all'endpoint di accesso, la pagina di accesso viene caricata e vengono presentate le opzioni di autenticazione configurate per il client all'utente.

> **Note**
>
> L'endpoint di accesso è un componente dell'accesso gestito. Nell'app, richiama le pagine di federazione e di accesso gestito che reindirizzano all'endpoint di accesso. L'accesso diretto da parte degli utenti all'endpoint di accesso non è una best practice.

GET /login

L'endpoint `/login` supporta solo `HTTPS GET` per la richiesta iniziale dell'utente. L'app richiama la pagina in un browser come Chrome o Firefox. Quando esegui il reindirizzamento `/login` da aEndpoint Authorize, vengono trasmessi tutti i parametri che hai fornito nella richiesta iniziale. L'endpoint di accesso supporta tutti i parametri di richiesta dell'endpoint Authorize. Puoi anche

accedere direttamente all'endpoint di accesso. Come best practice, crea tutte le sessioni degli utenti in `/oauth2/authorize`.

Esempio: richiedere all'utente di accedere

In questo esempio viene visualizzata la schermata di accesso.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/login?
                response_type=code&
                client_id=1example23456789&
                redirect_uri=https://YOUR_APP/redirect_uri&
                state=STATE&
                scope=openid+profile+aws.cognito.signin.user.admin
```

Esempio: risposta

Il server di autenticazione reindirizza all'app con lo stato e il codice di autorizzazione. IL server deve restituire il codice e lo stato nei parametri della stringa di query e non nel frammento.

```
HTTP/1.1 302 Found
                    Location: https://YOUR_APP/redirect_uri?
code=AUTHORIZATION_CODE&state=STATE
```

User-initiated richiesta di accesso

Dopo aver caricato l'endpoint `/login`, l'utente può inserire un nome utente e una password e scegliere Accedi. Quando lo fanno, generano una richiesta `HTTPS POST` con gli stessi parametri di richiesta di intestazione come richiesta `GET` e un corpo della richiesta con il nome utente, la password e l'impronta digitale del dispositivo.

Step-up richiesta di autenticazione

L'`/login`endpoint accetta i seguenti parametri URL per l'autenticazione avanzata. Questi parametri hanno lo stesso comportamento dell'endpoint di autorizzazione. Per ulteriori informazioni sui livelli ACR e sui valori AMR, vedere. the section called “Livelli di autenticazione con attestazioni ACR e AMR”

`acr_values`

Un URL-encoded elenco separato da spazi degli URI del livello di autenticazione ACR di destinazione, dal più alto al più basso. Amazon Cognito valuta l'elenco da sinistra a destra, applica il primo livello valido che l'utente può soddisfare e ignora i valori che non riconosce. Questo parametro si comporta come il parametro API. `TARGET_ACR_VALUES` Richiede il piano di funzionalità Essentials o Plus.

`max_age`

Un numero intero non negativo che specifica il numero massimo di secondi trascorsi dall'ultima autenticazione dell'utente. Se il valore dell'ora corrente in meno `auth_time` è maggiore di`max_age`, l'utente deve eseguire nuovamente l'autenticazione da zero anziché limitarsi a procedere. Questo comportamento è identico al parametro API. `MAX_AGE`

Per una richiesta di esempio che utilizza questi parametri, consulta l'endpoint di autorizzazione.

Combina `acr_values` con `max_age` per richiedere un livello di autenticazione specifico con l'autenticazione recente per un'operazione sensibile.

Il comportamento di un piano di funzionalità insufficiente è diverso da quello dell'API. L'`/login`endpoint di accesso gestito ignora silenziosamente l'autenticazione `acr_values` e procede con la normale autenticazione, senza errori. L'API restituisce un. `FeatureUnavailableInTierException` Per ulteriori informazioni, consulta Requisiti del piano di funzionalità.

L'endpoint di disconnessione di accesso gestito: /logout

L'endpoint `/logout` è un endpoint di reindirizzamento. Disconnette l'utente e lo reindirizza a un URL di disconnessione autorizzato per il client dell'app o all'endpoint. `/login` I parametri disponibili in una richiesta GET all'`/logout`endpoint sono personalizzati in base ai casi d'uso di accesso gestito di Amazon Cognito.

L'endpoint di logout è un'applicazione web front-end per sessioni utente interattive con i tuoi clienti. L'app deve richiamare questo e altri endpoint di accesso gestito nei browser degli utenti.

Per reindirizzare l'utente all'accesso gestito per accedere nuovamente, aggiungi un `redirect_uri` parametro alla tua richiesta. Una richiesta `logout` con un parametro `redirect_uri` deve includere anche i parametri per la richiesta successiva a Endpoint Login, come `client_id`, `response_type` e `scope`.

Per reindirizzare l'utente a una pagina preferita, aggiungi URL di disconnessione consentiti all'app del client. Nelle richieste degli utenti all'endpoint `logout`, aggiungi i parametri `logout_uri` e `client_id`. Se il valore di `logout_uri` è uno degli URL di disconnessione consentiti per il client dell'app, Amazon Cognito reindirizza gli utenti a tale URL.

Con il single logout (SLO) per SAML 2.0, Amazon IdPs Cognito reindirizza innanzitutto l'utente all'endpoint SLO definito nella configurazione IdP. Dopo che il tuo IdP ha reindirizzato l'utente a, Amazon `saml2/logout` Cognito risponde con un altro reindirizzamento alla o dalla tua richiesta. `redirect_uri` `logout_uri` Per ulteriori informazioni, consulta Disconnessione degli utenti SAML con accesso singolo.

L'endpoint di logout non disconnette gli utenti dall'OIDC o dai provider di identità social (). IdPs Per disconnettere gli utenti dalla sessione con un IdP esterno, indirizzali alla pagina di disconnessione di quel provider.

GET /logout

L'endpoint `/logout` supporta solo `HTTPS GET`. Il client del bacino d'utenza in genere invia questa richiesta tramite un browser di sistema. Il browser è in genere Custom Chrome Tab in Android o Safari View Control in iOS.

Parametri della richiesta

client_id

L'ID client dell'app per l'app. Per ottenere l'ID del client di un'app, devi registrare l'app nel bacino d'utenza. Per ulteriori informazioni, consulta Application-specific impostazioni con i client dell'app.

Obbligatorio.

logout_uri

Reindirizza l'utente a una pagina di disconnessione personalizzata con un parametro logout_uri. Imposta il suo valore sull'URL di disconnessione del client dell'app a cui vuoi reindirizzare l'utente dopo la disconnessione. Utilizza logout_uri solo con il parametro client_id. Per ulteriori informazioni, consulta Application-specific impostazioni con i client dell'app.

Puoi utilizzare il parametro logout_uri anche per reindirizzare l'utente alla pagina di accesso di un altro client di app. Imposta la pagina di accesso per l'altro client di app come Allowed callback URL (URL di callback consentito) nel tuo client di app. Nella richiesta all'`/logout`endpoint, imposta il valore del parametro logout_uri sulla pagina di accesso. URL-encoded

Amazon Cognito richiede un parametro logout_uri o redirect_uri nella richiesta all'endpoint `/logout`. Il parametro logout_uri reindirizza l'utente a un altro sito Web. Se nella richiesta all'endpoint sono inclusi entrambi i parametri logout_uri e redirect_uri, `/logout` Amazon Cognito utilizza esclusivamente il parametro logout_uri, sovrascrivendo il parametro redirect_uri.

`nonce`

(Facoltativo) Un valore casuale che puoi aggiungere alla richiesta. Il valore nonce fornito è incluso nel token ID emesso da Amazon Cognito. Per proteggersi da attacchi di tipo replay, l'app può analizzare la richiesta `nonce` nel token dell'ID e confrontarlo con quello generato. Per ulteriori informazioni sulla richiesta `nonce`, consulta la sezione relativa alla convalida del token dell'ID nella documentazione dello standard OpenID Connect.

redirect_uri

Reindirizza l'utente alla pagina di accesso per l'autenticazione con il parametro redirect_uri. Imposta il suo valore sull'URL di callback consentito del client dell'app dove vuoi reindirizzare l'utente quando questo ha nuovamente effettuato l'accesso. Utilizza i parametri client_id, scope, state e response_type che vuoi passare all'endpoint `/login`.

Amazon Cognito richiede un parametro logout_uri o redirect_uri nella richiesta all'endpoint `/logout`. Per reindirizzare l'utente all'*`/login`*endpoint per riautenticarsi e passare i token all'app, aggiungi un parametro redirect_uri. Se nella richiesta all'endpoint sono inclusi entrambi i parametri logout_uri e redirect_uri, `/logout` Amazon Cognito sovrascrive il parametro redirect_uri ed elabora esclusivamente il parametro logout_uri.

response_type

La risposta OAuth 2.0 che desideri ricevere da Amazon Cognito dopo l'accesso dell'utente. `code` e `token` sono i valori validi per il parametro response_type.

Obbligatorio se si utilizza un parametro redirect_uri.

stato

Quando l'applicazione aggiunge un parametro di stato a una richiesta, Amazon Cognito ne restituisce il valore all'app quando l'`/oauth2/logout`endpoint reindirizza l'utente.

Aggiungi questo valore alle richieste di protezione contro attacchi CSRF.

Non puoi impostare il valore di un `state` parametro su una stringa JSON. URL-encoded Per passare una stringa che corrisponda a questo formato in un `state` parametro, codificate la stringa in base64, quindi decodificatela nell'applicazione.

Vivamente consigliato quando si utilizza un parametro redirect_uri.

scope

Gli ambiti OAuth 2.0 che desideri richiedere ad Amazon Cognito dopo averli disconnessi con un parametro redirect_uri. Amazon Cognito reindirizza l'utente all'endpoint `/login` con il parametro scopenella richiesta all'endpoint `/logout`.

Facoltativo quando si utilizza un parametro redirect_uri. Se non includi un parametro scope, Amazon Cognito reindirizza l'utente all'endpoint `/login` con un parametro scope. Quando Amazon Cognito reindirizza l'utente e compila automaticamente `scope`, il parametro include tutti gli ambiti autorizzati per il client di app.

Richieste di esempio

Esempio: disconnettersi e reindirizzare l'utente al client

Quando le richieste includono `logout_uri` e `client_id`, Amazon Cognito reindirizza le sessioni utente all'URL nel valore di `logout_uri`, ignorando tutti gli altri parametri di richiesta. Questo URL deve essere un URL di disconnessione autorizzato per il client dell'app.

Di seguito è riportato un esempio di richiesta di disconnessione e reindirizzamento a `https://www.example.com/welcome`.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/logout?
  client_id=1example23456789&
  logout_uri=https%3A%2F%2Fwww.example.com%2Fwelcome
```

Esempio: disconnettersi e richiedere all'utente di accedere come altro utente

Quando le richieste vengono omesse `logout_uri` ma forniscono in altro modo i parametri che costituiscono una richiesta ben formata all'endpoint di autorizzazione, Amazon Cognito reindirizza gli utenti all'accesso gestito. L'endpoint di disconnessione aggiunge i parametri della richiesta originale alla destinazione di reindirizzamento.

I parametri aggiuntivi che aggiungi alla richiesta di disconnessione devono essere presenti nell'elenco in. Parametri della richiesta Ad esempio, l'endpoint di logout non supporta il reindirizzamento IdP automatico con parametri o. `identity_provider` `idp_identifier` Il parametro `redirect_uri` in una richiesta all'endpoint di logout non è un URL di disconnessione, ma un URL post-accesso che si desidera trasmettere all'endpoint di autorizzazione.

Di seguito è riportato un esempio di richiesta che disconnette un utente, reindirizza alla pagina di accesso e fornisce un codice di autorizzazione dopo l'accesso. `https://www.example.com`

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/logout?
  response_type=code&
  client_id=1example23456789&
  redirect_uri=https%3A%2F%2Fwww.example.com&
  state=example-state-value&
  nonce=example-nonce-value&
  scope=openid+profile+aws.cognito.signin.user.admin
```

Endpoint del provider di identità e del relying party

Gli endpoint federativi sono endpoint del pool di utenti che servono a uno degli standard di autenticazione utilizzati dai pool di utenti. Includono URL SAML ACS, endpoint di rilevamento OIDC ed endpoint di servizio per i ruoli del pool di utenti sia come provider di identità che come relying party. Gli endpoint federativi avviano i flussi di autenticazione, ricevono la prova dell'autenticazione dai client ed emettono token ai clienti. IdPs Interagiscono con IdPs le applicazioni e gli amministratori, ma non con gli utenti.

Gli argomenti a pagina intera che seguono questa pagina contengono dettagli sugli endpoint dei provider OAuth 2.0 e OIDC che diventano disponibili quando si aggiunge un dominio al pool di utenti. La tabella seguente è un elenco di tutti gli endpoint della federazione.

Esempi di domini di pool di utenti sono:

1. Dominio con prefisso: `mydomain.auth.us-east-1.amazoncognito.com`
2. Dominio personalizzato: `auth.example.com`

Endpoint di federazione del pool di utenti

URL dell'endpoint	Description	Modalità di accesso
https://*`Your user pool domain`*/oauth2/authorize	Reindirizza un utente all'acces so gestito o all'accesso con il proprio IdP.	Richiamato nel browser del cliente per iniziare l'autenti cazione dell'utente. Consulta Endpoint Authorize.

URL dell'endpoint	Description	Modalità di accesso
https:///*Your user pool domain*oauth2/token	Restituisce i token in base a un codice di autorizzazione o richiesta di credenziali del client.	Richiesto dall'app per recuperare i token. Consulta Endpoint Token.
https:///*Your user pool domain*oauth2/userInfo	Restituisce gli attributi utente in base agli ambiti OAuth 2.0 e all'identità utente in un token di accesso.	Richiesto dall'app per recuperare il profilo utente. Consulta Endpoint UserInfo.
https:///*Your user pool domain*oauth2/revoke	Revoca un token di aggiornamento e i token di accesso associati.	Richiesta dall'app di revocare un token. Consulta Endpoint Revoke.
https://cognito-idp.*Region*.amazonaws.com/*your user pool ID* /.well- -configurazione known/openid	Una directory dell'architettura OIDC del tuo pool di utenti. [1]	Richiesto dall'app per individuare i metadati dell'emittente del pool di utenti.
https://cognito-idp.*Region*.amazonaws.com/ /.well- .json *your user pool ID* known/jwks	Chiavi pubbliche che puoi utilizzare per convalidare i token Amazon Cognito. [2]	Richiesto dall'app per verificare i JWT.
https:///*Your user pool domain*oauth2/idpresponse	I gestori dell'identità digitale devono reindirizzare gli utenti a questo endpoint con un codice di autorizzazione. Amazon Cognito riscatta il codice per un token quando autentica l'utente federato.	Reindirizzato dall'accesso IdP OIDC come URL di callback del client IdP.
https://*Your user pool domain*/saml2/idpresponse	L'URL di Assertion Consumer Response (ACS) per l'integrazione con i provider di identità SAML 2.0.	Reindirizzato da SAML 2.0 IdP come URL ACS o punto di origine per l'accesso. IdP-initiated [3]

URL dell'endpoint	Description	Modalità di accesso
*Your user pool domain*https:///saml2/logout	L'URL Single Logout (SLO) per l'integrazione con i provider di identità SAML 2.0.	Reindirizzato da SAML 2.0 IdP come URL di accesso singolo (SLO). Accetta solo l'associazione POST.

[1] Il `openid-configuration` documento può essere aggiornato in qualsiasi momento con informazioni aggiuntive che mantengano l'endpoint conforme alle specifiche OIDC e OAuth2.

[2] Il file `jwks.json` JSON può essere aggiornato in qualsiasi momento con nuove chiavi pubbliche per la firma dei token.

[3] Per ulteriori informazioni sull'accesso IdP-initiated SAML, consulta. Implementa l' IdP-initiated accesso SAML

Per ulteriori informazioni sugli standard OpenID Connect e OAuth, consultare OpenID Connect 1.0 e OAuth 2.0.

Pool di utenti di Amazon Cognito come emittente OIDC

I pool di utenti di Amazon Cognito funzionano come provider di identità OpenID Connect (OIDC), fungendo da emittente che le librerie di applicazioni possono utilizzare per la federazione OIDC. Le librerie di applicazioni per la federazione OIDC possono fare riferimento ai due diversi percorsi, come illustrato di seguito, come endpoint di autodiscovery. Questo endpoint fornisce l'accesso al JSON Web Key Set (JWKS) `/.well-known/jwks.json` e ai metadati di rilevamento OIDC all'indirizzo`/.well-known/openid-configuration`, dove le applicazioni possono scoprire gli endpoint di autorizzazione, token e UserInfo.

Le applicazioni che supportano l'individuazione automatica OIDC possono configurarsi automaticamente interrogando questi endpoint noti. Per le applicazioni che non supportano l'individuazione automatica, è possibile codificare l'applicazione con gli endpoint OIDC specifici elencati nella sezione precedente.

Tipi di emittenti del pool di utenti

Emittente originale

L'attuale configurazione dell'emittente predefinito per i pool di utenti. L'URL dell'emittente è ospitato nella regione del pool di utenti e fornisce endpoint OIDC specifici per quella regione.

Gli emittenti originali adottano il formato. `https://cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE`

Emittente aggiornato

Consigliato per tutti i pool di utenti, inclusa la replica multiregione. Gli emittenti aggiornati ospitano gli stessi contenuti JWKS in più regioni, con conseguente miglioramento della resilienza e dell'efficienza.

Gli emittenti aggiornati adottano il formato`https://issuer-cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE`, che *Region* è il principale Regione AWS del pool di utenti.

> **Note**
>
> Il tipo di emittente aggiornato non è compatibile con l'integrazione nativa di Amazon Cognito per l'autenticazione `authenticate-cognito` Application Load Balancer (l'azione). Per utilizzare un emittente aggiornato con un Application Load Balancer, configura invece l'integrazione OIDC generica (l'azione) e fornisci `authenticate-oidc` gli endpoint OIDC del tuo pool di utenti. Per ulteriori informazioni, consulta Autenticazione degli utenti tramite Application Load Balancer.

Preparati per le modifiche al documento di scoperta

Il `openid-configuration` documento è un documento di discovery OpenID Connect standard. Amazon Cognito può aggiungere voci di metadati nel tempo per mantenerlo conforme alle specifiche OpenID Connect e OAuth 2.0 e per migliorare la compatibilità con le librerie client OIDC. Queste aggiunte sono retrocompatibili: Amazon Cognito aggiunge nuovi membri, ma non rinomina, modifica il tipo o rimuove i membri già letti dall'applicazione. Scrivi al tuo client in modo che continui a funzionare man mano che il documento cresce.

Ignora i membri che non riconosci

Come richiesto dalla specifica OpenID Connect Discovery, considera facoltativi i membri dei metadati che non comprendi. Non rifiutate, rifiutate o inviate un errore quando il documento contiene un membro che l'applicazione non utilizza.

Non codificate il set o l'ordine dei membri

Analizza il documento come un oggetto JSON generale anziché come una struttura fissa. I membri possono apparire in qualsiasi ordine e il set di membri può cambiare tra le richieste e nel tempo. Indirizza ogni membro per nome e leggi ogni membro come il tipo definito dalla specifica per esso.

Metti alla prova i client rigorosi confrontandoli con il documento corrente

Se il tuo client convalida rigorosamente il documento di scoperta, recupera il documento corrente dall'`/.well-known/openid-configuration`endpoint del tuo pool di utenti e conferma che il client lo accetti prima di dipendere da un insieme specifico di membri.

I token Amazon Cognito seguono lo stesso approccio additivo e compatibile con le versioni future. Per ulteriori informazioni sull'analisi delle attestazioni in un ID o in un token di accesso, consulta. Analizza i token del pool di utenti in modo sicuro

Argomenti

- L'endpoint di reindirizzamento e autorizzazione
- L'endpoint del token
- L'endpoint degli attributi utente
- L'endpoint di revoca del token
- L'endpoint di asserzione SAML IdP

L'endpoint di reindirizzamento e autorizzazione

L'endpoint `/oauth2/authorize` è un endpoint di reindirizzamento che supporta due destinazioni di reindirizzamento. Se si include un `identity_provider` o `idp_identifier` nell'URL, viene eseguito il reindirizzamento invisibile all'utente alla pagina di accesso del gestore dell'identità digitale (IdP) specificato. In caso contrario, viene eseguito il reindirizzamento all'Endpoint Login con gli stessi parametri URL inclusi nella richiesta.

L'endpoint di autorizzazione reindirizza all'accesso gestito o a una pagina di accesso IdP. La destinazione di una sessione utente su questo endpoint è una pagina web con cui l'utente deve interagire direttamente nel proprio browser.

Per utilizzare l'endpoint di autorizzazione, richiama il browser dell'utente all'indirizzo `/oauth2/authorize` con parametri che forniscono al pool di utenti informazioni sui seguenti dettagli del pool stesso.

- Il client dell'app a cui vuoi accedere.
- L'URL di callback a cui desideri essere reindirizzato.
- Gli ambiti OAuth 2.0 che desideri richiedere nel token di accesso dell'utente.
- Facoltativamente, il gestore dell'identità digitale (IdP) di terze parti che desideri utilizzare per accedere.

Puoi anche specificare i parametri `state` e `nonce` utilizzati da Amazon Cognito per convalidare le richieste in arrivo.

`OTTIENI/oauth2/authorize`

L'endpoint `/oauth2/authorize` supporta solo `HTTPS GET`. L'app in genere avvia questa richiesta nel browser dell'utente. È possibile effettuare richieste solo all'endpoint `/oauth2/authorize` tramite HTTPS.

È possibile trovare ulteriori informazioni sulla definizione dell'endpoint di autorizzazione nello standard OpenID Connect (OIDC) nella pagina relativa all'endpoint di autorizzazione.

Parametri della richiesta

`response_type`

Obbligatorio.

Il tipo di risposta. Deve essere `code` o `token`.

Una richiesta riuscita con `response_type` impostato su `code` restituisce una concessione del codice di autorizzazione. Una concessione del codice di autorizzazione è un parametro `code` che Amazon Cognito aggiunge all'URL di reindirizzamento. L'app scambia il codice con l'Endpoint Token per i token di accesso, ID e aggiornamento. Come best practice di sicurezza e per ricevere token di aggiornamento per gli utenti, usa una concessione del codice di autorizzazione nella tua app.

Una richiesta riuscita con `response_type` impostato su `token` restituisce una concessione implicita. Una concessione implicita è un token di ID e accesso che Amazon Cognito aggiunge all'URL di reindirizzamento. Una concessione implicita è meno sicura perché espone i token e le potenziali informazioni di identificazione agli utenti. Puoi disattivare il supporto per le concessioni implicite nella configurazione del client dell'app.

`client_id`

Obbligatorio.

L'ID del client dell'applicazione.

Il valore del parametro `client_id` deve essere l'ID di un client dell'app nel pool di utenti in cui si effettua la richiesta. Il client dell'app deve supportare l'accesso da parte degli utenti locali di Amazon Cognito o di almeno un gestore dell'identità digitale (IdP) di terze parti.

`redirect_uri`

Obbligatorio.

L'URL a cui il server di autenticazione reindirizza il browser una volta che Amazon Cognito concede l'autorizzazione all'utente.

L'URI (Uniform Resource Identifier) di reindirizzamento deve avere i seguenti attributi:

- Deve essere un URI assoluto.
- Deve essere preregistrato con un client.
- Non può includere un componente frammento.

Consulta OAuth 2.0 - Endpoint per il reindirizzamento.

Amazon Cognito richiede che l'URI di reindirizzamento utilizzi HTTPS, ad eccezione di `http://localhost`, che è possibile impostare come URL di callback a scopo di test.

Amazon Cognito supporta anche URL di callback come `myapp://example`.

`state`

Facoltativo, consigliato.

Quando l'app aggiunge un parametro state a una richiesta, Amazon Cognito restituisce il suo valore all'app quando l'endpoint `/oauth2/authorize` reindirizza l'utente.

Aggiungi questo valore alle richieste di protezione contro attacchi CSRF.

Non è possibile impostare il valore di un `state` parametro su una stringa URL-encoded JSON. Per passare una stringa che corrisponda a questo formato in un `state` parametro, codifica la stringa in base64, quindi decodificala nell'app.

`identity_provider`

Opzionale.

Aggiungi questo parametro per ignorare l'accesso gestito e reindirizzare l'utente a una pagina di accesso del provider. Il valore del parametro identity_provider è il nome del provider di identità come appare nel bacino d'utenza.

- Per i provider di servizi sociali, puoi utilizzare i valori identity_provider,, e. `Facebook` `Google` `LoginWithAmazon` `SignInWithApple`
- Per i pool di utenti di Amazon Cognito, usa il valore. `COGNITO`
- Per i provider di identità SAML 2.0 e OpenID Connect (OIDC) (IdPs), usa il nome che hai assegnato all'IdP nel tuo pool di utenti.

`idp_identifier`

Opzionale.

Aggiungi questo parametro per reindirizzare a un provider usando un nome alternativo per il parametro identity_provider. Puoi inserire gli identificatori per SAML 2.0 e OIDC IdPs dal menu Social e provider esterni della console Amazon Cognito.

`scope`

Opzionale.

Può essere una combinazione di qualsiasi ambito riservato al sistema o ambito personalizzato associato al client. Gli ambiti devono essere separati da spazi. Gli ambiti riservati al sistema sono `openid`, `email`, `phone`, `profile` e `aws.cognito.signin.user.admin`. Qualsiasi ambito utilizzato deve essere associato al client o verrà ignorato in fase di runtime.

Se il client non richiede alcun ambito, il server di autenticazione utilizza tutti gli ambiti associati al client.

Un token ID viene restituito solo viene richiesto l'ambito `openid`. Il token di accesso può essere utilizzato in relazione ai bacini d'utenza di Amazon Cognito solo se viene richiesto l'ambito

`aws.cognito.signin.user.admin`. Gli ambiti `phone`, `email` e `profile` possono essere richiesti solo se viene richiesto anche l'ambito `openid`. Questi ambiti impongono le attestazioni da includere all'interno del token ID.

`code_challenge_method`

Opzionale.

Il protocollo di hashing che hai usato per generare la sfida. Il PKCE RFC definisce due metodi, S256 e semplice; tuttavia, il server di autenticazione di Amazon Cognito supporta solo S256.

`code_challenge`

Opzionale.

La sfida Proof of Key Code Exchange (PKCE) che hai generato da. `code_verifier` Per ulteriori informazioni, consulta Utilizzo di PKCE nelle concessioni di codici di autorizzazione.

Obbligatorio solo quando specifichi un parametro `code_challenge_method`.

`nonce`

Opzionale.

Un valore casuale che puoi aggiungere alla richiesta. Il valore nonce fornito è incluso nel token ID emesso da Amazon Cognito. Per proteggersi da attacchi di tipo replay, l'app può analizzare la richiesta `nonce` nel token dell'ID e confrontarlo con quello generato. Per ulteriori informazioni sulla richiesta `nonce`, consulta la sezione relativa alla convalida del token dell'ID nella documentazione dello standard OpenID Connect.

`lang`

Opzionale.

La lingua in cui desideri visualizzare le pagine interattive con l'utente. Le pagine di accesso gestite possono essere localizzate, ma le pagine dell'interfaccia utente ospitate (classiche) no. Per ulteriori informazioni, consulta Managed login localization.

`login_hint`

Opzionale.

Una richiesta di nome utente da trasmettere al server di autorizzazione. Puoi raccogliere un nome utente, un indirizzo e-mail o un numero di telefono dall'utente e consentire al

provider di destinazione di precompilare il nome di accesso dell'utente. Quando invii un `login_hint` parametro e `identity_provider` i parametri no `idp_identifier` or all'`oauth2/authorize`endpoint, l'accesso gestito compila il campo del nome utente con il valore del suggerimento. Puoi anche passare questo parametro a Endpoint Login e inserire automaticamente il valore del nome utente.

Quando la tua richiesta di autorizzazione richiama un reindirizzamento a OIDC, IdPs Amazon Cognito aggiunge `login_hint` un parametro alla richiesta all'autorizzatore di terze parti. Non puoi inoltrare i suggerimenti di accesso a SAML, Apple, Login With Amazon, Google o Facebook (Meta). IdPs

`prompt`

Opzionale.

Un parametro OIDC che controlla il comportamento di autenticazione per le sessioni esistenti. Disponibile solo nella versione Managed Login Branding, non nella classica interfaccia utente ospitata. Per ulteriori informazioni sulla specifica OIDC, consulta Richiesta di autenticazione. I valori `none` `login` hanno un effetto sul comportamento di autenticazione del pool di utenti.

Amazon Cognito inoltra tutti i valori `prompt` tranne `none` IdPs quando gli utenti scelgono l'autenticazione con provider di terze parti. Ciò è vero quando l'URL a cui gli utenti accedono include un `idp_identifier` parametro `identity_provider` o o quando il server di autorizzazione li reindirizza a Endpoint Login e questi selezionano un IdP dai pulsanti disponibili.

Richiedi i valori dei parametri

`prompt=none`

Amazon Cognito continua silenziosamente l'autenticazione per gli utenti che dispongono di una sessione autenticata valida. Con questa richiesta, gli utenti possono autenticarsi silenziosamente tra diversi client di app nel tuo pool di utenti. Se l'utente non è già autenticato, il server di autorizzazione restituisce un errore. `login_required`

`prompt=login`

Amazon Cognito richiede agli utenti di effettuare nuovamente l'autenticazione anche se hanno una sessione esistente. Invia questo valore quando desideri verificare nuovamente l'identità dell'utente. Gli utenti autenticati che hanno una sessione esistente possono tornare all'accesso senza invalidare quella sessione. Quando un utente con una sessione esistente accede nuovamente, Amazon Cognito gli assegna un nuovo cookie di sessione. Questo parametro

può essere inoltrato anche al tuo. IdPs IdPsche accettano questo parametro richiedono anche un nuovo tentativo di autenticazione da parte dell'utente.

`prompt=select_account`

Questo valore non ha effetto sull'accesso locale e deve essere inviato nelle richieste che reindirizzano a. IdPs Se incluso nella richiesta di autorizzazione, questo parametro si aggiunge `prompt=select_account` al percorso URL per la destinazione di reindirizzamento IdP. Quando IdPs supportano questo parametro, richiedono agli utenti di selezionare l'account con cui desiderano accedere.

`prompt=consent`

Questo valore non ha effetto sull'accesso locale e deve essere inviato nelle richieste reindirizzate a. IdPs Se incluso nella richiesta di autorizzazione, questo parametro si aggiunge `prompt=consent` al percorso URL per la destinazione di reindirizzamento IdP. Quando IdPs supportano questo parametro, richiedono il consenso dell'utente prima di reindirizzare nuovamente al tuo pool di utenti.

Quando ometti il `prompt` parametro dalla tua richiesta, l'accesso gestito segue il comportamento predefinito: gli utenti devono accedere a meno che il loro browser non disponga di un cookie valido per la sessione di accesso gestito. Ad esempio, è possibile combinare più valori per `prompt` con un delimitatore di caratteri spaziali. `prompt=login consent`

`resource`

Opzionale.

L'identificatore di una risorsa che desideri associare al token di accesso nel claim. `aud` Quando includi questo parametro, Amazon Cognito verifica che il valore sia un URL e imposta il pubblico del token di accesso risultante sulla risorsa richiesta. Puoi richiedere un server di risorse del pool di utenti con un identificatore in un formato URL o un URL a tua scelta. I valori di questo parametro devono iniziare con `https://http://localhost`, o uno schema URL personalizzato come`myapp://`.

L'associazione delle risorse è definita nella RFC 8707. Per ulteriori informazioni sui server di risorse e sull'associazione delle risorse, vedere Resource binding.

Esempio: concessione del codice di autorizzazione

Questo è un esempio di richiesta di concessione di un codice di autorizzazione.

La seguente richiesta avvia una sessione per recuperare un codice di autorizzazione che l'utente trasmette all'app nella `redirect_uri` destinazione. Questa sessione richiede gli ambiti per gli attributi utente e per l'accesso alle operazioni API self-service di Amazon Cognito.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=openid+profile+aws.cognito.signin.user.admin
```

Il server di autenticazione di Amazon Cognito viene reindirizzato all'app con lo stato e il codice di autorizzazione. Il codice di autorizzazione è valido per cinque minuti.

```
HTTP/1.1 302 Found
Location: https://www.example.com?code=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111&state=abcdefg
```

Esempio: concessione del codice di autorizzazione con PKCE

Questo flusso di esempio esegue una concessione di codice di autorizzazione con PKCE.

Questa richiesta aggiunge un `code_challenge` parametro. Per completare lo scambio di un codice con un token, è necessario includere il `code_verifier` parametro nella richiesta all'`/oauth2/token`endpoint.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=aws.cognito.signin.user.admin&
code_challenge_method=S256&
code_challenge=a1b2c3d4...
```

Il server di autorizzazione reindirizza nuovamente all'applicazione con il codice e lo stato di autorizzazione. L'applicazione elabora il codice di autorizzazione e lo scambia con token.

```
HTTP/1.1 302 Found
```

```
Location: https://www.example.com?code=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111&state=abcdefg
```

Esempio: richiedi la riautenticazione con prompt=login

La richiesta seguente aggiunge un `prompt=login` parametro che richiede all'utente di autenticarsi nuovamente, anche se ha una sessione esistente.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=openid+profile+aws.cognito.signin.user.admin&
prompt=login
```

Il server di autorizzazione reindirizza all'endpoint di accesso, richiedendo una nuova autenticazione.

```
HTTP/1.1 302 Found Location: https://mydomain.auth.us-east-1.amazoncognito.com/
login?response_type=code&client_id=1example23456789&redirect_uri=https://
www.example.com&state=abcdefg&scope=openid+profile
+aws.cognito.signin.user.admin&prompt=login
```

Esempio: autenticazione silenziosa con prompt=none

La richiesta seguente aggiunge un `prompt=none` parametro che verifica silenziosamente se l'utente ha una sessione valida.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=openid+profile+aws.cognito.signin.user.admin&
prompt=none
```

Quando non esiste una sessione valida, il server di autorizzazione restituisce un errore all'URI di reindirizzamento

```
HTTP/1.1 302 Found Location: https://www.example.com?error=login_required&state=abcdefg
```

Quando esiste una sessione valida, il server di autorizzazione restituisce un codice di autorizzazione.

```
HTTP/1.1 302 Found Location: https://www.example.com?
code=AUTHORIZATION_CODE&state=abcdefg
```

Esempio: concessione del codice di autorizzazione con associazione di risorse

La seguente richiesta aggiunge un `resource` parametro per associare il token di accesso a un server di risorse specifico. Il token di accesso risultante crea le condizioni affinché l'API di destinazione confermi che si tratta del pubblico previsto della richiesta dell'utente autenticato.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=solar-system-data-api.example.com/asteroids.add&
resource=https://solar-system-data-api.example.com
```

Il server di autorizzazione restituisce un codice di autorizzazione che genera un token di accesso con un'audattestazione di. `https://solar-system-data-api.example.com`

```
HTTP/1.1 302 Found Location: https://www.example.com?
code=AUTHORIZATION_CODE&state=abcdefg
```

Esempio: concessione di token (implicita) senza ambito **`openid`**

Questo flusso di esempio genera una concessione implicita e restituisce JWT direttamente alla sessione dell'utente.

La richiesta riguarda una concessione implicita dal tuo server di autorizzazione. Richiede gli ambiti nel token di accesso che autorizzano le operazioni self-service del profilo utente.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=token&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=aws.cognito.signin.user.admin
```

Il server di autorizzazione reindirizza all'applicazione solo con un token di accesso. Poiché non è stato richiesto l'ambito del `openid`, Amazon Cognito non restituisce un token ID. Inoltre, Amazon Cognito non restituisce un token di aggiornamento in questo flusso.

```
HTTP/1.1 302 Found
Location: https://example.com/
callback#access_token=eyJra456defEXAMPLE&token_type=bearer&expires_in=3600&state=STATE
```

Esempio: concessione di token (implicita) con ambito openid

Questo flusso di esempio genera una concessione implicita e restituisce token al browser dell'utente.

La richiesta riguarda una concessione implicita dal server di autorizzazione. Richiede degli ambiti nel token di accesso che autorizzano l'accesso agli attributi utente e alle operazioni self-service.

```
GET
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=token&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=aws.cognito.signin.user.admin+openid+profile
```

Il server di autorizzazione reindirizza all'applicazione con il token di accesso e il token ID (poiché `openid` l'ambito era incluso):

```
HTTP/1.1 302 Found
Location: https://
www.example.com#id_token=eyJra67890EXAMPLE&access_token=eyJra12345EXAMPLE&token_type=bearer&exp
```

Esempi di risposte negative

Amazon Cognito potrebbe rifiutare la tua richiesta. Le richieste negative sono accompagnate da un codice di errore HTTP e da una descrizione che puoi utilizzare per correggere i parametri della richiesta. Di seguito sono riportati alcuni esempi di risposte negative.

- Se `client_id` e `redirect_uri` sono validi, ma i parametri della richiesta non sono formattati correttamente, il server di autenticazione reindirizza l'errore al client `redirect_uri` e aggiunge un

messaggio di errore in un parametro URL. Di seguito sono riportati alcuni esempi di formattazione errata.

- La richiesta non include un `response_type` parametro.
- La richiesta di autorizzazione ha fornito un `code_challenge` parametro, ma non un `code_challenge_method` parametro.
- Il valore del `code_challenge_method` parametro non lo è`S256`.

Di seguito è riportata la risposta a una richiesta di esempio con una formattazione errata.

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=invalid_request
```

- Se il client richiede `code` o `token` accede`response_type`, ma non dispone dell'autorizzazione per queste richieste, il server di autorizzazione Amazon Cognito torna `unauthorized_client` a quello del client`redirect_uri`, come segue:

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=unauthorized_client
```

- Se il client richiede un ambito non valido, sconosciuto o non corretto, il server di autorizzazione di Amazon Cognito restituisce `invalid_scope` al `redirect_uri` del client, come riportato di seguito:

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=invalid_scope
```

- Se si verifica un errore imprevisto nel server, il server di autenticazione torna `server_error` a quello del `redirect_uri` client. Poiché l'errore HTTP 500 non viene inviato al client, l'errore non viene visualizzato nel browser dell'utente. Il server di autorizzazione restituisce il seguente errore.

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=server_error
```

- Quando Amazon Cognito esegue l'autenticazione tramite federazione a terze parti, Amazon IdPs Cognito potrebbe riscontrare problemi di connessione, come i seguenti:
  - Se si verifica un timeout di connessione durante la richiesta del token dal provider di identità, il server di autenticazione reindirizza l'errore al `redirect_uri` del client come segue:

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=Timeout+occurred+in+calling+IdP+token
+endpoint
```

  - Se si verifica un timeout di connessione durante la chiamata all'`jwks_uri`endpoint per la convalida del token ID, il server di autenticazione reindirizza con un errore al client nel modo seguente: `redirect_uri`

    ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=error_description=Timeout+in+calling+jwks
    +uri
    ```

- Durante l'autenticazione mediante federazione a terze parti IdPs, i provider possono restituire risposte di errore. Ciò può essere dovuto a errori di configurazione o ad altri motivi, come i seguenti:
  - Se viene ricevuta una risposta di errore da altri provider, il server di autenticazione reindirizza l'errore al `redirect_uri` del client come segue:

    ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=[IdP name]+Error+-+[status code]+error
     getting token
    ```

  - Se viene ricevuta una risposta di errore da Google, il server di autenticazione reindirizza l'errore al `redirect_uri` del client come segue:

    ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=Google+Error+-+[status code]+[Google-
    provided error code]
    ```

- Quando Amazon Cognito rileva un'eccezione di comunicazione quando si connette a un IdP esterno, il server di autenticazione reindirizza con un errore al client con uno dei seguenti messaggi: `redirect_uri`
  - ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=Connection+reset
    ```
  - ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=Read+timed+out
    ```

Esempio: richiesta di autenticazione Step-up

Amazon Cognito supporta l'autenticazione avanzata con accesso gestito. Non è supportata nell'interfaccia utente ospitata. Se l'applicazione si basa su ACR e AMR, utilizza l'accesso gestito.

Per ulteriori informazioni sui livelli ACR e sui valori AMR, vedere. the section called “Livelli di autenticazione con attestazioni ACR e AMR”

L'`/oauth2/authorize`endpoint accetta i seguenti parametri URL per l'autenticazione avanzata.

`acr_values`

Un URL-encoded elenco separato da spazi degli URI del livello di autenticazione ACR di destinazione, dal più alto al più basso. Amazon Cognito valuta l'elenco da sinistra a destra, applica il primo livello valido che l'utente può soddisfare e ignora i valori che non riconosce. Questo parametro si comporta come il parametro API. `TARGET_ACR_VALUES` Richiede il piano di funzionalità Essentials o Plus.

`max_age`

Un numero intero non negativo che specifica il numero massimo di secondi trascorsi dall'ultima autenticazione dell'utente. Se il valore dell'ora corrente in meno `auth_time` è maggiore di`max_age`, l'utente deve eseguire nuovamente l'autenticazione da zero anziché limitarsi a procedere. Questo comportamento è identico al parametro API. `MAX_AGE`

Combinalo `acr_values` con `max_age` per richiedere un livello di autenticazione specifico con l'autenticazione recente per un'operazione sensibile.

La seguente `/oauth2/authorize` richiesta di esempio chiede ad Amazon Cognito di provare prima il livello 4. Se l'utente non riesce a soddisfare il livello 4, ad esempio perché non ha configurato un TOTP, Amazon Cognito torna al livello 3, supponendo che l'utente abbia configurato una password monouso per SMS o e-mail. La richiesta richiede inoltre che l'utente si sia autenticato negli ultimi 3.600 secondi. L'ID client di esempio *`1example23456789`* e gli altri valori in corsivo sono segnaposto. Sostituisci questi valori con i tuoi valori.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=1example23456789&redirect_uri=https://YOUR_APP/
redirect_uri&state=STATE&scope=openid+profile
+aws.cognito.signin.user.admin&acr_values=urn%3Acognito%3Aloa%3A4+urn%3Acognito%3Aloa
%3A3&max_age=3600
```

Il comportamento di un piano di funzionalità insufficiente è diverso da quello dell'API. L'endpoint di autorizzazione all'accesso gestito ignora silenziosamente `acr_values` e procede con la normale autenticazione, senza errori. L'API restituisce un. `FeatureUnavailableInTierException` Per ulteriori informazioni, consulta Requisiti del piano di funzionalità.

L'endpoint del token

L'endpoint del token OAuth 2.0 `/oauth2/token` emette token web JSON (JWT) per le applicazioni che desiderano completare i flussi di concessione del codice di autorizzazione e delle credenziali del client. Questi token sono il risultato finale dell'autenticazione con un pool di utenti. Contengono informazioni sull'utente (token ID), sul livello di accesso dell'utente (token di accesso) e sul diritto dell'utente a mantenere la sessione di accesso (token di aggiornamento). Le librerie relying-party OpenID Connect (OIDC) gestiscono le richieste e i payload di risposta da questo endpoint. I token forniscono una prova verificabile dell'autenticazione, informazioni sul profilo e un meccanismo per l'accesso ai sistemi di back-end.

L'endpoint del token restituisce un'intestazione di risposta. `Access-Control-Allow-Origin: *` È possibile chiamare l'endpoint del token cross-origin da un'applicazione basata su browser. Ad esempio, è possibile completare la concessione di un codice di autorizzazione con Proof Key for Code Exchange (PKCE) di un client pubblico. Amazon Cognito non supporta politiche di origine CORS (Cross-Origin Resource Sharing) personalizzate su questo endpoint. Per ulteriori informazioni, consulta la sezione sulle politiche CORS di. Accesso gestito dal pool di utenti

Il server di autorizzazione OAuth 2.0 del pool di utenti emette token web JSON (JWT) dall'endpoint del token ai seguenti tipi di sessioni:

1. Utenti che hanno completato una richiesta di concessione del codice di autorizzazione. Il riscatto riuscito di un codice restituisce i token ID, di accesso e di aggiornamento.
2. Machine-to-machine (M2M) sessioni che hanno completato una concessione di credenziali client. L'autorizzazione riuscita con il segreto del client restituisce un token di accesso.
3. Utenti che hanno precedentemente effettuato l'accesso e ricevuto token di aggiornamento. L'autenticazione con token di aggiornamento restituisce nuovi ID e token di accesso.

> **Note**
>
> Gli utenti che accedono con un codice di autorizzazione concesso nell'accesso gestito o tramite la federazione possono sempre aggiornare i propri token dall'endpoint del token. Gli utenti che accedono con le operazioni API `InitiateAuth` e `AdminInitiateAuth` possono aggiornare i propri token con l'endpoint token quando i dispositivi ricordati non sono attivi nel pool di utenti. Se i dispositivi ricordati sono attivi, aggiorna i token con l'API o l'operazione di aggiornamento dei token SDK pertinenti per il client dell'app.

L'endpoint token diventa disponibile pubblicamente quando si aggiunge un dominio al pool di utenti. Accetta le richieste POST HTTP. Per la sicurezza delle applicazioni, utilizzate PKCE con gli eventi di accesso con il codice di autorizzazione. PKCE verifica che l'utente che trasmette un codice di autorizzazione sia lo stesso utente che ha effettuato l'autenticazione. Per ulteriori informazioni su PKCE, vedere IETF RFC 7636.

Puoi saperne di più sui client dell'app del pool di utenti e sui relativi tipi di concessione, segreti dei client, ambiti consentiti e ID client all'indirizzo. Application-specific impostazioni con i client dell'app Puoi saperne di più sull'autorizzazione M2M, sulla concessione delle credenziali dei client e sull'autorizzazione con ambiti dei token di accesso all'indirizzo. Ambiti, M2M e server di risorse

Per recuperare informazioni su un utente dal suo token di accesso, passale alla tua richiesta Endpoint UserInfo o a una richiesta API. GetUser Il token di accesso deve contenere gli ambiti appropriati per queste richieste.

In che modo Amazon Cognito si differenzia dalle specifiche OIDC

Amazon Cognito non implementa il modello di `offline_access` ambito OpenID Connect per il controllo dell'emissione di refresh-token. Invece, l'endpoint del token restituisce un token di aggiornamento ogni volta che lo `grant_type` è `authorization_code` e tu controlli la durata del token di refresh tramite l'impostazione di validità del refresh-token del client dell'app anziché tramite richiesta. `offline_access` La richiesta dell'`offline_access`ambito non ha alcun effetto e la sua omissione non sopprime il token di aggiornamento.

Formatta una richiesta POST sull'endpoint del token

L'endpoint `/oauth2/token` supporta solo `HTTPS POST`. Questo endpoint non è interattivo con l'utente. Gestisci le richieste di token con una libreria OpenID Connect (OIDC) nella tua applicazione.

L'endpoint token supporta l'autenticazione `client_secret_basic` e `client_secret_post`. Per ulteriori informazioni sulla specifica OIDC, vedere Client Authentication. Per ulteriori informazioni sull'endpoint Token della specifica OpenID Connect, consulta Endpoint Token.

Parametri della richiesta nell'intestazione

È possibile passare i seguenti parametri nell'intestazione della richiesta all'endpoint del token.

`Authorization`

Se il client ha emesso un segreto, dovrà trasmettere i suoi `client_id` e `client_secret` nell'intestazione dell'autorizzazione come autorizzazione HTTP `client_secret_basic`. Puoi

inoltre includere `client_id` e `client_secret` nel corpo della richiesta come autorizzazione `client_secret_post`.

La stringa di intestazione di autorizzazione è Base `Base64Encode(client_id:client_secret)`. L'esempio seguente è un'intestazione di autorizzazione per il client dell'app `djc98u3jiedmi283eu928` con client secret`abcdef01234567890`, che utilizza la Base64-encoded versione della stringa: `djc98u3jiedmi283eu928:abcdef01234567890`

```
Authorization: Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw
```

`Content-Type`

Imposta il valore di questo parametro su `'application/x-www-form-urlencoded'`.

Parametri della richiesta nel corpo

Di seguito sono riportati i parametri che è possibile richiedere in `x-www-form-urlencoded` formato nel corpo della richiesta all'endpoint del token.

`grant_type`

Campo obbligatorio.

Il tipo di concessione OIDC che desideri richiedere.

Deve essere `authorization_code` o `refresh_token` o `client_credentials`. Puoi richiedere un token di accesso per un ambito personalizzato dall'endpoint del token alle seguenti condizioni:

- Hai abilitato l'ambito richiesto nella configurazione del client dell'app.
- Hai configurato il client dell'app con un client secret.
- Hai abilitato il client dell'app per la concessione delle credenziali del client. È possibile abilitare questa concessione in uno dei seguenti modi:
  - Per l'autenticazione machine-to-machine (M2M) nel dominio ospitato, `client_credentials` aggiungila al client dell'app. `AllowedOAuthFlows`
  - Per l'autenticazione M2M senza dominio, aggiungila al client dell'app. `ALLOW_CLIENT_TOKEN_AUTH` `ExplicitAuthFlows` Questo flusso richiede un client secret e deve essere l'unico flusso di autenticazione del client dell'app.

Se non abiliti nessuna di queste opzioni, l'endpoint del token restituisce un errore di classe non autorizzata.

> **Note**
>
> L'endpoint del token restituisce un token di aggiornamento solo quando lo è. `grant_type` `authorization_code`

`client_id`

Facoltativo. Non richiesto quando si fornisce l'ID client dell'app nell'*`Authorization`*intestazione.

L'ID di un client app nel tuo pool di utenti. Specificate lo stesso client dell'app che ha autenticato l'utente.

È necessario fornire questo parametro se il client è pubblico e non dispone di un segreto o è `client_secret_post` autorizzato. `client_secret`

`client_secret`

Facoltativo. Non richiesto quando si fornisce il segreto del client nell'*`Authorization`*intestazione e quando il client dell'app non dispone di un segreto.

Il segreto del client dell'app, se il client dell'app ne ha uno, per `client_secret_post` l'autorizzazione.

`scope`

Facoltativo.

Può essere una combinazione di qualsiasi ambito associato al client dell'app. Se la richiesta include un ambito definito su un server di risorse, Amazon Cognito verifica se tale ambito è consentito per il client dell'app richiesto. Se l'ambito non è consentito, Amazon Cognito rifiuta l'intera richiesta anziché emettere un token per gli ambiti rimanenti. Una richiesta che combina ambiti consentiti e non consentiti ha esito negativo per lo stesso motivo.

Amazon Cognito rifiuta un ambito inutilizzabile anziché eliminarlo e restituire un token per il resto. Una richiesta rifiutata viene restituita `HTTP 400` con un errore nel corpo della risposta. Il valore di errore dipende dal motivo per cui l'ambito era inutilizzabile. Quando esiste un ambito ma non è consentito per il client dell'app, viene restituito Amazon Cognito. `invalid_grant` Quando un ambito si riferisce a un server di risorse o a un ambito personalizzato che non esiste,

Amazon Cognito ritorna. `invalid_scope` Se non fornisci questo parametro di richiesta, il server di autorizzazione restituisce una `scope` richiesta di token di accesso con tutti gli ambiti di autorizzazione abilitati nella configurazione del client dell'app. Puoi richiedere uno qualsiasi degli ambiti consentiti per il client dell'app richiesto: ambiti standard, ambiti personalizzati dai server di risorse e ambito self-service dell'`aws.cognito.signin.user.admin`utente.

`redirect_uri`

Facoltativo. Non richiesto per la concessione delle credenziali del client.

Deve essere lo stesso `redirect_uri` utilizzato per ottenere `authorization_code` in `/oauth2/authorize`.

È necessario fornire questo parametro se lo è. `grant_type` `authorization_code`

`refresh_token`

Facoltativo. Utilizzato solo quando l'utente dispone già di un token di aggiornamento e necessita di nuovi ID e token di accesso.

Per generare nuovi token di accesso e ID per la sessione di un utente, imposta il valore di su un token `refresh_token` di aggiornamento valido emesso dal client dell'app richiesto.

Restituisce un nuovo token di aggiornamento con un nuovo ID e token di accesso quando la rotazione del token di aggiornamento è attiva, altrimenti restituisce solo ID e token di accesso. Se il token di accesso originale era associato a una risorsa API, il nuovo token di accesso mantiene l'URL dell'API richiesto nel claim. `aud`

`code`

Facoltativo. Richiesto solo nelle concessioni di codici di autorizzazione.

Il codice di autorizzazione derivante da una concessione di codice di autorizzazione. È necessario fornire questo parametro se la richiesta di autorizzazione includeva un `grant_type` di`authorization_code`.

`aws_client_metadata`

Facoltativo.

Informazioni da trasmettere ai flussi di autorizzazione da macchina Trigger Lambda di pre-generazione del token a macchina (M2M). L'applicazione può raccogliere informazioni contestuali sulla sessione e passarle in questo parametro. Quando esegui il passaggio `aws_client_metadata` in formato URL-encoded JSON, Amazon Cognito lo include nell'evento

di input della funzione Lambda trigger. La versione dell'evento pre token trigger o la versione globale del trigger Lambda deve essere configurata per la versione tre o successiva. Sebbene Amazon Cognito accetti le richieste a questo endpoint nei flussi M2M del codice di autorizzazione e delle credenziali client, il tuo pool di utenti passa `aws_client_metadata` al trigger precedente alla generazione del token solo dalle richieste di credenziali del client.

`code_verifier`

Opzionale. Richiesto solo se hai fornito i `code_challenge` parametri nella richiesta `code_challenge_method` di autorizzazione iniziale.

Il verificatore di codice generato `code_challenge` da cui l'applicazione ha calcolato in una richiesta di concessione del codice di autorizzazione con PKCE.

Sostituzione di un codice di autorizzazione per i token

La seguente richiesta genera correttamente ID, token di accesso e aggiornamento dopo l'autenticazione con una concessione del codice di autorizzazione. La richiesta trasmette il segreto del client nel formato dell'intestazione. `client_secret_basic` `Authorization`

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token&
Content-Type='application/x-www-form-urlencoded'&
Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

grant_type=authorization_code&
client_id=1example23456789&
code=AUTHORIZATION_CODE&
redirect_uri=com.myclientapp://myclient/redirect
```

La risposta invia all'utente nuovi ID, token di accesso e aggiornamento, con metadati aggiuntivi.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "eyJra1example",
    "id_token": "eyJra2example",
    "refresh_token": "eyJj3example",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Credenziali del client con autorizzazione di base

La seguente richiesta proveniente da un'applicazione M2M richiede la concessione delle credenziali del client. Poiché le credenziali del client richiedono un client secret, la richiesta viene autorizzata con un'`Authorization`intestazione derivata dall'ID e dal segreto del client dell'app. La richiesta genera un token di accesso con i due ambiti richiesti. La richiesta include anche i metadati del client che forniscono IP-address informazioni e un token rilasciato all'utente di cui questa concessione è per conto. Amazon Cognito passa i metadati del client al trigger Lambda prima della generazione di token.

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token >
Content-Type='application/x-www-form-urlencoded'&
Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

grant_type=client_credentials&
client_id=1example23456789&
scope=resourceServerIdentifier1%2Fscope1%20resourceServerIdentifier2%2Fscope2&
&aws_client_metadata=%7B%22onBehalfOfToken%22%3A%22eyJra789ghiEXAMPLE%22,
%20%22ClientIpAddress%22%3A%22192.0.2.252%22%7D
```

Amazon Cognito trasmette il seguente evento di input al trigger Lambda prima della generazione del token.

```
{
    version: '3',
    triggerSource: 'TokenGeneration_ClientCredentials',
    region: 'us-east-1',
    userPoolId: 'us-east-1_EXAMPLE',
    userName: 'ClientCredentials',
    callerContext: {
        awsSdkVersion: 'aws-sdk-unknown-unknown',
        clientId: '1example23456789'
    },
    request: {
        userAttributes: {},
        groupConfiguration: null,
        scopes: [
           'resourceServerIdentifier1/scope1',
           'resourceServerIdentifier2/scope2'
        ],
        clientMetadata: {
            'onBehalfOfToken': 'eyJra789ghiEXAMPLE',
```

```
            'ClientIpAddress': '192.0.2.252'
        }
    },
    response: { claimsAndScopeOverrideDetails: null }
}
```

La risposta restituisce un token di accesso. Le credenziali concesse ai client servono per l'autorizzazione da macchina a macchina (M2M) e restituiscono solo token di accesso.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "access_token": "eyJra1example",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Credenziali del client con autorizzazione dell'ente POST

La seguente richiesta di concessione di credenziali client include il `client_secret` parametro nel corpo della richiesta e non include un'intestazione. `Authorization` Questa richiesta utilizza la sintassi di autorizzazione. `client_secret_post` La richiesta genera un token di accesso con l'ambito richiesto. La richiesta include anche i metadati del client che forniscono IP-address informazioni e un token rilasciato all'utente di cui questa concessione è per conto. Amazon Cognito passa i metadati del client al trigger Lambda prima della generazione di token.

```
POST /oauth2/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
X-Amz-Target: AWSCognitoIdentityProviderService.Client credentials request
User-Agent: USER_AGENT
Accept: /
Accept-Encoding: gzip, deflate, br
Content-Length: 177
Referer: http://auth.example.com/oauth2/token
Host: auth.example.com
Connection: keep-alive

grant_type=client_credentials&
client_id=1example23456789&
scope=my_resource_server_identifier%2Fmy_custom_scope&
```

```
client_secret=9example87654321&
aws_client_metadata=%7B%22onBehalfOfToken%22%3A%22eyJra789ghiEXAMPLE%22,
%20%22ClientIpAddress%22%3A%22192.0.2.252%22%7D
```

Amazon Cognito trasmette il seguente evento di input al trigger Lambda prima della generazione del token.

```
{
    version: '3',
    triggerSource: 'TokenGeneration_ClientCredentials',
    region: 'us-east-1',
    userPoolId: 'us-east-1_EXAMPLE',
    userName: 'ClientCredentials',
    callerContext: {
        awsSdkVersion: 'aws-sdk-unknown-unknown',
        clientId: '1example23456789'
    },
    request: {
        userAttributes: {},
        groupConfiguration: null,
        scopes: [
           'resourceServerIdentifier1/my_custom_scope'
        ],
        clientMetadata: {
            'onBehalfOfToken': 'eyJra789ghiEXAMPLE',
            'ClientIpAddress': '192.0.2.252'
        }
    },
    response: { claimsAndScopeOverrideDetails: null }
}
```

La risposta restituisce un token di accesso. Le credenziali concesse ai client servono per l'autorizzazione da macchina a macchina (M2M) e restituiscono solo token di accesso.

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Tue, 05 Dec 2023 16:11:11 GMT
x-amz-cognito-request-id: 829f4fe2-a1ee-476e-b834-5cd85c03373b

{
    "access_token": "eyJra12345EXAMPLE",
    "expires_in": 3600,
```

```
    "token_type": "Bearer"
}
```

Concessione del codice di autorizzazione con PKCE

La seguente richiesta di esempio completa una richiesta di autorizzazione che includeva `code_challenge_methodcode_challenge` parametri in una richiesta di concessione del codice di autorizzazione con PKCE.

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token
Content-Type='application/x-www-form-urlencoded'&
Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

grant_type=authorization_code&
client_id=1example23456789&
code=AUTHORIZATION_CODE&
code_verifier=CODE_VERIFIER&
redirect_uri=com.myclientapp://myclient/redirect
```

La risposta restituisce ID, accesso e token di aggiornamento derivanti dall'esito positivo della verifica PKCE da parte dell'applicazione.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "eyJra1example",
    "id_token": "eyJra2example",
    "refresh_token": "eyJj3example",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Aggiornamento del token senza aggiornamento (rotazione del token)

La seguente richiesta di esempio fornisce un token di aggiornamento a un client di app in cui la rotazione del token di aggiornamento è inattiva. Poiché il client dell'app dispone di un client secret, la richiesta fornisce un'intestazione. `Authorization`

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token >
Content-Type='application/x-www-form-urlencoded'&
```

```
Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

grant_type=refresh_token&
client_id=1example23456789&
refresh_token=eyJj3example
```

La risposta restituisce nuovi ID e token di accesso.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "eyJra1example",
    "id_token": "eyJra2example",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Aggiornamento dei token con rotazione dei token di aggiornamento

La seguente richiesta di esempio fornisce un token di aggiornamento a un client di app in cui è attiva la rotazione del token di aggiornamento. Poiché il client dell'app dispone di un client secret, la richiesta fornisce un'`Authorization`intestazione.

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token >
Content-Type='application/x-www-form-urlencoded'&
Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

grant_type=refresh_token&
client_id=1example23456789&
refresh_token=eyJj3example
```

La risposta restituisce nuovi ID, token di accesso e aggiornamento.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "eyJra1example",
    "id_token": "eyJra2example",
    "refresh_token": "eyJj4example",
```

```
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Esempi di risposte negative

Le richieste non valide generano errori dall'endpoint del token. Di seguito è riportata una mappa generale del corpo della risposta quando le richieste di token generano un errore.

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error":"invalid_request|invalid_client|invalid_grant|unauthorized_client|
unsupported_grant_type"
}
```

`invalid_request`

Nella richiesta manca un parametro obbligatorio, include un valore di parametro non supportato (diverso da `unsupported_grant_type`) oppure è comunque non corretto. Ad esempio, `grant_type` è `refresh_token` ma `refresh_token` non è incluso.

`invalid_client`

L'autenticazione del client non è riuscita. Ad esempio, quando il client include `client_id` e `client_secret` nell'intestazione di autorizzazione, ma non esiste un client con valori `client_id` e `client_secret`.

`invalid_grant`

Il token di aggiornamento è stato revocato.

Il codice di autorizzazione è stato utilizzato già o non esiste.

Il client dell'app non ha accesso in lettura a tutti gli attributi nell'ambito richiesto. Ad esempio, l'app richiede l'ambito `email` e il client dell'app può leggere l'attributo `email`, ma non `email_verified`.

`unauthorized_client`

Il client non ha le autorizzazioni necessarie per il flusso di concessione del codice o per l'aggiornamento dei token.

Una richiesta di token con un valore `redirect_uri` che non corrisponde al valore della richiesta di autorizzazione viene restituita `unauthorized_client` con un `error_description` di`invalid_redirect`.

`unsupported_grant_type`

Restituito se `grant_type` è diverso da `authorization_code`, `refresh_token` o `client_credentials`.

L'endpoint degli attributi utente

Laddove OIDC emette token ID che contengono attributi utente, OAuth 2.0 implementa l'endpoint. `/oauth2/userInfo` Un utente o cliente autenticato riceve un token di accesso con un attestato. `scopes` Questa affermazione determina gli attributi che il server di autorizzazione deve restituire. Quando un'applicazione presenta un token di accesso all'`userInfo`endpoint, il server di autorizzazione restituisce un corpo di risposta che contiene gli attributi utente che rientrano nei limiti stabiliti dagli ambiti del token di accesso. L'applicazione può recuperare informazioni su un utente dall'`userInfo`endpoint purché disponga di un token di accesso valido con almeno un requisito di ambito. `openid`

L'endpoint `userInfo` è un endpoint userInfo OpenID Connect (OIDC). Risponde con gli attributi utente quando i provider di servizi presentano i token di accesso emessi dall'endpoint del token. Gli ambiti nel token di accesso dell'utente definiscono gli attributi utente restituiti dall'endpoint userInfo nella sua risposta. L'ambito `openid` deve essere una delle richieste del token di accesso.

Amazon Cognito emette token di accesso in risposta a richieste API dei pool di utenti come InitiateAuth. Poiché non contengono alcun ambito, l'endpoint userInfo non accetta questi token di accesso. È invece necessario presentare i token di accesso dall'endpoint del token.

Il gestore dell'identità digitale (IdP) di terze parti OAuth 2.0 ospita anche un endpoint userInfo. Quando l'utente si autentica con quell'IdP, Amazon Cognito scambia silenziosamente un codice di autorizzazione con l'endpoint IdP. `token` Il tuo pool di utenti passa il token di accesso IdP per autorizzare il recupero delle informazioni utente dall'endpoint IdP. `userInfo`

Gli ambiti nel token di accesso di un utente sono determinati dal parametro `scopes` request nelle richieste di autenticazione o dagli ambiti aggiunti dal trigger Lambda prima della generazione del token. È possibile decodificare i token di accesso ed esaminare le `scope` attestazioni per vedere gli ambiti di controllo degli accessi in essi contenuti. Di seguito sono riportate alcune combinazioni di

ambiti che influiscono sui dati restituiti dall'endpoint. `userInfo` L'ambito riservato di Amazon Cognito non `aws.cognito.signin.user.admin` ha alcun effetto sui dati restituiti da questo endpoint.

Esempi di ambiti nel token di accesso e loro effetto sulla risposta **`UserInfo`**

`openid`

Restituisce una risposta con tutti gli attributi utente che il client dell'app è in grado di leggere.

`openid profile`

Restituisce gli attributi utente `name` `family_name`,`given_name`,`middle_name`,`nickname`,`preferred_username`,`profile`,`picture`,`web` e`updated_at`. Restituisce anche attributi personalizzati. Nei client di app che non dispongono dell'accesso in lettura a ciascun attributo, la risposta a questo ambito è costituita da tutti gli attributi all'interno delle specifiche a cui il client dell'app ha accesso in lettura.

`openid email`

Restituisce le informazioni di base del profilo `email` e `email_verified` gli attributi and.

`openid phone`

Restituisce le informazioni di base del profilo `phone_number` e `phone_number_verified` gli attributi and.

OTTIENI/oauth2/userInfo

L'applicazione genera richieste a questo endpoint direttamente, non tramite un browser.

Per ulteriori informazioni, consulta Endpoint UserInfo nelle specifiche di OpenID Connect (OIDC).

Argomenti

- Parametri della richiesta nell'intestazione
- Esempio: richiesta
- Esempio: risposta positiva
- Esempio di risposte negative

Parametri della richiesta nell'intestazione

`Authorization: Bearer <access_token>`

Passa il token di accesso nel campo dell'intestazione dell'autorizzazione.

Obbligatorio.

Esempio: richiesta

```
GET /oauth2/userInfo HTTP/1.1
Content-Type: application/x-amz-json-1.1
Authorization: Bearer eyJra12345EXAMPLE
User-Agent: [User agent]
Accept: */*
Host: auth.example.com
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Esempio: risposta positiva

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: [Integer]
Date: [Timestamp]
x-amz-cognito-request-id: [UUID]
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Server: Server
Connection: keep-alive
{
    "sub": "[UUID]",
    "email_verified": "true",
    "custom:mycustom1": "CustomValue",
    "phone_number_verified": "true",
    "phone_number": "+12065551212",
    "email": "bob@example.com",
```

```
    "username": "bob"
}
```

Per un elenco di richieste OIDC, vedi la sezione relativa alle richieste standard. Attualmente, Amazon Cognito restituisce i valori per `email_verified` e `phone_number_verified` come stringhe.

Esempio di risposte negative

Esempio: richiesta errata

```
HTTP/1.1 400 Bad Request
WWW-Authenticate: error="invalid_request",
error_description="Bad OAuth2 request at UserInfo Endpoint"
```

`invalid_request`

Nella richiesta manca un parametro obbligatorio, include un valore di parametro non supportato o è altrimenti non valida.

Esempio: token errato

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: error="invalid_token",
error_description="Access token is expired, disabled, or deleted, or the user has
 globally signed out."
```

`invalid_token`

Il token di accesso è scaduto, revocato, non valido o non è valido.

L'endpoint di revoca del token

Gli utenti che possiedono un token di aggiornamento nella loro sessione hanno qualcosa di simile a un cookie del browser. Possono rinnovare la sessione esistente purché il token di aggiornamento sia valido. Invece di richiedere a un utente di accedere dopo la scadenza dell'ID o del token di accesso, l'applicazione può utilizzare il token di aggiornamento per ottenere nuovi token validi. Tuttavia, è possibile stabilire esternamente che la sessione di un utente debba essere terminata oppure l'utente potrebbe scegliere di dimenticare la sessione corrente. A quel punto, puoi revocare quel token di aggiornamento in modo che l'utente non possa più persistere nella sessione.

L'`/oauth2/revoke`endpoint revoca il token di accesso di un utente che Amazon Cognito aveva inizialmente emesso con il token di aggiornamento fornito. Questo endpoint revoca anche il token di aggiornamento stesso e tutti i token di accesso e di identità successivi dello stesso token di aggiornamento. Dopo che l'endpoint revoca i token, non puoi utilizzare i token di accesso revocati per accedere alle API autenticate con i token Amazon Cognito.

PUBBLICAZIONE/oauth2/revoke

L'endpoint `/oauth2/revoke` supporta solo `HTTPS POST`. Il client del bacino d'utenza effettua direttamente le richieste a questo endpoint e non tramite il browser di sistema.

Parametri della richiesta nell'intestazione

`Authorization`

Se il client dell'app dispone di un client secret, l'applicazione deve passare il relativo codice `client_id` `client_secret` nell'intestazione di autorizzazione tramite l'autorizzazione HTTP di base. Il segreto è Basic `Base64Encode(client_id:client_secret)`.

`Content-Type`

Deve sempre essere `'application/x-www-form-urlencoded'`.

Parametri della richiesta nel corpo

`token`

(Obbligatorio) Il token di aggiornamento che il client desidera revocare. La richiesta revoca anche tutti i token di accesso emessi da Amazon Cognito con questo token di aggiornamento.

Obbligatorio.

`client_id`

(Facoltativo) L'ID client dell'app per il token che desideri revocare.

Obbligatorio se il client è pubblico e non dispone di un segreto.

Esempi di richiesta di revoca

Questa richiesta di revoca revoca un token di aggiornamento per un client di app che non ha alcun segreto client. Nota il `client_id` parametro nel corpo della richiesta.

```
POST /oauth2/revoke HTTP/1.1
Host: mydomain.auth.us-east-1.amazoncognito.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
token=2YotnFZFEjr1zCsicMWpAA&
client_id=1example23456789
```

Questa richiesta di revoca revoca un token di aggiornamento per un client di app con un client segreto. Nota l'`Authorization`intestazione che contiene un ID client codificato e un client secret, ma non nel corpo della richiesta. `client_id`

```
POST /oauth2/revoke HTTP/1.1
Host: mydomain.auth.us-east-1.amazoncognito.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
token=2YotnFZFEjr1zCsicMWpAA
```

Risposta di errore di revoca

Una risposta riuscita contiene un corpo vuoto. La risposta di errore è un oggetto JSON con un campo `error` e possibilmente un campo `error_description`.

Errori di endpoint

- Se il token non è presente nella richiesta o se la funzionalità è disabilitata per il client dell'app, viene restituito un codice di errore HTTP 400 e `invalid_request`.
- Se il token inviato da Amazon Cognito nella richiesta di revoca non è un token di aggiornamento, riceverai un codice di errore HTTP 400 e `unsupported_token_type`.
- Se le credenziali client non sono valide, riceverai un codice di errore HTTP 401 e `invalid_client`.
- Se il token è stato revocato o se il client ha inviato un token non valido, viene restituito un codice di errore HTTP 200 OK.

L'endpoint di asserzione SAML IdP

`/saml2/idpresponse`riceve le asserzioni SAML. Nell'accesso avviato dal fornitore di servizi (SP-initiated), l'applicazione non interagisce direttamente con questo endpoint: il provider di identità (IdP) SAML 2.0 reindirizza l'utente qui con la sua risposta SAML. Per SP-initiated l'accesso, configura il tuo IdP con il percorso del `saml2/idpresponse` tuo URL As the Assertion Consumer Service (ACS). Per ulteriori informazioni sull'avvio della sessione, consulta. Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito

Durante IdP-initiated l'accesso, richiama le richieste a questo endpoint nell'applicazione dopo aver effettuato l'accesso come utente con il tuo provider SAML 2.0. I tuoi utenti accedono con il tuo IdP nel loro browser, quindi l'applicazione raccoglie l'asserzione SAML e la invia a questo endpoint. È necessario inviare le asserzioni SAML nel corpo di una richiesta tramite HTTPS. `HTTP POST` Il corpo della `POST` richiesta deve essere un `SAMLResponse` parametro e un `Relaystate` parametro. Per ulteriori informazioni, consulta Implementa l' IdP-initiated accesso SAML.

L'`saml2/idpresponse`endpoint può accettare asserzioni SAML lunghe fino a 100.000 caratteri.

`POSTA/saml2/idpresponse`

Per utilizzare l'`/saml2/idpresponse`endpoint in un IdP-initiated accesso, genera una richiesta POST con parametri che forniscano al tuo pool di utenti informazioni sulla sessione dell'utente.

- Il client dell'app a cui vogliono accedere.
- L'URL di callback a cui vogliono essere reindirizzati.
- Gli ambiti OAuth 2.0 che vogliono richiedere nel token di accesso dell'utente.
- L'IdP che ha avviato la richiesta di accesso.

IdP-initiated parametri del corpo della richiesta

Risposta SAML

Un'asserzione Base64-encoded SAML di un IdP associata a un client di app valido e a una configurazione IdP nel tuo pool di utenti.

RelayState

Un `RelayState` parametro contiene i parametri di richiesta che altrimenti passeresti all'endpoint. `oauth2/authorize` Per ulteriori informazioni su questi parametri, consulta Endpoint Authorize.

response_type

Il tipo di concessione OAuth 2.0.

client_id

L'ID del client dell'applicazione.

redirect_uri

L'URL a cui il server di autenticazione reindirizza il browser una volta che Amazon Cognito concede l'autorizzazione all'utente.

identity_provider

Il nome del provider di identità a cui desideri reindirizzare l'utente.

idp_identifier

L'identificatore del provider di identità a cui desideri reindirizzare l'utente.

scope

Gli ambiti OAuth 2.0 che desideri che l'utente richieda al server di autorizzazione.

Richieste di esempio con risposte positive

Esempio: richiesta POST

La seguente richiesta riguarda la concessione di un codice di autorizzazione per un utente da IdP `MySAMLIdP` nel client dell'app. `1example23456789` L'utente reindirizza a `https://www.example.com` con il proprio codice di autorizzazione, che può essere scambiato con token che includono un token di accesso con ambiti OAuth 2.0, e. `openid email` phone

```
POST /saml2/idpresponse HTTP/1.1
User-Agent: USER_AGENT
Accept: */*
Host: example.auth.us-east-1.amazoncognito.com
Content-Type: application/x-www-form-urlencoded

SAMLResponse=[Base64-encoded SAML assertion]&RelayState=identity_provider
%3DMySAMLIdP%26client_id%3D1example23456789%26redirect_uri%3Dhttps%3A%2F
%2Fwww.example.com%26response_type%3Dcode%26scope%3Demail%2Bopenid%2Bphone
```

Esempio: risposta

Di seguito è riportata la risposta alla richiesta precedente.

```
HTTP/1.1 302 Found
Date: Wed, 06 Dec 2023 00:15:29 GMT
Content-Length: 0
x-amz-cognito-request-id: 8aba6eb5-fb54-4bc6-9368-c3878434f0fb
Location: https://www.example.com?code=[Authorization code]
```

Concessioni OAuth 2.0

Il server di autorizzazione OAuth 2.0 del pool di utenti Amazon Cognito emette token in risposta a tre tipi di OAuth 2.0 di concessioni di autorizzazione. Puoi impostare i tipi di concessione supportati per ogni client dell'app nel pool di utenti. Non puoi abilitare concessioni delle credenziali del client nello stesso client dell'app come concessioni implicite o del codice di autorizzazione. Le richieste per concessioni implicite e del codice di autorizzazione iniziano in in Endpoint Authorize e le richieste per concessioni delle credenziali dei clienti iniziano in Endpoint Token.

Concessione codice autorizzazione

In risposta alla richiesta di autenticazione riuscita, il server di autorizzazione aggiunge un codice di autorizzazione in un parametro `code` all'URL di callback. Occorre quindi scambiare il codice per token ID, di accesso e di aggiornamento con Endpoint Token. Per richiedere una concessione del codice di autorizzazione, imposta `response_type` su `code` nella richiesta. Per una richiesta di esempio, consulta Esempio: concessione del codice di autorizzazione. Amazon Cognito supporta Proof Key for Code Exchange (PKCE) nelle concessioni di codici di autorizzazione.

La concessione del codice di autorizzazione è la forma più sicura di concessione di autorizzazione. Non mostra il contenuto dei token direttamente agli utenti. L'app è invece responsabile del recupero e dell'archiviazione sicura dei token dell'utente. In Amazon Cognito, una concessione del codice di autorizzazione è l'unico modo per ottenere tutti e tre i tipi di token (ID, accesso e aggiornamento) dal server di autorizzazione. Puoi anche ottenere tutti e tre i tipi di token dall'autenticazione tramite l'API dei pool di utenti di Amazon Cognito, ma l'API non emette token di accesso con ambiti diversi da. `aws.cognito.signin.user.admin`

Implicit grant (Concessione implicita)

In risposta alla richiesta di autenticazione riuscita, il server di autorizzazione aggiunge un token di accesso in un parametro `access_token` e un token ID in un parametro `id_token`, all'URL di callback. Una concessione implicita non richiede alcuna interazione aggiuntiva con Endpoint

Token. Per richiedere una concessione implicita, imposta `response_type` su `token` nella richiesta. La concessione implicita genera solo un ID e un token di accesso. Per una richiesta di esempio, consulta Esempio: concessione di token (implicita) senza ambito `openid`.

La concessione implicita è una concessione di autorizzazione legacy. A differenza della concessione del codice di autorizzazione, gli utenti possono intercettare e ispezionare i token. Per impedire la distribuzione dei token tramite concessione implicita, configura il client dell'app in modo che supporti solo la concessione del codice di autorizzazione.

Client credentials (Credenziali del client)

Credenziali del client è una concessione di sola autorizzazione per l'accesso tra computer. Per ricevere una concessione delle credenziali del client, ignora il Endpoint Authorize e genera una richiesta direttamente a Endpoint Token. Il client dell'app deve disporre di un segreto del client e supportare solo le concessioni delle credenziali del cliente. In risposta alla richiesta riuscita, il server di autorizzazione restituisce un token di accesso.

Il token di accesso proveniente dalla concessione delle credenziali del client è un meccanismo di autorizzazione che contiene ambiti OAuth 2.0. In genere, il token contiene attestazioni di ambito personalizzate che autorizzano operazioni HTTP alle API ad accesso protetto. Per ulteriori informazioni, consulta Ambiti, M2M e server di risorse.

Le credenziali del cliente garantiscono costi aggiuntivi alla fattura. AWS Per ulteriori informazioni, consultare Prezzi di Amazon Cognito.

Token di aggiornamento

Puoi richiedere la concessione di un token di aggiornamento direttamente da. Endpoint Token Questa concessione restituisce nuovi ID e token di accesso in cambio di un token di aggiornamento valido.

Per ulteriori informazioni su queste sovvenzioni e sulla loro implementazione, consulta Come usare OAuth 2.0 in Amazon Cognito: scopri le diverse sovvenzioni OAuth 2.0 nel blog sulla sicurezza.AWS

Utilizzo di PKCE nelle concessioni di codici di autorizzazione

Amazon Cognito supporta l'autenticazione Proof Key for Code Exchange (PKCE) nelle concessioni di codici di autorizzazione. PKCE è un'estensione della concessione del codice di autorizzazione OAuth 2.0 per i clienti pubblici. PKCE protegge dal riscatto dei codici di autorizzazione intercettati.

In che modo Amazon Cognito utilizza PKCE

Per avviare l'autenticazione con PKCE, l'applicazione deve generare un valore di stringa univoco. Questa stringa è il verificatore del codice, un valore segreto che Amazon Cognito utilizza per confrontare il client che richiede la concessione di autorizzazione iniziale con il client che scambia il codice di autorizzazione in token.

L'app deve applicare un hash SHA256 alla stringa del verificatore del codice e codificare il risultato in base64. Passa la stringa con hash al parametro as nel corpo della richiesta. Endpoint Authorize `code_challenge` Quando l'app scambia il codice di autorizzazione in cambio di token, deve includere la stringa del verificatore del codice in testo semplice come `code_verifier` parametro nel corpo della richiesta a. Endpoint Token Amazon Cognito esegue la stessa operazione di hash e codifica sul verificatore di codice. Amazon Cognito restituisce ID, accesso e token di aggiornamento solo se determina che il verificatore del codice genera la stessa richiesta di codice ricevuta nella richiesta di autorizzazione.

Per implementare Authorization Grant Flow con PKCE

1. Apri la console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo. Se crei un pool di utenti, durante la procedura guidata ti verrà richiesto di configurare un client per l'app e configurare l'accesso gestito.

    a. Se crei un nuovo pool di utenti, configura un client per l'app e configura l'accesso gestito durante la configurazione guidata.

    b. Se configuri un pool di utenti esistente, aggiungi un dominio e un client di app pubblico, se non l'hai già fatto.
4. Genera una stringa alfanumerica casuale, in genere un identificatore univoco universale (UUID), per creare una sfida di codice per il PKCE. Questa stringa è il valore del `code_verifier` parametro che verrà inviato nella richiesta a. Endpoint Token
5. Effettua l'hash della `code_verifier` stringa con l'algoritmo SHA256. Codifica il risultato dell'operazione di hashing in base64. Questa stringa è il valore del `code_challenge` parametro che verrà inviato nella richiesta a. Endpoint Authorize

    L'Pythonesempio seguente genera un `code_verifier` e calcola: `code_challenge`

```
#!/usr/bin/env python3
```

```
import secrets
from base64 import urlsafe_b64encode
from hashlib import sha256
from string import ascii_letters
from string import digits

# use the secrets module for cryptographically strong random values
alphabet = ascii_letters + digits
code_verifier = ''.join(secrets.choice(alphabet) for _ in range(128))
code_verifier_hash = sha256(code_verifier.encode()).digest()
code_challenge = urlsafe_b64encode(code_verifier_hash).decode().rstrip('=')

print(f"code challenge: {code_challenge}")
print(f"code verifier: {code_verifier}")
```

Di seguito è riportato un esempio di output dello Python script:

```
code challenge: Eh0mg-OZv7BAyo-tdv_vYamx1boOYDulDklyXoMDtLg
code verifier: 9D-aW_iygXrgQcWJd0y0tNVMPSXSChIc2xceDhvYVdGLCBk-
JWFTmBNjvKSdOrjTTYazOFbUmrFERrjWx6oKtK2b6z_x4_gHBDlr4K1mRFGyE8yA-05-_v7Dxf3EIYJH
```

6. Completa l'accesso gestito con una richiesta di concessione del codice di autorizzazione con PKCE. Di seguito è riportato un esempio di URL:

```
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=1example23456789&redirect_uri=https://
www.example.com&code_challenge=Eh0mg-OZv7BAyo-
tdv_vYamx1boOYDulDklyXoMDtLg&code_challenge_method=S256
```

7. Raccogli l'autorizzazione `code` e riscattala in token con l'endpoint del token. Di seguito è riportato un esempio di richiesta:

```
POST /oauth2/token HTTP/1.1
Host: mydomain.auth.us-east-1.amazoncognito.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 296

redirect_uri=https%3A%2F%2Fwww.example.com&
client_id=1example23456789&
code=7378f445-c87f-400c-855e-0297d072ff03&
grant_type=authorization_code&
```

```
code_verifier=9D-aW_iygXrgQcWJd0y0tNVMPSXSChIc2xceDhvYVdGLCBk-
JWFTmBNjvKSdOrjTTYazOFbUmrFERrjWx6oKtK2b6z_x4_gHBDlr4K1mRFGyE8yA-05-_v7Dxf3EIYJH
```

8. Esamina la risposta. Contiene ID, token di accesso e aggiornamento. Per ulteriori informazioni sull'utilizzo dei token del pool di utenti di Amazon Cognito, consulta. Comprensione dei token web JSON (JWT) del pool di utenti

Risposte gestite agli errori di accesso e federazione

Una procedura di accesso in accesso gestito o accesso federato potrebbe restituire un errore. Di seguito sono riportate alcune condizioni che possono determinare un errore di autenticazione.

- Un utente esegue un'operazione che il pool di utenti non è in grado di eseguire.
- Un trigger Lambda non risponde con la sintassi prevista.
- Il gestore dell'identità digitale restituisce un errore.
- Amazon Cognito non è riuscito a convalidare le informazioni sugli attributi fornite dall'utente.
- L'IdP non ha inviato richieste che corrispondono agli attributi richiesti.

Quando Amazon Cognito rileva un errore, lo comunica in uno dei seguenti modi.

1. Amazon Cognito invia un URL di reindirizzamento con l'errore nei parametri della richiesta.
2. Amazon Cognito mostra un errore nell'accesso gestito.

Gli errori aggiunti da Amazon Cognito ai parametri della richiesta hanno il seguente formato.

```
https://<Callback URL>/?error_description=error+description&error=error+name
```

Quando fornisci assistenza agli utenti per inviare informazioni sugli errori quando non è possibile eseguire un'operazione, richiedi che acquisiscano l'URL e il testo o uno screenshot della pagina.

> **Note**
>
> Le descrizioni degli errori di Amazon Cognito non sono stringhe fisse e non devi usare una logica basata su un modello o un formato fisso.

Messaggi di errore del provider di identità OIDC e social

Il provider di identità potrebbe restituire un errore. Quando un IdP OIDC o OAuth 2.0 restituisce un errore conforme agli standard, Amazon Cognito reindirizza l'utente all'URL di callback e aggiunge la risposta di errore del provider ai parametri della richiesta di errore. Amazon Cognito aggiunge il nome del provider e il codice di errore HTTP alle stringhe di errore esistenti.

L'URL seguente è un reindirizzamento di esempio da un IdP che ha restituito un errore ad Amazon Cognito.

```
https://www.amazon.com/?error_description=LoginWithAmazon+Error+-+400+invalid_request
+The+request+is+missing+a+required+parameter+%3A+client_secret&error=invalid_request
```

Poiché Amazon Cognito restituisce solo ciò che riceve da un provider, l'utente potrebbe visualizzare un sottoinsieme di queste informazioni.

Quando l'utente rileva un problema con l'accesso iniziale tramite l'IdP, questo invia eventuali messaggi di errore direttamente all'utente. Amazon Cognito inoltra un messaggio di errore all'utente quando genera una richiesta all'IdP per convalidare la sessione dell'utente. Amazon Cognito inoltra i messaggi di errore IdP OAuth e OIDC dai seguenti endpoint.

`/token`

Amazon Cognito scambia un codice di autorizzazione IdP per un token di accesso.

`/.well-known/openid-configuration`

Amazon Cognito rileva il percorso agli endpoint dell'emittente.

`/.well-known/jwks.json`

Per verificare i token web JSON (JWT) dell'utente, Amazon Cognito rileva le chiavi Web JSON (JWK) utilizzate dal tuo IdP per firmare i token.

Poiché Amazon Cognito non avvia sessioni in uscita verso i provider SAML 2.0 che potrebbero restituire errori HTTP, gli errori degli utenti durante una sessione con un IdP SAML 2.0 non includono questa forma di messaggio di errore del provider.

Implementazione e funzionamento

Una volta che il design di Amazon Cognito è pronto, lo gestisci come parte di un servizio di produzione. Questa sezione tratta i problemi operativi legati all'esecuzione di Amazon Cognito su larga scala: velocità di richiesta e quote di risorse, capacità fornita, replica in più regioni, etichettatura delle risorse e risoluzione dei problemi. Per la registrazione e il monitoraggio (metriche e cronologia degli eventi), consulta il capitolo SicurezzaCloudWatch . CloudTrail Registrazione e monitoraggio in Amazon Cognito

Argomenti

- Quote in Amazon Cognito
- Multi-Region replica per pool di utenti
- Assegnazione di tag alle risorse Amazon Cognito
- Risoluzione dei problemi relativi ad Amazon Cognito

Quote in Amazon Cognito

Amazon Cognito dispone di quote predefinite, precedentemente definite limiti, per il numero massimo di operazioni che possono essere eseguite nell'account. Amazon Cognito dispone anche di quote relative al numero massimo e alle dimensioni massime delle risorse di Amazon Cognito.

Ogni quota di Amazon Cognito rappresenta un volume massimo di richieste Regione AWS in una a una. Account AWS Ad esempio, le app possono effettuare richieste API fino alla frequenza di quota predefinita (RPS) per operazioni `UserAuthentication` rispetto a tutti i pool di utenti negli Stati Uniti orientali (Virginia settentrionale). Le tue app in Asia Pacifico (Tokyo) possono generare lo stesso volume di richieste per tutti i tuoi pool di utenti nella rispettiva regione. AWS può concedere una richiesta di aumento della quota solo in una regione alla volta. Un aumento della quota nella regione Stati Uniti orientali (Virginia settentrionale) non ha alcun effetto sulla frequenza massima di richiesta nella regione Asia Pacifico (Tokyo).

Argomenti

- Informazioni sulle quote di frequenza di richiesta API
- Gestione delle quote di frequenza di richiesta API
- Le operazioni API del bacino d'utenza di Amazon Cognito e le quote di frequenza delle richieste

- Frequenza delle richieste di operazione API dei pool di identità di Amazon Cognito (identità federate)
- Quote relative al numero e alla dimensione delle risorse

Informazioni sulle quote di frequenza di richiesta API

Categorizzazione delle quote

Amazon Cognito impone una frequenza massima di richieste per le operazioni API. Per ulteriori informazioni sulle operazioni API rese disponibili da Amazon Cognito, consulta le guide di riferimento API per pool di utenti e pool di identità. Per i pool di utenti, queste operazioni sono raggruppate in categorie di casi d'uso comuni come `UserAuthentication` o. `UserCreation` Per un elenco delle operazioni API del pool di utenti per categoria, vedereLe operazioni API del bacino d'utenza di Amazon Cognito e le quote di frequenza delle richieste.

Nella console Service Quotas, puoi monitorare l'utilizzo delle quote per pool di utenti di categoria e pool di identità. Se il tasso di richieste dei tuoi pool di utenti Amazon Cognito supera o supera una quota, puoi acquistare capacità aggiuntiva. Puoi tenere traccia dell'utilizzo delle quote del pool di utenti per categoria e degli aumenti delle quote di acquisto nella console Service Quotas.

Le quote di operazione sono definite come il numero massimo di richieste consentite al secondo (RPS) per tutte le operazioni all'interno di una categoria. Il servizio dei bacini d'utenza Amazon Cognito applica quote a tutte le operazioni in ciascuna categoria. Ad esempio, la categoria `UserCreation` include quattro operazioni: `SignUp`, `ConfirmSignUp`, `AdminCreateUser` e `AdminConfirmSignUp`. È assegnata con una quota combinata di 50 RPS. Se vengono eseguite più operazioni contemporaneamente, ogni operazione all'interno di questa categoria può richiamare fino a 50 RPS separate o combinate.

> **Note**
>
> Le quote di categoria si applicano solo a pool di utenti. Amazon Cognito applica ogni quota del pool di identità a una singola operazione. Per le quote di frequenza delle richieste, sia per categoria che per operazione, AWS misura il tasso aggregato di tutte le richieste provenienti da tutti i pool di utenti o pool di identità presenti in un'unica regione. Account AWS

Operazioni API del bacino d'utenza di Amazon Cognito con una gestione speciale della frequenza delle richieste

Le quote di operazione sono misurate e applicate per le richieste totali combinate a livello di categoria, ad eccezione delle operazioni `AdminRespondToAuthChallenge` e `RespondToAuthChallenge`, in cui vengono applicate regole di gestione speciali.

La `UserAuthentication` categoria include quattro operazioni nell'API dei pool di utenti di Amazon Cognito:`AdminInitiateAuth`, `InitiateAuthAdminRespondToAuthChallenge`, e. `RespondToAuthChallenge` Inoltre, l'autenticazione degli utenti nell'interfaccia utente ospitata contribuisce a questa quota. Le operazioni `InitiateAuth` e `AdminInitiateAuth` sono misurate e applicate per quota di categoria. Le operazioni di corrispondenza `RespondToAuthChallenge` e `AdminRespondToAuthChallenge` sono soggette a una quota separata che è tre volte superiore al limite di categoria `UserAuthentication`. Questa quota elevata soddisfa le molteplici sfide di autenticazione impostate nelle tue app. La quota è sufficiente a coprire la grande maggioranza dei casi d'uso. Dopo che l'app ha fornito fino a tre risposte ai problemi di autenticazione, le richieste aggiuntive vengono conteggiate ai fini della quota di `UserAuthentication` categoria. Multi-factor l'autenticazione (MFA), l'autenticazione del dispositivo e l'autenticazione personalizzata sono tutti esempi di richieste di sfida che potresti inserire nel tuo pool di utenti.

Ad esempio, se la tua quota per la `UserAuthentication` categoria è di 80 RPS, puoi chiamare `RespondToAuthChallenge` o `AdminRespondToAuthChallenge` a una velocità fino a 240 RPS (3* 80 RPS). Se il tuo pool di utenti richiede quattro round di sfida per autenticazione e 70 utenti accedono al secondo, il totale `RespondToAuthChallenge` è di 280 RPS (70 x 4), ovvero 40 RPS in più rispetto alla quota. Il valore di 40 RPS extra viene aggiunto a 70 chiamate `InitiateAuth`, rendendo l'utilizzo totale della categoria `UserAuthentication` 110 RPS (40 + 70). Poiché questo valore supera la quota di categoria fissata a 80 RPS per 30 RPS, Amazon Cognito limita le richieste provenienti dalla tua app.

Utenti mensili attivi

Quando Amazon Cognito calcola la fatturazione del pool di utenti, ti addebita una tariffa per ogni utente attivo mensile (MAU). Considera il numero di MAU attuale e previsto nella pianificazione delle richieste di aumento delle quote. Un utente viene conteggiato come MAU se, entro un mese di calendario, viene eseguita un'operazione di identità correlata a tale utente. Quando colleghi utenti federati a utenti locali, con la federazione SAML o OIDC, l'utente locale viene conteggiato come MAU dell'elenco aziendale o`EnterpriseMAU`, indipendentemente dal fatto che l'utente acceda direttamente o tramite federazione. Per ulteriori informazioni, consulta Prezzi di Amazon Cognito.

- Sign-up o creazione amministrativa di un utente. L'importazione di file CSV degli utenti non contribuisce al conteggio dei MAU.
- Conferma dell'account utente o verifica degli attributi.
- Sign-in e risposta alla sfida. Le operazioni che autorizzi con il token di accesso dell'utente attualmente connesso non contribuiscono al conteggio MAU; tuttavia, poiché l'accesso produce token di accesso, queste operazioni indicano che l'utente associato è un MAU.
- Sign-out e revoca dei token.
- Reimpostazione self-service delle password e impostazione delle password degli utenti come amministratore. La reimpostazione delle password degli utenti come amministratore (AdminResetUserPassword) non contribuisce al conteggio delle MAU.
- Modifica gli attributi utente o l'appartenenza al gruppo.
- Interroga gli attributi dettagliati di un utente in qualità di amministratore.

> **Note**
>
> La categoria Interroga gli attributi dettagliati di un utente in qualità di amministratore include il funzionamento dell'API AdminGetUser, ma non ListUsers. Una richiesta dettagliata utente per utente in un ampio pool di utenti può avere un impatto significativo sulla fattura AWS . Per evitare costi aggiuntivi, raccogli i dati degli utenti `ListUsers` o archivia le informazioni sugli utenti in un database esterno.

Non ti vengono addebitati costi per sessioni aggiuntive da parte di alcun utente attivo o per gli utenti che non lo sono stati entro un mese di calendario. Nel mese in cui hai cambiato il piano di funzionalità del pool di utenti tra le opzioni disponibili di Lite, Essentials e Plus, la fattura per quel mese viene calcolata sulla base della somma degli utenti attivi mensili (MAU) di ciascun livello, con ogni MAU assegnato al livello assegnato con il prezzo più alto quando l'utente era attivo. Ad esempio:

1. All'inizio del mese, il tuo pool di utenti ha sottoscritto il piano di funzionalità Plus.
2. L'utente A accede il primo giorno del mese.
3. L'utente B accede il primo e l'ultimo giorno del mese.
4. Il decimo giorno del mese, passi il tuo piano di funzionalità a Essentials.
5. L'utente C accede l'ultimo giorno del mese.

In questo scenario, l'utente A e l'utente B sono MAU Plus e l'utente C è un MAU Essentials.

MAU Lite

Un utente che era attivo almeno una volta al mese quando il pool di utenti utilizzava il piano di funzionalità Lite e non lo era mai quando il pool di utenti utilizzava i piani Essentials o Plus.

Essentials MAU

Un utente che era attivo almeno una volta al mese quando il pool di utenti utilizzava il piano di funzionalità Essentials e non era mai attivo quando il pool di utenti era nel piano Plus.

Plus MAU

Un utente che era attivo almeno una volta al mese quando il pool di utenti era incluso nel piano Plus.

Per ulteriori informazioni, consulta Piani di funzionalità del pool di utenti.

Gestione delle quote di frequenza di richiesta API

Identificazione dei requisiti delle quote

> **Important**
>
> Se aumenti le quote di Amazon Cognito per categorie come`UserAuthentication`, oppure `UserCreationAccountRecovery`, potresti dover aumentare le quote per altre. Servizi AWS Ad esempio, i messaggi inviati da Amazon Cognito con Amazon Simple Notification Service (Amazon SNS) e Amazon Simple Email Service (Amazon SES) possono non riuscire se le quote di richiesta sono insufficienti in tali servizi.

Per calcolare i requisiti delle quota, determina quanti utenti attivi interagiranno con l'applicazione in un determinato periodo di tempo. Ad esempio, se prevedi l'accesso medio di un milione di utenti attivi in un periodo di otto ore, sarà necessario essere in grado di autenticare una media di 35 utenti al secondo.

Inoltre, se si suppone che la sessione utente media sia di due ore e che i token siano configurati per scadere dopo un'ora, durante questa sessione di due ore ogni utente dovrà aggiornare una volta i propri token. La quota media richiesta per la categoria `UserAuthentication` per supportare questo carico è 70 RPS.

Se si suppone un rapporto picco/medio di 3:1 tenendo conto della variazione della frequenza di accesso degli utenti durante il periodo di otto ore, è necessaria la quota `UserAuthentication` desiderata di 200 RPS.

> **Note**
>
> Se si chiamano più operazioni per ogni azione utente, dovrai sommare i tassi di chiamata delle singole operazioni a livello di categoria.

Ottimizzare i tassi di richiesta per i limiti di quota

Poiché l'aumento dei limiti tariffari delle API comporta costi aggiuntivi AWS , valuta la possibilità di apportare modifiche al modello di utilizzo prima di richiedere un aumento della quota. Di seguito sono riportati alcuni esempi di architettura di app che ottimizza i tassi di richiesta.

Nuovo tentativo dopo un periodo di attesa di back-off

È possibile rilevare gli errori con ogni chiamata API e quindi riprovare dopo un periodo di back-off. È possibile regolare l'algoritmo di back-off in base alle esigenze aziendali e al carico. Gli SDK di Amazon dispongono di una logica di tentativi incorporata. Per ulteriori informazioni, consulta Strumenti su AWS cui basarsi.

Utilizzo di un database esterno per gli attributi aggiornati di frequente

Se l'applicazione richiede diverse chiamate a un bacino d'utenza per leggere o scrivere attributi personalizzati, utilizza l'archiviazione esterna. Puoi utilizzare il tuo database preferito per archiviare attributi personalizzati oppure utilizzare un livello di cache per caricare un profilo utente durante l'accesso. Puoi fare riferimento a questo profilo dalla cache quando necessario invece di ricaricare il profilo utente da un bacino d'utenza.

Convalida i token web JSON (JWT) sul lato client

Le applicazioni devono convalidare i token JWT prima di potersi fidare di loro. Puoi verificare la firma e la validità dei token sul lato client senza inviare richieste API a un pool di utenti. Una volta convalidato il token, sarà possibile considerare attendibili le attestazioni nel token e utilizzare le attestazioni invece di creare più chiamate API `getUser`. Per ulteriori informazioni, consulta Verifica di un token web JSON.

Limitare il traffico verso la tua applicazione Web con una sala d'attesa

Se prevedi traffico da un numero elevato di utenti che effettuano l'accesso durante un evento limitato nel tempo, come un esame o un evento live, puoi ottimizzare il traffico delle richieste con meccanismi di limitazione automatica. Ad esempio, puoi impostare una sala d'attesa in cui gli utenti possono attendere fino a quando non è disponibile una sessione, consentendoti di elaborare le richieste quando la capacità è disponibile. Consulta Soluzione Virtual Waiting Room AWS per un'architettura di riferimento di una sala d'attesa.

JWT nella cache

Riutilizza i token di accesso fino alla loro scadenza. Per un esempio di framework con memorizzazione nella cache dei token in un API Gateway, vedereGestione della scadenza e della memorizzazione nella cache dei token del pool di utenti. Invece di generare richieste API per interrogare le informazioni degli utenti, memorizzate nella cache i token ID fino alla loro scadenza e leggete gli attributi utente dalla cache.

Per ulteriori informazioni sull'utilizzo delle frequenze di richiesta API in AWS, consulta Gestione e monitoraggio della limitazione delle API nei carichi di lavoro. Per informazioni sull'ottimizzazione delle operazioni di Amazon Cognito che aggiungono costi alla bolletta, AWS consulta. Gestione dei costi

Monitoraggio dell'uso delle quote

Amazon Cognito genera `CallCount` e `ThrottleCount` misura in Amazon CloudWatch per ogni categoria di operazioni API a livello di account. Puoi utilizzare `CallCount` per tenere traccia del numero totale di chiamate effettuate dai clienti in relazione a una categoria. Puoi utilizzare `ThrottleCount` per tenere traccia del numero totale di chiamate limitate relative a una categoria. Puoi utilizzare i parametri `CallCount` e `ThrottleCount` con la statistica `Sum` per contare il numero totale di chiamate in una categoria. Per ulteriori informazioni, consulta le metriche di CloudWatch utilizzo.

Quando si monitorano le quote di servizio, l'utilizzo è la percentuale di una quota di servizio in uso. Ad esempio, se il valore della quota è 200 risorse e 150 risorse sono in uso, l'utilizzo è 75%. L'uso è invece il numero di risorse o operazioni in uso per una quota di servizio.

Monitoraggio dell'utilizzo tramite metriche CloudWatch

Puoi tracciare e raccogliere le metriche di utilizzo dei pool di utenti di Amazon Cognito con. CloudWatch La CloudWatch dashboard mostra le metriche relative a tutti Servizio AWS gli elementi

utilizzati. Con CloudWatch, puoi creare allarmi metrici per avvisarti o modificare una risorsa specifica che stai monitorando. Per ulteriori informazioni sulle CloudWatch metriche, consulta Tieni traccia delle metriche di utilizzo CloudWatch .

Monitoraggio dell'utilizzo tramite i parametri Service Quotas

I pool di utenti di Amazon Cognito sono integrati con Service Quotas, un'interfaccia console per visualizzare e gestire l'utilizzo delle quote di servizio. Nella console Service Quotas, puoi cercare il valore di una quota specifica, visualizzare le informazioni di monitoraggio, richiedere un aumento della quota o impostare CloudWatch allarmi. Dopo un certo periodo di attività dell'account, puoi visualizzare un grafico dell'utilizzo delle risorse.

La colonna Valore di quota a livello di account applicato nella console Service Quotas per i pool di utenti di Amazon Cognito e i pool di identità di Amazon Cognito mostra la tua quota attuale. La colonna Utilizzo mostra il tasso attuale di utilizzo delle quote. Le quote di richieste al secondo (RPS) dei pool di utenti regolabili di Amazon Cognito mostrano l'utilizzo corrente. La console Service Quotas può anche accedere alle CloudWatch metriche per esaminare più da vicino una metrica di quota selezionata. Per ulteriori informazioni sulla visualizzazione delle quote nella console Service Quotas, consulta Visualizzazione di Service Quotas.

Tieni traccia degli utenti attivi mensili (MAU)

Il numero di utenti attivi mensili (MAU) nel tuo pool di utenti fornisce dati importanti per pianificare l'aumento delle quote relative alle richieste. Puoi confrontare i tassi di richiesta API con il numero di utenti attivi in un determinato periodo di tempo. Con queste conoscenze, potete calcolare in che modo un aumento degli utenti attivi delle vostre applicazioni influenzi le vostre quote nel modello di utilizzo. Ad esempio, immaginate che le vostre applicazioni combinate negli Stati Uniti occidentali (Oregon) generino 2 milioni di utenti attivi in un mese e che la vostra `UserAuthentication` categoria riceva occasionali errori di limitazione rispetto alla quota predefinita di 120 richieste al secondo (RPS). Nel mese precedente, prima del successo della vostra campagna pubblicitaria, avevate 1 milione di MAU e le vostre applicazioni non superavano mai gli 80 RPS. Se prevedi un picco simile a seguito di un nuovo spot televisivo, potresti acquistare altri 40 RPS per soddisfare il prossimo milione di utenti con una quota adeguata di 160 RPS.

Per recensire i tuoi MAU

Accedi alla AWS Billing console e rivedi una fattura recente. Nella sezione Addebiti per servizio, puoi filtrare su Cognito per visualizzare un'analisi dettagliata delle tue MAU per quel periodo di fatturazione.

Richiedere un aumento della quota

Amazon Cognito ha una quota per il numero massimo di operazioni al secondo che puoi eseguire nei tuoi pool di utenti e pool di identità in ciascuno. Regione AWS Puoi acquistare un aumento delle quote di richiesta API dei pool di utenti di Amazon Cognito regolabili. Per le categorie di pool di utenti regolabili, puoi richiedere un limite massimo a livello di account più elevato in Service Quotas e quindi utilizzare l'API di provisioning di Amazon Cognito per impostare la tariffa desiderata. Per ulteriori informazioni, consulta Gestione dei limiti predisposti.

Per richiedere un limite massimo più elevato a livello di account, utilizza la console Service Quotas o le operazioni dell'API Service Quotas e. `ListAWSDefaultServiceQuotas` `RequestServiceQuotaIncrease`

- Per richiedere un aumento del limite massimo a livello di account utilizzando la console Service Quotas, consulta Richiesta di un aumento della quota API nella Service Quotas User Guide.
- Alcuni aumenti del limite massimo a livello di account vengono approvati automaticamente. In caso contrario, AWS prevede il completamento delle richieste di aumento delle quote entro 10 giorni. Tuttavia, diverse considerazioni potrebbero far sì che il tempo di elaborazione della richiesta superi i 10 giorni. Alcune richieste, ad esempio, potrebbero richiedere che Amazon Cognito fornisca capacità hardware aggiuntiva e gli aumenti stagionali dei volumi di richieste potrebbero causare ritardi.
- Se la quota non è disponibile in Service Quotas, utilizza il modulo di incremento dei limiti di servizio.

> **Note**
>
> Solo le quote modificabili possono essere aumentate.

> **Note**
>
> Gli aumenti delle quote sono configurazioni regionali. Un aumento della quota in una Regione AWS regione non influisce sulle quote nelle altre regioni. Se hai bisogno di quote API più elevate in più regioni, devi richiedere un aumento della quota in ciascuna regione separatamente.

> **Note**
>
> La richiesta di un aumento del limite massimo a livello di account è solo il primo passaggio. Dopo l'approvazione della richiesta, devi chiamare `UpdateProvisionedLimit` o utilizzare la console Amazon Cognito per impostare il limite assegnato. Per ulteriori informazioni, consulta Gestione dei limiti predisposti.

Gestione dei limiti predisposti

I pool di utenti di Amazon Cognito supportano un modello di quote a due livelli per quote tariffarie API regolabili. Con questo modello, puoi aumentare o diminuire in modo programmatico la frequenza di richieste assegnata senza contattare l'assistenza. AWS I due livelli funzionano nel modo seguente:

Account-level limite massimo

: La tariffa massima che è possibile fornire, impostata tramite Service Quotas. Per aumentare il limite massimo a livello di account, invia una richiesta di aumento della quota nella console Service Quotas. Alcune richieste vengono approvate automaticamente. Per ulteriori informazioni, consulta Richiedere un aumento della quota.

Limite di provisioning

: La frequenza, espressa in richieste al secondo (RPS), che Amazon Cognito applica attualmente al tuo account. Puoi regolare questo valore fino al limite massimo a livello di account con le operazioni dell'API di provisioning di Amazon Cognito.

Ad esempio, se il limite massimo a `UserAuthentication` livello di account è di 500 RPS, puoi utilizzare l'operazione `UpdateProvisionedLimit` API per impostare il limite assegnato su 300 RPS. Successivamente puoi ridurre il limite previsto riportandolo al valore predefinito e ti verrà addebitata solo la tariffa temporale più elevata.

> **Important**
>
> Ti viene fatturata la capacità assegnata superiore alla tariffa predefinita, ripartita proporzionalmente in base al tempo di utilizzo effettivo. Per i prezzi, consulta i prezzi di Amazon Cognito.

Gestione dei limiti assegnati nella console

Puoi visualizzare e gestire i limiti assegnati nella console Amazon Cognito. Apri la pagina User pool e scegli la scheda Provisioned limits. In questa scheda, puoi visualizzare i limiti attualmente assegnati e i limiti massimi a livello di account per ogni categoria di API regolabile. Puoi aumentare o diminuire la tariffa assegnata entro il limite massimo approvato a livello di account.

Operazioni API di provisioning

Amazon Cognito fornisce le seguenti operazioni API per la gestione dei limiti assegnati. Queste operazioni richiedono l'autenticazione e le autorizzazioni IAM corrispondenti.

[GetProvisionedLimit]

Restituisce il valore limite attualmente fornito per una categoria di API specifica. La risposta include il valore limite fornito e il valore limite predefinito (gratuito).

[UpdateProvisionedLimit]

Imposta il valore limite assegnato per una categoria di API specifica. Il valore richiesto deve essere compreso tra il limite predefinito e il limite massimo a livello di account Service Quotas. Questa operazione ha effetto immediato.

Entrambe le operazioni accettano un `LimitDefinition` parametro che identifica il limite da gestire. Per i limiti di velocità delle API dei pool di utenti, la definizione del limite ha una classe `API_CATEGORY` e una mappa degli attributi con la `Category` chiave. Ad esempio, per gestire il `UserAuthentication` limite, specifica una definizione di limite con`{"Category": "UserAuthentication"}`.

Example Esempio GetProvisionedLimit di richiesta e risposta

L'esempio seguente restituisce il limite fornito per la categoria `UserAuthentication` API.

Richiesta

```
{
    "LimitDefinition": {
        "LimitClass": "API_CATEGORY",
        "Attributes": {"Category": "UserAuthentication"}
    }
}
```

Risposta

```
{
    "Limit": {
        "LimitDefinition": {
            "LimitClass": "API_CATEGORY",
            "Attributes": {"Category": "UserAuthentication"}
        },
        "ProvisionedLimitValue": 120,
        "FreeLimitValue": 120
    }
}
```

Example Esempio UpdateProvisionedLimit di richiesta e risposta

L'esempio seguente imposta il limite fornito per la categoria `UserAuthentication` API su 300 RPS.

Richiesta

```
{
    "LimitDefinition": {
        "LimitClass": "API_CATEGORY",
        "Attributes": {"Category": "UserAuthentication"}
    },
    "RequestedLimitValue": 300
}
```

Risposta

```
{
    "Limit": {
        "LimitDefinition": {
            "LimitClass": "API_CATEGORY",
            "Attributes": {"Category": "UserAuthentication"}
        },
        "ProvisionedLimitValue": 300,
        "FreeLimitValue": 120
    }
}
```

Le seguenti azioni IAM controllano l'accesso a queste operazioni:

- `cognito-idp:GetProvisionedLimit`
- `cognito-idp:UpdateProvisionedLimit`

> **Note**
>
> Cross-account l'accesso non è supportato. Solo l'account proprietario può gestire i limiti previsti.

Flusso di lavoro di provisioning

Il seguente flusso di lavoro descrive come aumentare il limite assegnato.

1. Determina la tariffa richiesta. Per ulteriori informazioni, consulta Identificazione dei requisiti delle quote.
2. Se la tariffa richiesta supera il limite massimo attuale a livello di account Service Quotas, richiedi un aumento nella console Service Quotas.
3. Dopo l'approvazione del limite massimo a livello di account, chiama con la tariffa desiderata. `UpdateProvisionedLimit`
4. Per ridurre la tariffa prevista e interrompere la fatturazione aggiuntiva, chiama `UpdateProvisionedLimit` con il valore limite predefinito o un valore inferiore.

Considerazioni

- I limiti assegnati sono risorse a livello di account. Si applicano alla velocità aggregata di tutte le richieste provenienti da tutti i pool di utenti in un'unica soluzione. Regione AWS Account AWS
- La `LimitManagement` categoria, che include `GetProvisionedLimit` e`UpdateProvisionedLimit`, ha una tariffa limitata a 1 richiesta al secondo per account.
- Solo le categorie di quote regolabili supportano il provisioning. Per un elenco di categorie regolabili, vedereLe operazioni API del bacino d'utenza di Amazon Cognito e le quote di frequenza delle richieste.
- Se il tuo pool di utenti utilizza l'accesso gestito, non puoi utilizzare l'API di provisioning per modificare le `UserFederation` categorie `UserAuthentication` o. Per aumentare queste quote, invia una richiesta di aumento delle quote di servizio tramite la console Service Quotas o utilizza il modulo di aumento del limite di servizio.

- Le modifiche alle quote assegnate sono regionali. Una modifica delle quote assegnate in una regione Regione AWS non influisce sulle quote in altre regioni.
- Tutte le chiamate alle operazioni API di provisioning vengono registrate. AWS CloudTrail

Le operazioni API del bacino d'utenza di Amazon Cognito e le quote di frequenza delle richieste

Poiché Amazon Cognito presenta classi di operazioni API sovrapposte con modelli di autorizzazione diversi, ogni operazione appartiene a una categoria. Ogni categoria ha la propria quota in pool per tutte le operazioni API del membro, in tutti i pool di utenti in una Regione AWS nell'account. È possibile richiedere solo l'incremento a quote di categoria regolabili. Per ulteriori informazioni, consulta Richiedere un aumento della quota. Gli adeguamenti della quota si applicano ai pool di utenti nell'account di una singola regione. Amazon Cognito limita le operazioni in alcune categorie[3] a 5 richieste al secondo (RPS), per pool di utenti. La quota predefinita (RPS) si applica inoltre a tutti i pool di utenti in un. Account AWS

> **Note**
>
> La quota per ogni categoria viene misurata in Utenti attivi mensili (MAU). Account AWS con meno di due milioni di MAU possono funzionare entro la quota predefinita. Se disponi di meno di un milione di MAU e Amazon Cognito limita le richieste, valuta la possibilità di ottimizzare la tua app. Per ulteriori informazioni, consulta Ottimizzare i tassi di richiesta per i limiti di quota.

Le quote delle operazioni di categoria vengono applicate a tutti gli utenti in tutti i pool di utenti all'interno di una Regione AWS. Amazon Cognito mantiene anche una quota per il numero di richieste che l'app può generare rispetto a un singolo utente. È necessario limitare le richieste API per utente come mostrato nella tabella seguente.

Quote frequenza di richiesta per utente di pool di utenti di Amazon Cognito

Operation	Operazioni per utente al secondo
Lettura profilo utente	10

Operation	Operazioni per utente al secondo
Esempi: `GetUser`, `GetDevice` , `InitiateAuth` , `RespondToAuthChallenge`	
Scrittura profilo utente Esempi: `UpdateUserAttributes` , `SetUserSettings`	10

È necessario limitare le richieste API per categoria come mostrato nella tabella seguente.

Quote frequenza di richiesta per categoria di pool di utenti di Amazon Cognito

Categoria	Description	Quota predefinita (RPS)	Regolabile
`UserAuthentication` • InitiateAuth • Aggiornamento del token con o con l'endpoint del token `InitiateAuth` • RespondToAuthChallenge[1] • AdminInitiateAuth • AdminRespondToAuthChallenge[1] • GetTokensFromRefreshToken • Accesso gestito o accesso classico all'interfaccia utente	Operazioni che autenticano (accesso) un utente. Queste operazioni sono soggette a Operazioni API del bacino d'utenza di Amazon Cognito con una gestione speciale della frequenza delle richieste.	120	Sì

Categoria	Description	Quota predefinita (RPS)	Regolabile
ospitata e MFA per utenti locali con codice di autorizzazione o concessioni implicite [2]			
`UserCreation` • SignUp • ConfirmSignUp • AdminCreateUser • AdminConfirmSignUp	Operazioni che creano o confermano un utente locale Amazon Cognito. Si tratta di un utente creato e verificato direttamente dai bacini d'utenza di Amazon Cognito.	50	Sì
`UserFederation` Operazioni che federano (autenticano) gli utenti con un provider di identità di terze parti nei bacini d'utenza Amazon Cognito.	Operazioni che inviano una risposta IdP a un endpoint di federazione di pool di utenti. Le operazioni OIDC o del provider social che generano un token IdP e tutte le richieste SAML contribuiscono a questa quota.	25	Sì

Categoria	Description	Quota predefinita (RPS)	Regolabile
`UserAccountRecovery` • ChangePassword • ConfirmForgotPassword • ForgotPassword • AdminResetUserPassword • AdminSetUserPassword • RespondToAuthChallenge[1] • AdminRespondToAuthChallenge[1] • Reimpostazione gestita della password di accesso	Operazioni che recuperano l'account di un utente o modificano o aggiornano la password di un utente.	30	No
`UserRead` • AdminGetUser • AdminGetUserAuthFactors • GetUser	Operazioni che recuperano un utente dai bacini d'utenza.	120	Sì

Categoria	Description	Quota predefinita (RPS)	Regolabile
`UserUpdate` • AdminAddUserToGroup • AdminDeleteUserAttributes • AdminUpdateUserAttributes • AdminDeleteUser • AdminDisableUser • AdminEnableUser • AdminLinkProviderForUser • AdminDisableProviderForUser • VerifyUserAttribute • DeleteUser • DeleteUserAttributes • UpdateUserAttributes • AdminUserGlobalSignOut • GlobalSignOut • AdminRemoveUserFromGroup	Operazioni utilizzi per gestire gli utenti e gli attributi utente.	25	No
`UserToken` • RevokeToken	Operazioni per la gestione dei token	120	Sì

Categoria	Description	Quota predefinita (RPS)	Regolabile
`UserResourceRead` • AdminGetDevice • AdminListGroupsForUser • AdminListDevices • GetDevice • ListDevices • GetUserAttributeVerificationCode • ResendConfirmationCode • AdminListUserAuthEvents	Operazioni che recuperano informazioni sulle risorse utente da Amazon Cognito, ad esempio un dispositivo memorizzato o l'appartenenza a un gruppo di utenti.	50	Sì

Categoria	Description	Quota predefinita (RPS)	Regolabile
`UserResourceUpdate` • AdminForgetDevice • AdminUpdateAuthEventFeedback • AdminSetUserMFAPreference • AdminDeleteSoftwareToken • AdminSetUserSettings • AdminUpdateDeviceStatus • UpdateDeviceStatus • UpdateAuthEventFeedback • ConfirmDevice • SetUserMFAPreference • SetUserSettings • VerifySoftwareToken • AssociateSoftwareToken • ForgetDevice	Operazioni che aggiornano informazioni sulle risorse per un utente, ad esempio un dispositivo memorizzato o l'appartenenza a un gruppo di utenti.	25	No

Categoria	Description	Quota predefinita (RPS)	Regolabile
`UserList` • ListUsers • ListUsersInGroup	Operazioni che restituiscono un elenco di utenti.	30	No
`UserPoolRead` • DescribeUserPool • ListUserPools	Operazioni che leggono i bacini d'utenza.	15	No
`UserPoolUpdate` • CreateUserPool • UpdateUserPool • DeleteUserPool	Operazioni che creano, aggiornano o eliminano i bacini d'utenza.	15	No

Categoria	Description	Quota predefinita (RPS)	Regolabile
`UserPoolResourceRead` • DescribeIdentityProvider • DescribeResourceServer • DescribeUserImportJob • DescribeUserPoolDomain • GetCSVHeader • GetGroup • GetSigningCertificate • GetIdentityProviderByIdentifier • GetUserPoolMfaConfig • ListGroups • ListIdentityProviders • ListResourceServers • ListTagsForResource • ListUserImportJobs • DescribeRiskConfiguration • GetUICustomization	Operazioni che recuperano informazioni sulle risorse, come gruppi o server di risorse, da un pool di utenti.[3]	20	No

Categoria	Description	Quota predefinita (RPS)	Regolabile
• DescribeManagedLoginBranding • DescribeManagedLoginBrandingByClient • ListUserPoolReplicas			

Categoria	Description	Quota predefinita (RPS)	Regolabile
`UserPoolResourceUpdate` • AddCustomAttributes • CreateGroup • CreateIdentityProvider • CreateResourceServer • CreateUserImportJob • CreateUserPoolDomain • DeleteGroup • DeleteIdentityProvider • DeleteResourceServer • DeleteUserPoolDomain • SetUserPoolMfaConfig • StartUserImportJob • StopUserImportJob • UpdateGroup • UpdateIdentityProvider • UpdateResourceServer	Operazioni che modificano le risorse, come gruppi o server di risorse, in un pool di utenti.[3]	15	No

Categoria	Description	Quota predefinita (RPS)	Regolabile
• UpdateUserPoolDomain • SetRiskConfiguration • SetUICustomization • TagResource • UntagResource • CreateManagedLoginBranding • UpdateManagedLoginBranding • DeleteManagedLoginBranding • CreateUserPoolReplica • UpdateUserPoolReplica • DeleteUserPoolReplica			
`UserPoolClientRead` • DescribeUserPoolClient • ListUserPoolClients	Operazioni che recuperano informazioni sui client del pool di utenti.[3]	15	No

Categoria	Description	Quota predefinita (RPS)	Regolabile
`UserPoolClientUpdate` • CreateUserPoolClient • DeleteUserPoolClient • UpdateUserPoolClient	Operazioni che creano, aggiornano ed eliminano i client del pool di utenti.[3]	15	No
`ClientAuthentication` • `client_credentials` richieste di tipo di concessione all'endpoint del token • GetClientToken	Operazioni che generano credenziali da utilizzare per l'autorizzazione di richieste da macchina a macchina	150	No
`LimitManagement` • GetProvisionedLimit • UpdateProvisionedLimit	Operazioni che recuperano o aggiornano i limiti di velocità API forniti	1	No

[1] A `RespondToAuthChallenge` o `AdminRespondToAuthChallenge` risposta con un `ChallengeName` di `NEW_PASSWORD_REQUIRED` conteggi ai fini della `UserAccountRecovery` categoria. Tutte le altre risposte alla sfida contano ai fini della `UserAuthentication` categoria.

[2] Ogni accesso gestito o operazione classica dell'interfaccia utente ospitata durante l'accesso contribuisce con una richiesta alla quota. Ad esempio, un utente che accede e fornisce un codice MFA contribuisce con 2 richieste. Il riscatto dei token nell'ambito della concessione di codici di

autorizzazione è soggetto all'assegnazione di quote aggiuntive allo stesso tasso della quota prevista per la categoria. `UserAuthentication`

[3] Ogni singola operazione in questa categoria presenta un vincolo che impedisce che l'operazione venga richiamata a una velocità superiore a 5 RPS per un singolo pool di utenti.

Limiti di frequenza delle richieste in blocco per i domini del pool di utenti

Le seguenti quote si applicano al volume complessivo di richieste a un dominio del pool di utenti.

Operation	Description	Quota predefinita (RPS)	Regolabile
Richieste dall'IP di origine	Volume di richieste da un indirizzo IP a un dominio	300	No
Richieste al client dell'app	Volume di richieste per un ID client dell'app in un dominio	300	No
Richieste al dominio	Volume complessivo di richieste per i servizi di un dominio con pool di utenti	500	No
Richieste di documenti chiave web JSON	Volume di richieste «`jwks.json` tutto Account AWS in uno» Regione AWS	50.000	No

Frequenza delle richieste di operazione API dei pool di identità di Amazon Cognito (identità federate)

Operation	Description	Quota predefinita (RPS)[1]	Regolabile	Idoneità all'incremento delle quote
`GetId`	Recupera un ID di identità da un pool di identità.	25	Sì	Contatta il team del tuo account.
`GetOpenIdToken`	Recupera un token OpenID da un pool di identità nel flusso di lavoro classico.	200	Sì	Contatta il team del tuo account.
`GetCredentialsForIdentity`	Recupera AWS le credenziali da un pool di identità nel flusso di lavoro avanzato.	200	Sì	Contatta il team del tuo account.
`GetOpenIdTokenForDeveloperIdentity`	Recupera un token OpenID da un pool di identità nel flusso di lavoro per gli sviluppatori.	50	Sì	Contatta il team del tuo account.
`ListIdentities`	Recupera un elenco di ID identità in un pool di identità.	5	Sì	Contatta il team del tuo account.

Operation	Description	Quota predefinita (RPS)[1]	Regolabile	Idoneità all'incremento delle quote
`DeleteIdentities`	Elimina una o più identità registrate da un pool di identità.	10	Sì	Contatta il team del tuo account.
`TagResource`	Applica un tag a un pool di identità.	5	Sì	Contatta il team del tuo account.
`UntagResource`	Rimuovi un tag da un pool di identità.	5	Sì	Contatta il team del tuo account.
`ListTagsForResource`	Visualizza un elenco dei tag applicati a un pool di identità.	10	Sì	Contatta il team del tuo account.

[1] La quota predefinita è la quota minima di richieste per i pool di identità di qualsiasi Regione AWS paese dell'Unione. Account AWS La quota RPS potrebbe essere più alta in alcune regioni.

Quote relative al numero e alla dimensione delle risorse

Le quote delle risorse sono il numero o la dimensione massima di risorse, campi di input, durata temporale e altre funzionalità varie in Amazon Cognito.

È possibile richiedere un adeguamento di alcune quote delle risorse nella console Service Quotas o da un modulo di incremento dei limiti di servizio. Per richiedere un aumento delle quote utilizzando la console Service Quotas, consulta Richiesta di aumento di una quot nella Guida per l'utente di Service Quotas. Se la quota non è disponibile in Service Quotas, utilizza il modulo di incremento dei limiti di servizio.

> **Note**
>
> Le quote di risorse a Account AWS livello, ad esempio i pool di utenti per regione, si applicano alle risorse di Amazon Cognito in ciascuna regione. Regione AWS Ad esempio, si potrebbe disporre di 1.000 pool di utenti negli Stati Uniti orientali (Virginia settentrionale) e altri 1.000 in Europa (Stoccolma).

Le tabelle seguenti indicano le quote delle risorse predefinite e se sono regolabili.

Quote di risorse dei pool di utenti di Amazon Cognito

Le quote seguenti descrivono il numero o la lunghezza massima di elementi che puoi creare nei pool di utenti.

Risorsa	Quota	Regolabile	Quota massima
Client app per bacino d'utenza	1.000	Sì	10.000
Pool di utenti per regione	1.000	Sì	10.000
Provider di identità per pool di utenti	300	Sì	1.000
Server delle risorse per pool di utenti	100	Sì	300
Utenti per bacino d'utenza	40.000.000	Sì	Contatta il team del tuo account.
Modifiche combinate totali nel trigger Lambda di pre-generazione di token [1]	5.000	Sì	Contatta il team del tuo account.

Risorsa	Quota	Regolabile	Quota massima
Stili di branding di accesso gestiti per pool di utenti	20	No	N/A
Documenti con termini di accesso gestiti per pool di utenti	40	No	N/A
Attributi personalizzati per bacino d'utenza	50	No	N/A
Caratteri per attributo	2.048 byte	No	N/A
Caratteri per il nome di un attributo personalizzato	20	No	N/A
Caratteri minimi richiesti per la password nella policy per la password	6–99	No	N/A
Messaggi di posta elettronica inviati giornalmente per Account AWS[2]	50	No	N/A
Messaggi MFA via e-mail inviati a un indirizzo e-mail ogni ora per indirizzo IP del richiedente	5-20	No	N/A
Caratteri nell'oggetto dell'e-mail	140	No	N/A

Risorsa	Quota	Regolabile	Quota massima
Caratteri nel messaggio e-mail	20.000	No	N/A
Caratteri nel messaggio di verifica SMS	140	No	N/A
Caratteri nella password	256	No	N/A
Caratteri per il nome del provider di identità	32	No	N/A
Caratteri in una risposta SAML	100.000	No	N/A
Identificatori per provider di identità	50	No	N/A
Identità collegate a un utente	5	No	N/A
Passkey/WebAuthn autenticatori per utente	20	No	N/A
URL di callback per client di app	100	No	N/A
URL di disconnessione per client di app	100	No	N/A
Ambiti per server di risorse	100	No	N/A
Ambiti per client di app	50	No	N/A

Risorsa	Quota	Regolabile	Quota massima
Domini personalizzati per regione	4	No	N/A
Gruppi a cui ogni utente può appartenere	100	No	N/A
Gruppi per pool di utenti	10.000	No	N/A

[1] Questa quota potrebbe essere riscontrata nei token di un Trigger Lambda di pre-generazione del token. Il numero di rivendicazioni esistenti e aggiunte più gli ambiti di accesso e i token di identità in una transazione deve sommarsi a un numero inferiore o uguale a questa quota. Le attestazioni e gli ambiti soppressi non contribuiscono a questa quota.

[2] Questa quota vale solo se utilizzi la funzionalità e-mail predefinita per un pool di utenti di Amazon Cognito. Per un volume di consegna e-mail più elevato, configura il pool di utenti in modo da utilizzare la configurazione e-mail di Amazon SES. Questa restrizione viene ripristinata ogni giorno alle 09:00 UTC. Per ulteriori informazioni, consulta Impostazioni e-mail per i bacini d'utenza di Amazon Cognito.

Parametri di validità della sessione dei bacini d'utenza di Amazon Cognito

Le quote seguenti descrivono le impostazioni disponibili per la durata degli artefatti di autenticazione e delle sessioni utente nei pool di utenti.

Token	Quota
Token ID	5 minuti - 1 giorno
Token di aggiornamento	1 ora - 3.650 giorni
Token di accesso	5 minuti - 1 giorno
Cookie di sessione per l'interfaccia utente ospitata	1 ora
Token di sessione di autenticazione	3 minuti - 15 minuti

Quote di risorse di sicurezza del codice dei pool di utenti di Amazon Cognito (non modificabili)

Le quote seguenti descrivono i periodi di tempo disponibili relativi ai codici per l'accesso, la registrazione e la reimpostazione della password.

Risorsa	Quota
Sign-up periodo di validità del codice di conferma	24 ore
Periodo di validità del codice di verifica dell'attributo utente	24 ore
Multi-factor periodo di validità del codice di autenticazione (MFA)	3–15 minuti
Periodo di validità del codice della password dimenticata	1 ora
Numero massimo di richieste `ConfirmForgotPassword` e `ForgotPassword` per utente all'ora [1]	5-20
Numero massimo di richieste `ResendConfirmationCode` per utente all'ora	5
Numero massimo di richieste `ConfirmSignUp` per utente all'ora	15
Numero massimo di richieste `ChangePassword` per utente all'ora	5
Numero massimo di richieste `GetUserAttributeVerificationCode` per utente all'ora	5
Numero massimo di richieste `VerifyUserAttribute` per utente all'ora	15

[1] Amazon Cognito valuta i fattori di rischio nella richiesta di aggiornamento delle password e assegna una quota collegata al livello di rischio valutato. Per ulteriori informazioni, consulta Come comportarsi in caso di password dimenticata.

Quote delle risorse dei processi di importazione utenti del pool di utenti di Amazon Cognito

Le quote seguenti descrivono le risorse e i limiti disponibili per i lavori di importazione degli utenti.

Risorsa	Quota	Regolabile	Quota massima
Processi di importazione utente per pool di utenti	1.000	Sì	Contatta il team del tuo account.
Numero massimo di caratteri per riga CSV di importazione utente	16,000	No	N/A
Dimensione massima dei file CSV	100 MB	No	N/A
Numero massimo di utenti per file CSV	500.000	No	N/A

Quote di risorse relative ai pool di identità di Amazon Cognito (identità federate)

Le quote seguenti descrivono il numero o la lunghezza massima di elementi che puoi creare nei pool di identità.

Risorsa	Quota	Regolabile	Quota massima
Pool di identità per account	1.000	Sì	N/A
Provider del bacino d'utenza di Amazon	50	Sì	1000

Risorsa	Quota	Regolabile	Quota massima
Cognito per pool di identità			
Lunghezza dei caratteri per il nome del pool di identità	128 byte	No	N/A
Lunghezza dei caratteri per il nome del provider di accesso	2.048 byte	No	N/A
Identità per pool di identità	Illimitato	No	N/A
Provider di identità per i quali è possibile specificare mappature di ruoli	10	No	N/A
Risultati da una singola chiamata di ricerca o elenco	60	No	N/A
Role-based regole di controllo degli accessi (RBAC)	25	No	N/A

Quote di risorse Amazon Cognito Sync

Le seguenti quote descrivono il numero o la lunghezza massima di articoli che puoi creare in Amazon Cognito Sync.

Risorsa	Quota	Regolabile	Quota massima
Set di dati per identità	20	Sì	Contatta il team del tuo account.
Record per set di dati	1,024	Sì	Contatta il team del tuo account.
Dimensione di un singolo set di dati	1 MB	Sì	Contatta il team del tuo account.
Caratteri nel nome del set di dati	128 byte	No	N/A
Attesa per una pubblicazione in blocco, dopo una richiesta eseguita correttamente	24 ore	No	N/A

Multi-Region replica per pool di utenti

Con la replica multiregionale (MRR), è possibile creare un pool di utenti di replica aggiuntivo Regione AWS per fornire funzionalità di continuità aziendale e disaster recovery per l'infrastruttura di autenticazione. Con MRR, gli utenti registrati possono continuare ad autenticarsi con le applicazioni anche quando si perde la connettività alle risorse in una regione, garantendo che le applicazioni rimangano disponibili.

Quando configuri MRR, Amazon Cognito crea pool di utenti separati con un ID pool di utenti condiviso. Ogni pool di utenti di replica ospita servizi di autenticazione per una directory di utenti condivisa. Il pool di utenti primario funge da fonte autorevole per la configurazione amministrativa e le operazioni di scrittura come la reimpostazione delle password e la registrazione degli utenti. I pool di utenti secondari non possono creare utenti. Ereditano la maggior parte delle impostazioni dal pool di utenti primario e, in uno stato di failover, possono gestire operazioni di autenticazione come l'accesso degli utenti e la generazione di token.

Cose da sapere sulla replica in più regioni

- Multi-Region la replica ha costi aggiuntivi separati e richiede che il pool di utenti disponga del piano di funzionalità Essentials o Plus. Non è possibile abilitare MRR sui pool di utenti con il piano di funzionalità Lite.
- È necessario configurare il pool di utenti con una chiave multiregionale gestita dal cliente AWS KMS prima di abilitare la replica. La chiave deve essere disponibile in tutti coloro Regioni AWS che dispongono di repliche di pool di utenti. Per ulteriori informazioni, consulta the section called “Crittografia dei dati”.
- Per una convalida coerente dei token in tutte le regioni, consigliamo di configurare il pool di utenti con un emittente aggiornato. Per ulteriori informazioni, consulta the section called “Emittente OIDC”.
- I nuovi pool di utenti secondari iniziano nello `INACTIVE` stato. Rivedi e configura le impostazioni regionali prima di attivare il pool di utenti per l'uso in produzione.
- Le configurazioni regionali possono differire tra le repliche. È possibile configurare le seguenti impostazioni in modo indipendente nelle repliche. Tutte le altre impostazioni sono impostate nel pool di utenti primario e sincronizzate automaticamente con quello secondario.
  - Configurazione della posta elettronica
  - Configurazione e-mail per le notifiche di protezione dalle minacce
  - Configurazione SMS
  - Trigger Lambda
  - Tag
  - Configurazione dell'esportazione dei log
  - AWS WAF ACL web
- La replica dei dati tra regioni potrebbe introdurre brevi ritardi. Il pool di utenti primario sincronizza le impostazioni e gli aggiornamenti delle directory utente con quello secondario e questo processo alla fine è coerente.

Limitazioni della replica in più regioni

- Non è possibile generare nuovi utenti in pool di utenti secondari, né tramite registrazione né tramite creazione da parte dell'amministratore. Gli utenti federati possono accedere a un pool di utenti secondario nello stato di failover solo se hanno precedentemente effettuato l'accesso al pool di utenti primario.

- Gli utenti non possono reimpostare le proprie password o modificare i propri profili nei pool di utenti secondari. In uno stato di failover, disabilita queste operazioni nell'interfaccia utente e rendile disponibili dopo che il controllo di integrità ha ripristinato l'accesso al pool di utenti primario.
- È possibile avere al massimo una replica secondaria in una regione aggiuntiva per directory utente. Qualsiasi pool di utenti idoneo può avere una replica secondaria.
- L'MFA TOTP non è supportata nelle repliche secondarie. Gli utenti con TOTP MFA configurato devono autenticarsi quando il pool di utenti nella regione principale sta gestendo le richieste.
- Il numero di tentativi di autenticazione basati su password prima del blocco non è sincronizzato tra le regioni. Ogni replica mantiene il proprio numero di tentativi di autenticazione non riusciti.

Configurazione della replica in più regioni

Prima di abilitare la replica in più regioni, assicurati che il pool di utenti soddisfi i prerequisiti: piano di funzionalità Essentials o Plus e chiave KMS gestita dal cliente in più regioni.

Console di gestione AWS

Per configurare la replica multiregionale per un pool di utenti

1. Accedi alla console Amazon Cognito.
2. Scegli Bacini d'utenza.
3. Scegli un pool di utenti esistente dall'elenco o crea un nuovo pool di utenti.
4. Seleziona la scheda Impostazioni.
5. Nel menu di navigazione a sinistra, scegli Multi-Regionreplica.
6. Scegli Crea un pool di utenti di replica.
7. Per Regione, seleziona il Regione AWS luogo in cui desideri creare il pool di utenti di replica.
8. Esamina il riepilogo della configurazione e scegli Crea replica.
9. Dopo aver creato la replica, rivedi le impostazioni di configurazione regionali nella tabella di confronto. Configura qualsiasi Region-specific impostazione, ad esempio la configurazione e-mail, le impostazioni SMS o i trigger Lambda, in base alle esigenze della tua regione di replica.
10. Per configurare un controllo dello stato di Route 53 per il tuo dominio, vai al menu Servizi di dominio, modifica o aggiungi un dominio personalizzato e configura un ID di controllo dello stato di Route 53.

11. Quando sei pronto a utilizzare la replica per il traffico di produzione, modifica lo stato della replica da Inattiva a Attiva.

API

Per creare un pool di utenti di replica, usa l'operazione. CreateUserPoolReplica L'esempio seguente crea una replica nella `us-west-2` regione per un pool di utenti primario in. `us-east-1`

```
{
 "UserPoolId": "us-east-1_EXAMPLE",
 "RegionName": "us-west-2",
 "UserPoolTags": {
    "Environment": "Production",
    "Application": "MyApp"
 }
}
```

La risposta include le informazioni sulla replica:

```
{
 "Replica": {
    "RegionName": "us-west-2",
    "UserPoolArn": "arn:aws:cognito-idp:us-west-2:111122223333:userpool/us-
east-1_EXAMPLE",
    "Status": "PENDING_CREATE",
    "Role": "SECONDARY"
 }
}
```

È inoltre necessario configurare il dominio per il failover. Imposta un controllo dello stato di salute in Route 53 e applicalo al tuo dominio in una UpdateUserPoolDomainrichiesta:

```
{
 "CustomDomainConfig": {
    "CertificateArn": "arn:aws:acm:us-
east-1:111122223333:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
 },
 "Domain": "auth.example.com",
 "ManagedLoginVersion": 2,
 "Routing": {
    "Failover": {
```

```
        "SecondaryRegion": "us-west-2",
        "PrimaryRoute53HealthCheckId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
 },
 "UserPoolId": "us-east-1_EXAMPLE"
}
```

Per attivare la replica per l'uso in produzione, usa l'UpdateUserPoolReplicaoperazione:

```
{
 "UserPoolId": "us-east-1_EXAMPLE",
 "RegionName": "us-west-2",
 "Status": "ACTIVE"
}
```

La risposta conferma lo stato aggiornato della replica:

```
{
 "Replica": {
    "RegionName": "us-west-2",
    "UserPoolArn": "arn:aws:cognito-idp:us-west-2:111122223333:userpool/us-
east-1_EXAMPLE",
    "Status": "ACTIVE",
    "Role": "SECONDARY"
 }
}
```

Operazioni API supportate nelle regioni secondarie

Amazon Cognito supporta un sottoinsieme di operazioni API nelle regioni secondarie. Le operazioni disponibili dipendono dallo stato della replica. Le repliche in `INACTIVE` status supportano un set limitato di operazioni di lettura e configurazione. Le repliche in `ACTIVE` status supportano operazioni aggiuntive di autenticazione e gestione delle sessioni. Le operazioni che non sono elencate qui sono disponibili solo nella regione principale.

Operazioni per le regioni secondarie INATTIVE

I pool di utenti di replica nelle regioni secondarie in `INACTIVE` stato di status consentono le seguenti operazioni API di Amazon Cognito.

- AdminGetDevice
- AdminGetUser
- AdminListDevices
- AdminListGroupsForUser
- AdminListUserAuthEvents
- AssociateWebACL
- CreateUserPoolDomain
- DeleteUserPoolDomain
- DeleteUserPoolReplica
- DescribeIdentityProvider
- DescribeManagedLoginBranding
- DescribeManagedLoginBrandingByClient
- DescribeResourceServer
- DescribeRiskConfiguration
- DescribeTerms
- DescribeUserPool
- DescribeUserPoolClient
- DescribeUserPoolDomain
- DisassociateWebACL
- GetGroup
- GetLogDeliveryConfiguration
- GetSigningCertificate
- GetUICustomization
- GetUserPoolMfaConfig
- ListIdentityProviders
- ListGroups
- ListResourceServers
- ListTagsForResource

- ListTerms
- ListUserPoolClients
- ListUserPoolClientSecrets
- ListUserPoolReplicas
- ListUserPools
- ListUsers
- ListUsersInGroup
- SetLogDeliveryConfiguration
- SetRiskConfiguration
- SetUserPoolMfaConfig
- TagResource
- UntagResource
- UpdateUserPool
- UpdateUserPoolDomain
- UpdateUserPoolReplica

Operazioni aggiuntive per le regioni secondarie ACTIVE

I pool di utenti di replica nelle regioni secondarie in `ACTIVE` stato di status consentono tutte le operazioni precedenti, oltre alle seguenti operazioni di autenticazione e gestione delle sessioni.

- AdminInitiateAuth
- AdminRespondToAuthChallenge
- AdminUpdateAuthEventFeedback
- AdminUserGlobalSignOut
- GetDevice
- GetIdentityProviderByIdentifier
- GetTokensFromRefreshToken
- GetUser
- GetUserAuthFactors

- GlobalSignOut
- InitiateAuth
- ListDevices
- ListWebAuthnCredentials
- RespondToAuthChallenge
- RevokeToken
- UpdateAuthEventFeedback

Failover in pool di utenti multiregionali

Con i pool di utenti multiregionali, è possibile eseguire il failover di accesso gestito, accesso federato e chiamate API dirette tra due. Regioni AWS L'accesso gestito e il failover federativo sono disponibili con un dominio personalizzato o un dominio con prefisso (Cognito) configurato con il pool di utenti. Non è possibile configurare un dominio personalizzato diverso con pool di utenti di replica.

> **Note**
>
> Amazon Cognito attualmente non supporta il failover automatico con un controllo dello stato di Route 53 nelle regioni AWS GovCloud (Stati Uniti). Per informazioni sull'instradamento del traffico verso la regione secondaria negli AWS GovCloud Stati Uniti, consulta. the section called “Failover in AWS GovCloud Regioni (Stati Uniti)”

Failover per accesso gestito, federazione e autorizzazione da macchina a macchina

Il failover è disponibile quando il pool di utenti principale ha un dominio personalizzato o un dominio con prefisso. Il dominio del pool di utenti serve le risorse OAuth 2.0, inclusi gli endpoint di autorizzazione e token, e gestisce le risposte IdP di provider federativi di terze parti, tra cui OIDC, SAML e provider social.

Per abilitare il failover, configura un controllo dello stato in Route 53 e imposta il campo sul tuo dominio. `Routing` Sei tu a determinare cosa innesca uno stato sano o malsano. Quando il controllo dello stato non è integro, Amazon Cognito fornisce pagine di accesso gestite e operazioni di autenticazione dal pool di utenti di replica secondaria. Quando il controllo di integrità entra in uno stato integro, Amazon Cognito inizia a reindirizzare il traffico verso la replica principale.

Per i domini personalizzati, il record DNS può utilizzare Route 53 o qualsiasi provider DNS di terze parti. Assicurati di avere un record CNAME valido nel tuo provider DNS che punti all'alias di destinazione, che è una distribuzione. CloudFront Puoi trovare la destinazione dell'alias nella pagina Dominio della console Amazon Cognito.

Per aggiornare l'ID del controllo sanitario nella console

1. Accedi alla console Amazon Cognito.
2. Scegli Pool di utenti, quindi scegli il tuo pool di utenti.
3. Scegli Dominio sotto Branding dal menu.
4. Nella sezione Dominio personalizzato, scegli l'opzione di modifica e seleziona Modifica failover multiregionale.
5. Attiva l'opzione Abilita failover multiregionale.
6. Seleziona l'ID del tuo test sanitario Route 53 tra i controlli sanitari disponibili.
7. Scegli Save changes (Salva modifiche).

Failover per API e SDK di Amazon Cognito

Se utilizzi le API o gli SDK di Amazon Cognito, non viene utilizzato un dominio personalizzato e l'applicazione è responsabile dell'instradamento del traffico verso l'endpoint regionale del servizio Amazon Cognito per gestire l'autenticazione e altre chiamate API.

Se disponi solo di un'applicazione frontend che utilizza un client pubblico, ad esempio un'applicazione a pagina singola (SPA) o un'app mobile, l'applicazione deve essere dinamica per instradare le chiamate API di conseguenza. Prendi in considerazione un back-end applicativo serverless per determinare da quale regione iniziare l'autenticazione con Amazon Cognito.

Se disponi di un'applicazione con un back-end, puoi determinare qui la logica per determinare con quale pool di utenti eseguire l'autenticazione.

Se utilizzi sia endpoint di accesso gestiti che API, utilizza lo stesso controllo di integrità di Route 53 per determinare a quale regione l'applicazione indirizza le chiamate API di Amazon Cognito.

Failover in AWS GovCloud Regioni (Stati Uniti)

Amazon Cognito attualmente non supporta il failover automatico con un controllo dello stato di Route 53 nelle regioni AWS GovCloud (Stati Uniti). Non puoi impostare il `Routing` campo su un dominio

con pool di utenti e i domini personalizzati non sono disponibili. Per continuare a fornire endpoint gestiti di accesso, federazione e OAuth 2.0 dalla regione secondaria, configura un dominio con prefisso per ogni pool di utenti di replica e indirizza tu stesso il traffico tra i domini.

Completa i seguenti passaggi prima di eseguire il failover. Prima di iniziare, crea un pool di utenti di replica nella regione secondaria.

Per prepararsi al failover in AWS GovCloud Regioni (Stati Uniti)

1. Nella regione secondaria, crea un dominio con prefisso per il pool di utenti di replica con accesso gestito. In una `CreateUserPoolDomain` richiesta, imposta `ManagedLoginVersion` o scegli Accesso gestito nel menu Dominio nella console Amazon Cognito. 2 I pool di utenti di replica non supportano i domini con la classica interfaccia utente ospitata. Ogni dominio ha un Region-specific URL.
2. Se il tuo pool di utenti dispone di provider di identità federati (IdPs), aggiungi l'URL di reindirizzamento del dominio secondario a ciascun IdP. Per OIDC e social IdPs, aggiungilo `https://`*`secondary-domain`*`/oauth2/idpresponse` come URI di reindirizzamento consentito. Per SAML IdPs, aggiungi `https://`*`secondary-domain`*`/saml2/idpresponse` come URL ACS (Assertion Consumer Service). Se salti questo passaggio, l'accesso federato non riesce dopo aver trasferito il traffico al dominio secondario.
3. Configura l'applicazione per inviare richieste agli endpoint di autorizzazione, token, logout e ad altri endpoint OAuth 2.0 sul dominio nella regione che serve il traffico. Includi i client machine-to-machine (M2M) che richiedono token con la concessione delle credenziali del client.
4. Prima di inviare il traffico alla regione secondaria, modifica lo stato del pool di utenti di replica su. `ACTIVE`

Quando indirizzate il traffico verso il dominio secondario, tenete presente quanto segue:

- Passkey: le passkey utilizzate dagli utenti nel dominio primario non funzionano nel dominio secondario, poiché l'ID del relying party è diverso per ogni dominio. Per accedere all'accesso gestito sul dominio secondario, gli utenti devono utilizzare un altro metodo di accesso. Ciò non influisce sulle applicazioni che utilizzano gli SDK e impostano il proprio dominio come ID del relying party.
- Failback: Amazon Cognito non indirizza il traffico verso la regione principale. Quando la regione principale sarà nuovamente disponibile, aggiorna l'applicazione per inviare traffico al dominio principale.

Per informazioni sul routing del traffico API e SDK, consulta. the section called “Failover per API e SDK di Amazon Cognito”

Assegnazione di tag alle risorse Amazon Cognito

Un tag è un'etichetta di metadati che tu o AWS assegni a una AWS risorsa. Ciascun tag è formato da una chiave e da un valore, Per i tag assegnati da te, puoi definire la chiave e il valore. Ad esempio, potresti definire la chiave come `stagee` il valore di una risorsa come `test`.

I tag consentono di:

- Identifica e organizza le tue risorse. AWS Molti AWS servizi supportano l'etichettatura, in modo da poter assegnare lo stesso tag a risorse di servizi diversi. Questo ti aiuta a indicare quali risorse sono correlate. Ad esempio, puoi assegnare a un bacino d'utenza Amazon Cognito lo stesso tag che si assegna a una tabella Amazon DynamoDB.
- Tieni traccia dei costi. AWS Puoi attivare questi tag sulla Gestione costi e fatturazione AWS dashboard. AWS utilizza i tag di allocazione dei costi per classificare i costi e fornirti un rapporto mensile sull'allocazione dei costi. Per ulteriori informazioni, consulta la pagina sull'utilizzo dei tag per l'allocazione dei costi nella AWS Billing Guida per l'utente.
- Controllare gli accessi alle risorse in base ai tag a loro assegnati. Puoi controllare gli accessi specificando chiavi e valori di tag nelle condizioni di una policy AWS Identity and Access Management (IAM). Ad esempio, puoi consentire a un utente di aggiornare un pool di utenti, ma solo se questo dispone di un tag `owner` con un valore del nome di tale utente. Per ulteriori informazioni, consulta Controllo degli accessi tramite tag nella Guida per l'utente di IAM.

Puoi utilizzare l'API Amazon Cognito AWS Command Line Interface o l'API Amazon Cognito per aggiungere, modificare o eliminare tag per pool di utenti e identità. In particolare per i bacini d'utenza, puoi anche gestire i tag utilizzando la console di Amazon Cognito.

Per suggerimenti sull'utilizzo dei tag, consulta il post AWS strategie di assegnazione di tag nel blog AWS Risposte.

Nelle sezioni seguenti vengono fornite ulteriori informazioni sui tag per Amazon Cognito.

Risorse supportate in Amazon Cognito

Le seguenti risorse in Amazon Cognito supportano l'assegnazione di tag:

- Bacini d'utenza
- Pool di identità

Limitazioni applicate ai tag

Le seguenti restrizioni si applicano ai tag sulle risorse di Amazon Cognito:

- Numero massimo di tag che è possibile assegnare a una risorsa: 50
- Lunghezza massima della chiave: 128 caratteri Unicode
- Lunghezza massima del valore: 256 caratteri Unicode
- Caratteri validi per chiavi e valori: a-z A-Z, 0-9, spazio e i seguenti caratteri: _.:/= + - @
- Chiavi e valori distinguono tra maiuscole e minuscole
- Non utilizzare `aws:` come prefisso per le chiavi; l'utilizzo di questo prefisso è riservato per AWS

Gestione dei tag tramite la console di Amazon Cognito

Puoi utilizzare la console Amazon Cognito per gestire i tag assegnati ai tuoi bacini d'utenza.

Per aggiungere tag a un bacino d'utenza

1. Passa alla console di Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Impostazioni e individua la scheda Tag.
5. Scegli Add tags (Aggiungi tag) per aggiungere il primo tag. Se hai precedentemente assegnato tag a questo bacino d'utenza, in Gestisci tag seleziona Aggiungi un altro.
6. Specifica i valori per Tag Key (Chiave tag) e Tag Value (Valore tag).
7. Per ogni ulteriore tag che desideri aggiungere, seleziona Add another (Aggiungi un altro).
8. Una volta completata l'aggiunta di tag, scegli Save changes (Salva modifiche).

Per etichettare un pool di identità, vai al menu Pool di identità e seleziona o crea un pool di identità. Nella scheda delle proprietà del pool di identità, individua Tag. Seleziona Aggiungi tag.

AWS CLI esempi

AWS CLI Fornisce comandi che ti aiutano a gestire i tag che assegni ai tuoi pool di utenti e pool di identità di Amazon Cognito.

Assegnazione di tag

Utilizza i seguenti comandi per assegnare tag ai bacini d'utenza e di identità esistenti.

Example **`tag-resource`** Comando per pool di utenti

Assegna tag a un bacino d'utenza utilizzando il comando `tag-resource` all'interno del set di comandi `cognito-idp`.

```
$ aws cognito-idp tag-resource \
> --resource-arn user-pool-arn \
> --tags Stage=Test
```

Questo comando include i seguenti parametri:

- `resource-arn`: l'Amazon Resource Name (ARN) del bacino d'utenza a cui stai applicando i tag. Per trovare l'ARN, seleziona il bacino d'utenza nella console Amazon Cognito e visualizza il valore ARN del pool nella scheda Impostazioni generali.
- `tags`: le coppie chiave-valore dei tag, nel formato *`key`*=*`value`*.

Per assegnare più tag in una sola volta, specificali in un elenco separato da virgole:

```
$ aws cognito-idp tag-resource \
> --resource-arn user-pool-arn \
> --tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Example **`tag-resource`** Comando per pool di identità

Assegna tag a un pool di identità utilizzando il comando `tag-resource` all'interno del set di comandi `cognito-identity`.

```
$ aws cognito-identity tag-resource \
> --resource-arn identity-pool-arn \
> --tags Stage=Test
```

Questo comando include i seguenti parametri:

- `resource-arn`: l'Amazon Resource Name (ARN) del pool di identità a cui stai applicando i tag. Per cercare l'ARN, seleziona il pool di identità nella console Amazon Cognito e scegli Modifica pool di identità. Quindi, in Identity pool ID (ID pool di identità), seleziona Show ARN (Mostra ARN).
- `tags`: le coppie chiave-valore dei tag, nel formato *`key`*=*`value`*.

Per assegnare più tag in una sola volta, specificali in un elenco separato da virgole:

```
$ aws cognito-identity tag-resource \
> --resource-arn identity-pool-arn \
> --tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Visualizzazione dei tag

Utilizza i seguenti comandi per visualizzare i tag che hai assegnato ai bacini d'utenza e di identità.

Example **`list-tags-for-resource` Comando** per pool di utenti

Visualizza i tag assegnati a un bacino d'utenza utilizzando il comando `list-tags-for-resource` all'interno del set di comandi `cognito-idp`:

```
$ aws cognito-idp list-tags-for-resource --resource-arn user-pool-arn
```

Example **`list-tags-for-resource`** Comando per pool di identità

Visualizza i tag assegnati a un pool di identità utilizzando il comando `list-tags-for-resource` all'interno del set di comandi `cognito-identity`.

```
$ aws cognito-identity list-tags-for-resource --resource-arn identity-pool-arn
```

Rimozione dei tag

Utilizza i seguenti comandi per rimuovere tag dai bacini d'utenza e di identità.

Example Comando **`untag-resource`** per pool di utenti

Rimuovi tag da un bacino d'utenza utilizzando il comando `untag-resource` all'interno del set di comandi `cognito-idp`:

```
$ aws cognito-idp untag-resource \
> --resource-arn user-pool-arn \
> --tag-keys Stage CostCenter Owner
```

Per il parametro `--tag-keys`, specifica una o più chiavi di tag. Non includere i valori di tag. Separa le chiavi con spazi.

Example Comando **`untag-resource`** per pool di identità

Rimuovi tag da un pool di identità utilizzando il comando `untag-resource` all'interno del set di comandi `cognito-identity`:

```
$ aws cognito-identity untag-resource \
> --resource-arn identity-pool-arn \
> --tag-keys Stage CostCenter Owner
```

Per il parametro `--tag-keys`, specifica una o più chiavi di tag. Non includere i valori di tag.

> **Important**
>
> Dopo aver eliminato un utente o un pool di identità, i tag correlati al pool eliminato possono ancora apparire nella console o nelle chiamate API fino a 30 giorni dopo l'eliminazione.

Applicazione di tag durante la creazione delle risorse

Utilizza i seguenti comandi per assegnare tag al momento della creazione di un bacino d'utenza o un pool di identità.

Example Comando **`create-user-pool`** con tag

Quando crei un bacino d'utenza utilizzando il comando `create-user-pool`, puoi specificare tag con il parametro `--user-pool-tags`:

```
$ aws cognito-idp create-user-pool \
> --pool-name user-pool-name \
> --user-pool-tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Key-value le coppie per i tag devono essere nel formato. *`key`*=*`value`* Se aggiungi più tag, specificali in un elenco separato da virgole.

Example **`create-identity-pool`** Comando con tag

Quando crei un pool di identità utilizzando il comando `create-identity-pool`, puoi specificare tag con il parametro `--identity-pool-tags`:

```
$ aws cognito-identity create-identity-pool \
> --identity-pool-name identity-pool-name \
> --allow-unauthenticated-identities \
> --identity-pool-tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Key-value le coppie per i tag devono essere nel formato. *key*=*value* Se aggiungi più tag, specificali in un elenco separato da virgole.

Gestione dei tag tramite l'API di Amazon Cognito

Puoi utilizzare le seguenti operazioni nell'API di Amazon Cognito per gestire i tag dei bacini d'utenza e dei pool di identità.

Operazioni API per i tag del bacino d'utenza

Utilizza le seguenti operazioni API per assegnare, visualizzare e rimuovere tag per i bacini d'utenza.

- TagResource
- ListTagsForResource
- UntagResource
- CreateUserPool

Operazioni API per i tag del pool di identità

Utilizza le seguenti operazioni API per assegnare, visualizzare e rimuovere tag per i pool di identità.

- TagResource
- ListTagsForResource
- UntagResource
- CreateIdentityPool

CloudFormation autorizzazioni di etichettatura

Quando crei o aggiorni un pool di utenti o risorse del pool di identità di Amazon Cognito con AWS CloudFormation, CloudFormation puoi propagare tag a livello di stack e tag a livello di sistema alle tue risorse. Per consentire questa propagazione dei tag, il responsabile IAM che distribuisce lo stack deve disporre delle autorizzazioni di tagging oltre alle autorizzazioni per la creazione di risorse.

Ad esempio, per creare un pool di utenti con`cognito-idp:CreateUserPool`, il chiamante deve disporre anche delle seguenti autorizzazioni:

- `cognito-idp:TagResource`
- `cognito-idp:UntagResource`
- `cognito-idp:ListTagsForResource`

Analogamente, per creare un pool di identità con`cognito-identity:CreateIdentityPool`, il chiamante deve disporre anche delle seguenti autorizzazioni:

- `cognito-identity:TagResource`
- `cognito-identity:UntagResource`
- `cognito-identity:ListTagsForResource`

Senza queste autorizzazioni, la creazione o l'aggiornamento dello stack possono fallire quando si CloudFormation tenta di applicare tag alla risorsa. Se vedi un errore di accesso negato relativo al tagging durante un'operazione di stack, verifica che la tua policy IAM includa queste autorizzazioni.

Risoluzione dei problemi relativi ad Amazon Cognito

Questo capitolo fornisce soluzioni ai problemi più comuni che potresti incontrare mentre lavori con Amazon Cognito. Le implementazioni di Amazon Cognito possono affrontare diverse sfide relative ai flussi di autenticazione, alle configurazioni dei pool di utenti e alle configurazioni di federazione delle identità. Che tu stia sviluppando una nuova applicazione o gestendo una esistente, questa guida alla risoluzione dei problemi ti aiuta a identificare e risolvere rapidamente i problemi più comuni.

Errori di configurazione del dominio personalizzato

Quando configuri nomi di dominio personalizzati in Amazon Cognito, potresti ricevere messaggi di errore. Gli errori più comuni includono problemi di convalida, problemi con i certificati o conflitti di dominio.

`Il dominio personalizzato non è un sottodominio valido`

Questo errore indica un problema con la risoluzione DNS per il dominio principale. Amazon Cognito non supporta i domini di primo livello e richiede che il dominio principale disponga di un record DNS A per la convalida.

Problema

Questo errore indica un problema con la risoluzione DNS per il dominio principale. Amazon Cognito non supporta i domini di primo livello e richiede che il dominio principale disponga di un record DNS A per la convalida.

Soluzione

- Crea un record A per il dominio principale: devi creare un record A nella configurazione DNS per il dominio principale del tuo dominio personalizzato.
  - Esempio: se il tuo dominio personalizzato è`auth.xyz.yourdomain.com`, il dominio principale è`xyz.yourdomain.com`. Se desideri configurarlo `xyz.yourdomain.com` come dominio personalizzato, il dominio principale è`yourdomain.com`.
  - Il dominio principale deve essere risolto in un indirizzo IP valido. Se non punta a un indirizzo IP reale, puoi utilizzare un indirizzo IP fittizio, ad esempio`8.8.8.8`.
- Verifica della propagazione DNS (facoltativa ma consigliata): per assicurarti che il tuo provider DNS abbia propagato la modifica, puoi eseguire un comando. `dig`
  - Se si utilizza `auth.xyz.yourdomain.com` come dominio personalizzato: `dig A xyz.yourdomain.com +short`
  - Se si utilizza `xyz.yourdomain.com` come dominio personalizzato: `dig A yourdomain.com +short`
  - Il comando dovrebbe restituire l'indirizzo IP configurato. In caso contrario, attendi che la modifica sia stata completamente propagata.
- Rimuovi il record A del dominio principale: dopo aver creato correttamente il dominio personalizzato in Amazon Cognito, puoi rimuovere il record A che hai creato per il dominio principale se si trattava di un record fittizio.

Per ulteriori informazioni, consulta Usare il proprio dominio per l'interfaccia utente ospitata.

Dominio già associato a un altro pool di utenti

I nomi di dominio personalizzati devono essere unici in tutte Account AWS le regioni.

Problema

I nomi di dominio personalizzati devono essere unici in tutte Account AWS le regioni.

Soluzione

- Per utilizzare il nome di dominio per un nuovo pool di utenti, è necessario eliminare il dominio personalizzato dal pool di utenti a cui è attualmente associato.
- Attendi dopo l'eliminazione: l'eliminazione completa del dominio personalizzato dal primo pool di utenti richiede tempo. La creazione di un nuovo dominio personalizzato con lo stesso nome subito dopo l'eliminazione potrebbe comunque causare questo errore. Attendi qualche minuto prima di riprovare.

Uno o più dei CName che hai fornito sono già associati a una risorsa diversa

Quando crei un dominio personalizzato, Amazon Cognito crea una distribuzione Amazon AWS CloudFront gestita. Un nome di dominio può essere utilizzato solo con una CloudFront distribuzione Amazon. Questo errore si verifica se il nome di dominio è già in uso come dominio alternativo per un'altra CloudFront distribuzione Amazon.

Problema

Quando crei un dominio personalizzato, Amazon Cognito crea una distribuzione Amazon AWS CloudFront gestita. Un nome di dominio può essere utilizzato solo con una CloudFront distribuzione Amazon. Questo errore si verifica se il nome di dominio è già in uso come dominio alternativo per un'altra CloudFront distribuzione Amazon.

Soluzione

- Opzione 1: usa un nome di dominio diverso per il tuo dominio personalizzato Amazon Cognito.
- Opzione 2: se utilizzi il nome di dominio per Amazon Cognito, non utilizzarlo con un'altra distribuzione Amazon CloudFront .

Il certificato SSL specificato non esiste

Amazon Cognito utilizza Amazon CloudFront, che richiede che il certificato AWS Certificate Manager (ACM) si trovi nella `us-east-1` (Virginia settentrionale) Regione AWS, indipendentemente dalla regione del pool di utenti.

Problema

Amazon Cognito utilizza Amazon CloudFront, che richiede che il certificato AWS Certificate Manager (ACM) si trovi nella `us-east-1` (Virginia settentrionale) Regione AWS, indipendentemente dalla regione del pool di utenti.

Soluzione

- Verifica la regione del certificato: conferma che il certificato ACM si trova nella regione. `us-east-1`
- Verifica la validità del certificato: assicurati che il certificato selezionato non sia scaduto.
- Certificati importati: se hai importato il certificato in ACM, assicurati che:
  - È stato rilasciato da un'autorità di certificazione pubblica.
  - Include la catena di certificati corretta.
- AWS KMS Controllo delle politiche: un'`deny`istruzione esplicita in una policy AWS Key Management Service (AWS KMS) per l'utente o il ruolo IAM che crea il dominio può causare questo errore. In particolare, verifica la presenza di negazioni o azioni esplicite. `kms:DescribeKey` `kms:CreateGrant` `kms:*`

Errore del token di aggiornamento non valido

Problema

Ricevi un `Invalid refresh token` errore quando tenti di utilizzare un token di aggiornamento per ottenere nuovi token di accesso e ID dal tuo pool di utenti Amazon Cognito utilizzando `AdminInitiateAuth` l'`InitiateAuth`operazione o API.

Soluzione

Implementa i seguenti passaggi per la risoluzione dei problemi in base alla configurazione del pool di utenti e all'utilizzo dell'API:

- Garantisci un ID client dell'app coerente: quando chiami l'`InitiateAuth`API `AdminInitiateAuth` o l'API per l'aggiornamento del token, devi utilizzare esattamente lo

stesso ID client dell'app utilizzato durante l'autenticazione iniziale che ha generato il token di aggiornamento.

- Conferma il dispositivo: se hai abilitato il tracciamento dei dispositivi nel tuo pool di utenti, ma il dispositivo dell'utente non è stato confermato, devi prima chiamare l'ConfirmDeviceAPI. Dopo che l'utente ha confermato il dispositivo, puoi scambiare il token di aggiornamento.
- Includi la chiave del dispositivo nella richiesta di aggiornamento: se il tracciamento del dispositivo è abilitato, includi la chiave univoca del dispositivo come elemento `AuthParameter` durante l'utilizzo del `REFRESH_TOKEN_AUTH` flusso:

```
{
  "AuthFlow": "REFRESH_TOKEN_AUTH",
  "AuthParameters": {
    "REFRESH_TOKEN": "example_refresh_token",
    "SECRET_HASH": "example_secret_hash", // Required if your app client uses a
 client secret
    "DEVICE_KEY": "example_device_key"
  }
}
```

- Utilizzo **`USER_SRP_AUTH`** per il tracciamento dei dispositivi: se utilizzi il tracciamento dei dispositivi, il flusso di autenticazione iniziale deve essereUSER_SRP_AUTH.

Per ulteriori informazioni, consulta Utilizzo dei dispositivi utente nel pool di utenti.

Errori di risposta SAML non validi nella federazione

Gli utenti ricevono errori diversi `Invalid SAML response` e simili quando tentano di eseguire la federazione in Amazon Cognito utilizzando SAML 2.0. Questi errori possono verificarsi a causa di problemi di mappatura degli attributi, problemi di certificati o mancate corrispondenze di configurazione.

`Attributi utente non validi: attributo obbligatorio`

Problema

A un utente manca un valore per un attributo richiesto nel pool di utenti oppure l'IdP sta tentando di rimuovere o aggiornare un attributo immutabile.

Soluzione

- Controlla gli attributi obbligatori definiti nella configurazione del tuo pool di utenti.

- Utilizzando gli strumenti di acquisizione di rete nel browser, recupera la risposta SAML. Potrebbe essere necessario eseguire la decodifica di URL e base64. Verifica che l'attributo sia presente nell'asserzione SAML.
- Accedi al tuo provider di identità e controlla gli attributi che sta inviando ad Amazon Cognito. Verifica che l'IdP sia configurato per inviare l'attributo richiesto utilizzando il nome corretto.
- Per gli attributi immutabili, esegui il seguente AWS CLI comando per identificarli:. `aws cognito-idp describe-user-pool --user-pool-id USER-POOL-ID --query 'UserPool.SchemaAttributes[?Mutable==`false`].Name'`
- Nelle mappature degli attributi SAML del tuo IdP, elimina qualsiasi mappatura che abbia come target un attributo Amazon Cognito immutabile. In alternativa, aggiorna l'attributo di destinazione con un altro attributo mutabile.

Risposta SAML non valida ricevuta: la firma di risposta SAML non è valida

Problema

Il tuo IdP ha aggiornato il certificato di firma SAML, causando una mancata corrispondenza tra il certificato nella risposta SAML e il file di metadati archiviato in Amazon Cognito.

Soluzione

1. Scarica il file di metadati più recente dal tuo IdP.
2. Nella console Amazon Cognito, accedi ai provider social ed esterni per il tuo pool di utenti, modifica il tuo provider SAML e sostituisci il file di metadati esistente con il file appena scaricato.

Limitazione del pubblico o applicazione con identificatore non trovata

Problema

Nel tuo IdP è configurato un Entity ID errato oppure l'asserzione utilizza l'Uniform Resource Name (URN) di un altro pool di utenti.

Soluzione

1. Ottieni l'ID del tuo pool di utenti Amazon Cognito dalla sezione Panoramica della console.

2. Nella console di gestione del tuo IdP, aggiorna l'ID dell'entità nell'applicazione SAML per il tuo pool di utenti. Configura l'ID dell'entità in modo che corrisponda al formato. `urn:amazon:cognito:sp:`*`USER_POOL_ID`* Sostituisci `USER_POOL_ID` con l'ID del pool di utenti del passaggio precedente.

Si è verificato un errore nella pagina richiesta

Problema

L'URL di Assertion Consumer Service (ACS) registrato con il tuo IdP non è configurato correttamente oppure l'IdP non invia la risposta SAML utilizzando l'associazione POST richiesta.

Soluzione

- Nella console di gestione del tuo IdP, aggiorna l'applicazione con il formato URL ACS corretto, assicurandoti che utilizzi l'associazione. `HTTP POST`
- Formato di dominio predefinito: `https://cognito-idp.`*`Region`*`.amazonaws.com/`*`your user pool ID`*`/saml2/idpresponse`
- Formato di dominio personalizzato: `https://`*`auth.example.com`*`/saml2/idpresponse`

RelayState non valido dal provider di identità

Problema

Il `RelayState` parametro è mancante o non valido oppure l'URL non corrisponde tra l'IdP e Amazon Cognito.

Soluzione

- Per i flussi iniziati dal fornitore di servizi (SP-initiated): avvia sempre l'autenticazione sull'endpoint del tuo pool di utenti. `/oauth2/authorize` Le asserzioni SAML che non restituiscono `RelayState` i parametri della visita iniziale dell'utente ad Amazon Cognito non sono richieste valide. SP-initiated
- Per i flussi iniziati dal provider di identità (IdP-initiated): l'IdP deve includere `RelayState` il parametro con l'asserzione SAML all'endpoint, utilizzando `/saml2/idpresponse` il formato richiesto:. `redirect_uri=REDIRECT_URI&state=STATE`

Per ulteriori informazioni, consulta Utilizzo di provider di identità SAML con un pool di utenti.

Gli utenti con accesso gestito non possono selezionare un fattore MFA

Problema

Gli utenti non possono scegliere il metodo MFA preferito quando accedono tramite accesso gestito, oppure non viene loro richiesto il metodo MFA previsto.

Soluzione

Amazon Cognito segue una logica specifica per determinare quali fattori MFA sono disponibili per gli utenti in base alle impostazioni del pool di utenti, agli attributi utente e alla configurazione di ripristino dell'account. Ad esempio, gli utenti non possono utilizzare l'MFA della posta elettronica se l'e-mail è configurata come metodo di ripristino dell'account principale.

Per ignorare la selezione del fattore MFA predefinita, puoi implementare uno di questi approcci:

- Per le applicazioni pubbliche: utilizzare SetUserMFAPreferencecon un token di accesso valido per consentire agli utenti di impostare le proprie preferenze MFA
- Per le applicazioni gestite dall'amministratore: utilizzare AdminSetUserMFAPreferencecon le AWS credenziali per configurare le preferenze MFA dell'utente

Entrambe le operazioni consentono di abilitare o disabilitare i metodi MFA di SMS, e-mail e TOTP per i singoli utenti e di impostare un metodo come preferito.

Per ulteriori informazioni, consulta Aggiunta dell'autenticazione MFA a un bacino d'utenza.

Gli utenti senza password e passkey non possono utilizzare l'MFA

Problema

Gli utenti che accedono con metodi di autenticazione monouso (OTP) non possono aggiungere o utilizzare l'autenticazione a più fattori.

Soluzione

I flussi di accesso OTP non supportano l'autenticazione a più fattori. Tuttavia, l'autenticazione con chiave di accesso con verifica dell'utente può soddisfare i requisiti MFA. Imposta su `FactorConfiguration` `MULTI_FACTOR_WITH_USER_VERIFICATION` nel tuo pool di utenti `WebAuthnConfiguration` per consentire alle passkey di contare come autenticazione a più fattori.

Per ulteriori informazioni, consulta Autenticazione con pool di utenti.

Non è possibile ricevere il codice di reimpostazione della password tramite email/SMS

Problema

Gli utenti non possono ricevere i codici di verifica durante il flusso di lavoro relativo alla password dimenticata tramite e-mail o SMS.

Soluzione

Esistono vari motivi per cui i codici di verifica potrebbero non essere ricevuti. Segui questa lista di controllo per risolvere il problema:

- Controlla le cartelle di posta indesiderata e posta indesiderata dell'utente.
- Verifica che l'utente esista nel pool di utenti.
- Verifica che lo stato dell'utente non lo sia`FORCE_CHANGE_PASSWORD`. In tal caso, l'utente è stato creato da un amministratore e Amazon Cognito gli richiede di impostare una password quando accede con la password temporanea.
- Verifica che l'utente disponga di un attributo e-mail o numero di telefono verificato.
- Verifica il limite di spesa del tuo account per gli SMS
- Controlla la quota di invio di messaggi di Amazon Simple Email Service (Amazon SES).
- Verifica se sia SMS che Amazon SES (se il tuo pool di utenti è configurato per Inviare e-mail con Amazon SES) sono stati spostati fuori dalla sandbox. Se non sono stati rimossi dalla sandbox, indirizzi e-mail o numeri di telefono che non sono stati verificati da Amazon SES o che non AWS End User Messaging SMS saranno in grado di ricevere codici di reimpostazione della password.
- Se l'utente ha un fattore MFA attivo, verifica che non stia cercando di generare un messaggio sullo stesso fattore. Ad esempio, gli utenti con MFA e-mail attivo non possono inviare codici di reimpostazione della password tramite e-mail.

Per ulteriori informazioni, consulta Impostazioni e-mail per i bacini d'utenza di Amazon Cognito e Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.

La reimpostazione della password non riesce con attributi di ripristino **non verificati: impossibile reimpostare la password per l'account, contatta l'assistenza o** riprova

Problema

Gli utenti ricevono questo errore quando tentano di reimpostare la password a causa di metodi di ripristino non verificati o conflitti di configurazione MFA. L'errore si verifica quando l'attributo e-mail o numero di telefono dell'utente non è verificato o quando le impostazioni MFA impediscono l'utilizzo del metodo di ripristino configurato.

Soluzione

Controlla la configurazione di ripristino dell'account del tuo pool di utenti e lo stato di verifica dell'utente interessato:

- Verifica le impostazioni di ripristino: nel tuo pool di utenti, vai su Accedi > Ripristino dell'account utente. Assicurati che il ripristino dell'account self-service sia abilitato e controlla l'impostazione del metodo di recapito dei messaggi di ripristino.
- Verifica la verifica degli attributi utente: verifica che l'utente abbia un indirizzo email o un numero di telefono verificato che corrisponda alla configurazione del metodo di ripristino. Nella console, vai al profilo dell'utente, seleziona Attributi utente > Modifica e contrassegna l'attributo appropriato come verificato.
- Risolvi i conflitti MFA: gli utenti il cui metodo MFA preferito è l'e-mail non possono ricevere codici di reimpostazione della password tramite e-mail e gli utenti con MFA SMS non possono ricevere codici tramite SMS. Aggiorna il metodo di recapito dei messaggi di ripristino per fornire opzioni alternative come SMS se disponibili, altrimenti e-mail o e-mail se disponibile, altrimenti SMS.
- Verifica amministrativa: utilizza l'operazione API per verificare AdminUpdateUserAttributesa livello di codice gli attributi utente quando l'accesso alla console non è disponibile.

Per ulteriori informazioni, consulta Password, ripristino dell'account e politiche relative alle password.

Errori SECRET_HASH

Problema

Le richieste API di autenticazione ai client di app con segreti client restituiscono errori come `An error occurred (NotAuthorizedException) when calling the ForgotPassword operation: Client 1example23456789 is configured with secret but SECRET_HASH was not received.`

Soluzione

L'applicazione deve calcolare `SECRET_HASH` il segreto dell'utente corrente, del client dell'app e del client. Il metodo di calcolo è:

```
Base64 ( HMAC_SHA256 ( "client secret", "Username" + "Client Id" ) )
```

Per ottenere il segreto del cliente, puoi:

1. Apri la console Amazon Cognito e accedi al client dell'app dal menu App client. Nella sezione Informazioni sul client dell'app, individua Client secret. Seleziona Mostra il segreto del client per visualizzare il segreto del client dell'app.
2. Genera una DescribeUserPoolClientrichiesta. Il segreto del client è incluso nella risposta.

Per ulteriori informazioni, consulta Calcolo dei valori SecretHash.

La console Amazon Cognito sceglie una configurazione predefinita per un nuovo pool di utenti

Problema

Quando configuri un nuovo pool di utenti nella console, Amazon Cognito sceglie diverse impostazioni predefinite per te. Alcune impostazioni non possono essere modificate dopo la creazione del pool di utenti. Come puoi fare scelte consapevoli e capire cosa Amazon Cognito ha selezionato automaticamente?

Soluzione

La nuova configurazione del pool di utenti nella console Amazon Cognito è progettata per test e prototipazione rapidi. La console offre solo le scelte di configurazione più importanti, quelle che

non possono essere modificate dopo la creazione del pool di utenti. Tutte le altre impostazioni che Amazon Cognito configura automaticamente possono essere modificate in un secondo momento.

Consigliamo quanto segue:

1. Usa la console per creare pool di utenti di prova mentre perfezioni l'implementazione.
2. Dopo aver determinato la configurazione di produzione, applica tali impostazioni ai pool di utenti di test.
3. Utilizza DescribeUserPoolle operazioni DescribeUserPoolClientAPI per generare modelli JSON della configurazione testata.
4. Usa questi modelli con strumenti di distribuzione come AWS SDK, CDK, API REST o CloudFormation per creare risorse di produzione.

Per ulteriori informazioni, consulta Crea una nuova applicazione nella console Amazon Cognito.

Altre risorse per la risoluzione dei problemi

Per ulteriori indicazioni sulla risoluzione dei problemi e soluzioni fornite dalla community, puoi anche esplorare le seguenti risorse esterne:

- AWS Community di Amazon Cognito di re:post - Sfoglia le domande e le soluzioni della community
- AWS Articoli su Amazon Cognito del Knowledge Center - Articoli selezionati per la risoluzione dei problemi

Esempi di codice per l'utilizzo di Amazon Cognito AWS SDKs

I seguenti esempi di codice mostrano come usare Amazon Cognito con un kit di sviluppo AWS software (SDK).

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, 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 Amazon Cognito Identity AWS SDKs
  - Esempi di base per l'utilizzo di Amazon Cognito Identity AWS SDKs
    - Azioni per l'utilizzo di Amazon Cognito Identity AWS SDKs
      - Utilizzo CreateIdentityPool con un AWS SDK o una CLI
      - Utilizzo DeleteIdentityPool con un AWS SDK o una CLI
      - Utilizzare DescribeIdentityPool con una CLI
      - Utilizzare GetCredentialsForIdentity con un SDK AWS
      - Utilizzare GetIdentityPoolRoles con una CLI
      - Utilizzo ListIdentityPools con un AWS SDK o una CLI
      - Utilizzare SetIdentityPoolRoles con una CLI
      - Utilizzare UpdateIdentityPool con una CLI
  - Scenari per l'utilizzo di Amazon Cognito Identity AWS SDKs
    - Creazione di un’applicazione Amazon Textract explorer
- Esempi di codice per l'utilizzo di Amazon Cognito Identity Provider AWS SDK
  - Esempi di base per l'utilizzo di Amazon Cognito Identity Provider AWS SDK
    - Ciao Amazon Cognito
    - Azioni per l'utilizzo di Amazon Cognito Identity Provider AWS SDK
      - Da utilizzare AdminCreateUsercon un AWS SDK o CLI
      - Utilizzare AdminGetUsercon un AWS SDK o CLI
      - Utilizzare AdminInitiateAuthcon un AWS SDK o CLI
      - Utilizzare AdminRespondToAuthChallengecon un AWS SDK o CLI

- Utilizzare AdminSetUserPasswordcon un AWS SDK o CLI
- Utilizzare AssociateSoftwareTokencon un AWS SDK o CLI
- Utilizzare ConfirmDevicecon un AWS SDK o CLI
- Utilizzare ConfirmForgotPasswordcon un AWS SDK o CLI
- Utilizzare ConfirmSignUpcon un AWS SDK o CLI
- Utilizzare CreateUserPoolcon un AWS SDK o CLI
- Utilizzare CreateUserPoolClientcon un AWS SDK o CLI
- Utilizzare DeleteUsercon un AWS SDK o CLI
- Utilizzare DescribeTermsByClientcon un AWS SDK
- Utilizzare DescribeUserPoolClientcon un AWS SDK o CLI
- Utilizzare ForgotPasswordcon un AWS SDK o CLI
- Utilizzare GetClientTokencon un AWS SDK
- Utilizzare InitiateAuthcon un AWS SDK o CLI
- Utilizzare ListUserPoolscon un AWS SDK o CLI
- Utilizzare ListUserscon un AWS SDK o CLI
- Utilizzare ResendConfirmationCodecon un AWS SDK o CLI
- Utilizzare RespondToAuthChallengecon un AWS SDK o CLI
- Utilizzare SignUpcon un AWS SDK o CLI
- Utilizzare UpdateUserPoolcon un AWS SDK o CLI
- Utilizzare VerifySoftwareTokencon un AWS SDK o CLI

- Scenari per l'utilizzo di Amazon Cognito Identity Provider AWS SDK
  - Conferma automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda utilizzando un AWS SDK
  - Esegui automaticamente la migrazione di utenti Amazon Cognito noti con una funzione Lambda utilizzando un AWS SDK
  - Ottieni token client M2M e scopri i termini di Amazon Cognito utilizzando un AWS SDK
  - Nozioni di base sui bacini d'utenza
  - Registra un utente con un pool di utenti Amazon Cognito che richiede l'autenticazione a più fattori utilizzando un AWS SDK
  - Usa i pool di identità e i flussi di autenticazione di Amazon Cognito

    - Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito utilizzando un AWS SDK
- Esempi di codice per l'utilizzo di Amazon Cognito Sync AWS SDKs
  - Esempi di base per l'utilizzo di Amazon Cognito Sync AWS SDKs
    - Azioni per Amazon Cognito Sync utilizzando AWS SDKs
      - Utilizzalo ListIdentityPoolUsage con un AWS SDK

Esempi di codice per l'utilizzo di Amazon Cognito Identity AWS SDKs

I seguenti esempi di codice mostrano come usare Amazon Cognito Identity 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 di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Amazon Cognito Identity

- Esempi di base per l'utilizzo di Amazon Cognito Identity AWS SDKs
  - Azioni per l'utilizzo di Amazon Cognito Identity AWS SDKs
    - Utilizzo CreateIdentityPool con un AWS SDK o una CLI
    - Utilizzo DeleteIdentityPool con un AWS SDK o una CLI
    - Utilizzare DescribeIdentityPool con una CLI
    - Utilizzare GetCredentialsForIdentity con un SDK AWS
    - Utilizzare GetIdentityPoolRoles con una CLI
    - Utilizzo ListIdentityPools con un AWS SDK o una CLI
    - Utilizzare SetIdentityPoolRoles con una CLI

    - Utilizzare UpdateIdentityPool con una CLI
- Scenari per l'utilizzo di Amazon Cognito Identity AWS SDKs
  - Creazione di un'applicazione Amazon Textract explorer

Esempi di base per l'utilizzo di Amazon Cognito Identity AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di Amazon Cognito Identity con. AWS SDKs

Esempi

- Azioni per l'utilizzo di Amazon Cognito Identity AWS SDKs
  - Utilizzo CreateIdentityPool con un AWS SDK o una CLI
  - Utilizzo DeleteIdentityPool con un AWS SDK o una CLI
  - Utilizzare DescribeIdentityPool con una CLI
  - Utilizzare GetCredentialsForIdentity con un SDK AWS
  - Utilizzare GetIdentityPoolRoles con una CLI
  - Utilizzo ListIdentityPools con un AWS SDK o una CLI
  - Utilizzare SetIdentityPoolRoles con una CLI
  - Utilizzare UpdateIdentityPool con una CLI

Azioni per l'utilizzo di Amazon Cognito Identity AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole azioni di Amazon Cognito Identity con. AWS SDKs 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 Amazon Cognito Identity 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 Amazon Cognito Identity AWS SDKs.

Gli esempi seguenti includono solo le azioni più comunemente utilizzate. Per l'elenco completo, consulta la documentazione di riferimento dell'API Amazon Cognito Identity.

Esempi

- Utilizzo CreateIdentityPool con un AWS SDK o una CLI

- Utilizzo DeleteIdentityPool con un AWS SDK o una CLI
- Utilizzare DescribeIdentityPool con una CLI
- Utilizzare GetCredentialsForIdentity con un SDK AWS
- Utilizzare GetIdentityPoolRoles con una CLI
- Utilizzo ListIdentityPools con un AWS SDK o una CLI
- Utilizzare SetIdentityPoolRoles con una CLI
- Utilizzare UpdateIdentityPool con una CLI

Utilizzo **`CreateIdentityPool`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `CreateIdentityPool`.

CLI

AWS CLI

Come creare un pool di identità con il provider del pool di identità Cognito

Questo esempio crea un pool di identità denominato MyIdentityPool. Dispone di un provider del pool di identità Cognito. Le identità non autenticate non sono consentite.

Comando:

```
aws cognito-identity create-identity-pool --identity-pool-
name MyIdentityPool --no-allow-unauthenticated-identities --cognito-
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_aaaaaaaaa",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Output:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",
```

```
            "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
            "ServerSideTokenCheck": false
        }
    ]
}
```

- Per i dettagli sull'API, vedere CreateIdentityPoolin AWS CLI Command 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.cognitoidentity.CognitoIdentityClient;
import
 software.amazon.awssdk.services.cognitoidentity.model.CreateIdentityPoolRequest;
import
 software.amazon.awssdk.services.cognitoidentity.model.CreateIdentityPoolResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc

/**
 * 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 CreateIdentityPool {
    public static void main(String[] args) {
        final String usage = """
                Usage:
                    <identityPoolName>\s
```

```
                Where:
                    identityPoolName - The name to give your identity pool.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String identityPoolName = args[0];
        CognitoIdentityClient cognitoClient = CognitoIdentityClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String identityPoolId = createIdPool(cognitoClient, identityPoolName);
        System.out.println("Unity pool ID " + identityPoolId);
        cognitoClient.close();
    }

    public static String createIdPool(CognitoIdentityClient cognitoClient, String
 identityPoolName) {
        try {
            CreateIdentityPoolRequest poolRequest =
 CreateIdentityPoolRequest.builder()
                    .allowUnauthenticatedIdentities(false)
                    .identityPoolName(identityPoolName)
                    .build();

            CreateIdentityPoolResponse response =
 cognitoClient.createIdentityPool(poolRequest);
            return response.identityPoolId();

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Per i dettagli sull'API, consulta la CreateIdentityPoolsezione AWS SDK for Java 2.x API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: crea un nuovo pool di identità che consente di utilizzare identità non autenticate.

```
New-CGIIdentityPool -AllowUnauthenticatedIdentities $true -IdentityPoolName
 CommonTests13
```

Output:

```
LoggedAt                        : 8/12/2015 4:56:07 PM
AllowUnauthenticatedIdentities  : True
DeveloperProviderName           :
IdentityPoolId                  : us-east-1:15d49393-ab16-431a-b26e-EXAMPLEGUID3
IdentityPoolName                : CommonTests13
OpenIdConnectProviderARNs       : {}
SupportedLoginProviders         : {}
ResponseMetadata                : Amazon.Runtime.ResponseMetadata
ContentLength                   : 136
HttpStatusCode                  : OK
```

- Per i dettagli sull'API, vedere CreateIdentityPool in AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: crea un nuovo pool di identità che consente di utilizzare identità non autenticate.

```
New-CGIIdentityPool -AllowUnauthenticatedIdentities $true -IdentityPoolName
 CommonTests13
```

Output:

```
LoggedAt                        : 8/12/2015 4:56:07 PM
AllowUnauthenticatedIdentities  : True
DeveloperProviderName           :
IdentityPoolId                  : us-east-1:15d49393-ab16-431a-b26e-EXAMPLEGUID3
IdentityPoolName                : CommonTests13
OpenIdConnectProviderARNs       : {}
SupportedLoginProviders         : {}
ResponseMetadata                : Amazon.Runtime.ResponseMetadata
ContentLength                   : 136
```

```
HttpStatusCode                 : OK
```

- Per i dettagli sull'API, vedere CreateIdentityPoolin AWS Strumenti per PowerShell Cmdlet Reference (V5).

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 AWSCognitoIdentity


    /// Create a new identity pool and return its ID.
    ///
    /// - Parameters:
    ///     - name: The name to give the new identity pool.
    ///
    /// - Returns: A string containing the newly created pool's ID, or `nil`
    ///   if an error occurred.
    ///
    func createIdentityPool(name: String) async throws -> String? {
        do {
            let cognitoInputCall = CreateIdentityPoolInput(developerProviderName:
"com.exampleco.CognitoIdentityDemo",
                                                           identityPoolName:
name)

            let result = try await
cognitoIdentityClient.createIdentityPool(input: cognitoInputCall)
            guard let poolId = result.identityPoolId else {
                return nil
            }

            return poolId
        } catch {
```

```
            print("ERROR: createIdentityPool:", dump(error))
            throw error
        }
    }
```

- Per ulteriori informazioni, consulta Guida per gli sviluppatori di AWS SDK per Swift.
- Per i dettagli sull'API, consulta la CreateIdentityPoolguida 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 **`DeleteIdentityPool`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DeleteIdentityPool`.

CLI

AWS CLI

Come eliminare un pool di identità

L'esempio `delete-identity-pool` seguente elimina il pool di identità specificato.

Comando:

```
aws cognito-identity delete-identity-pool \
    --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, consulta DeleteIdentityPool 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.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.awscore.exception.AwsServiceException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cognitoidentity.CognitoIdentityClient;
import
 software.amazon.awssdk.services.cognitoidentity.model.DeleteIdentityPoolRequest;

/**
 * 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 DeleteIdentityPool {

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <identityPoolId>\s

                Where:
                    identityPoolId - The Id value of your identity pool.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String identityPoold = args[0];
        CognitoIdentityClient cognitoIdClient = CognitoIdentityClient.builder()
                .region(Region.US_EAST_1)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        deleteIdPool(cognitoIdClient, identityPoold);
        cognitoIdClient.close();
    }

    public static void deleteIdPool(CognitoIdentityClient cognitoIdClient, String
 identityPoold) {
        try {

            DeleteIdentityPoolRequest identityPoolRequest =
 DeleteIdentityPoolRequest.builder()
                    .identityPoolId(identityPoold)
                    .build();

            cognitoIdClient.deleteIdentityPool(identityPoolRequest);
            System.out.println("Done");

        } catch (AwsServiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la DeleteIdentityPoolsezione AWS SDK for Java 2.x API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: elimina un pool di identità specifico.

```
Remove-CGIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

- Per i dettagli sull'API, vedere DeleteIdentityPoolin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: elimina un pool di identità specifico.

```
Remove-CGIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

- Per i dettagli sull'API, vedere DeleteIdentityPoolin AWS Strumenti per PowerShell Cmdlet Reference (V5).

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 AWSCognitoIdentity


    /// Delete the specified identity pool.
    ///
    /// - Parameters:
    ///   - id: The ID of the identity pool to delete.
    ///
    func deleteIdentityPool(id: String) async throws {
        do {
            let input = DeleteIdentityPoolInput(
                identityPoolId: id
            )

            _ = try await cognitoIdentityClient.deleteIdentityPool(input: input)
        } catch {
            print("ERROR: deleteIdentityPool:", dump(error))
            throw error
        }
```

```
    }
```

- Per ulteriori informazioni, consulta Guida per gli sviluppatori di AWS SDK per Swift.
- Per i dettagli sull'API, consulta la DeleteIdentityPoolguida 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 **`DescribeIdentityPool`** con una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DescribeIdentityPool`.

CLI

AWS CLI

Come descrivere un pool di identità

Questo esempio descrive un pool di identità.

Comando:

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111"
```

Output:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
```

```
}
```

- Per i dettagli sull'API, consulta DescribeIdentityPool AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: recupera informazioni su uno specifico pool di identità in base al relativo ID.

```
Get-CGIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

Output:

```
LoggedAt                       : 8/12/2015 4:29:40 PM
AllowUnauthenticatedIdentities : True
DeveloperProviderName          :
IdentityPoolId                 : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
IdentityPoolName               : CommonTests1
OpenIdConnectProviderARNs      : {}
SupportedLoginProviders        : {}
ResponseMetadata               : Amazon.Runtime.ResponseMetadata
ContentLength                  : 142
HttpStatusCode                 : OK
```

- Per i dettagli sull'API, vedere DescribeIdentityPoolin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: recupera informazioni su uno specifico pool di identità in base al relativo ID.

```
Get-CGIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

Output:

```
LoggedAt                       : 8/12/2015 4:29:40 PM
AllowUnauthenticatedIdentities : True
DeveloperProviderName          :
IdentityPoolId                 : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
```

```
IdentityPoolName               : CommonTests1
OpenIdConnectProviderARNs      : {}
SupportedLoginProviders        : {}
ResponseMetadata               : Amazon.Runtime.ResponseMetadata
ContentLength                  : 142
HttpStatusCode                 : OK
```

- Per i dettagli sull'API, vedere DescribeIdentityPoolin 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 **`GetCredentialsForIdentity`** con un SDK AWS

Il seguente esempio di codice mostra come utilizzare`GetCredentialsForIdentity`.

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.cognitoidentity.CognitoIdentityClient;
import
 software.amazon.awssdk.services.cognitoidentity.model.GetCredentialsForIdentityRequest;
import
 software.amazon.awssdk.services.cognitoidentity.model.GetCredentialsForIdentityResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc

/**
 * 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 GetIdentityCredentials {
    public static void main(String[] args) {

        final String usage = """

                Usage:
                    <identityId>\s

            Where:
                identityId - The Id of an existing identity in the format
 REGION:GUID.
            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String identityId = args[0];
        CognitoIdentityClient cognitoClient = CognitoIdentityClient.builder()
                .region(Region.US_EAST_1)
                .build();

        getCredsForIdentity(cognitoClient, identityId);
        cognitoClient.close();
    }

    public static void getCredsForIdentity(CognitoIdentityClient cognitoClient,
 String identityId) {
        try {
            GetCredentialsForIdentityRequest getCredentialsForIdentityRequest =
 GetCredentialsForIdentityRequest
                    .builder()
                    .identityId(identityId)
                    .build();

            GetCredentialsForIdentityResponse response = cognitoClient
                    .getCredentialsForIdentity(getCredentialsForIdentityRequest);
            System.out.println(
```

```
                    "Identity ID " + response.identityId() + ", Access key ID " +
 response.credentials().accessKeyId());

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la GetCredentialsForIdentitysezione AWS SDK for Java 2.x 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.

Utilizzare **`GetIdentityPoolRoles`** con una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetIdentityPoolRoles`.

CLI

AWS CLI

Come ottenere i ruoli del pool di identità

Questo esempio elenca i ruoli del pool di identità.

Comando:

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111"
```

Output:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "Roles": {
      "authenticated": "arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolAuth_Role",
```

```
      "unauthenticated": "arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolUnauth_Role"
  }
}
```

- Per i dettagli sull'API, consulta GetIdentityPoolRoles AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: ottiene informazioni sui ruoli per un pool di identità specifico.

```
Get-CGIIdentityPoolRole -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

Output:

```
LoggedAt         : 8/12/2015 4:33:51 PM
IdentityPoolId   : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
Roles            : {[unauthenticated, arn:aws:iam::123456789012:role/
CommonTests1Role]}
ResponseMetadata : Amazon.Runtime.ResponseMetadata
ContentLength    : 165
HttpStatusCode   : OK
```

- Per i dettagli sull'API, vedere GetIdentityPoolRolesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: ottiene informazioni sui ruoli per un pool di identità specifico.

```
Get-CGIIdentityPoolRole -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

Output:

```
LoggedAt         : 8/12/2015 4:33:51 PM
IdentityPoolId   : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
Roles            : {[unauthenticated, arn:aws:iam::123456789012:role/
CommonTests1Role]}
```

```
ResponseMetadata : Amazon.Runtime.ResponseMetadata
ContentLength    : 165
HttpStatusCode   : OK
```

- Per i dettagli sull'API, vedere GetIdentityPoolRolesin 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 **`ListIdentityPools`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListIdentityPools`.

CLI

AWS CLI

Come elencare pool di identità

In questo esempio vengono elencati i pool di identità. Sono elencate un massimo di 20 identità.

Comando:

```
aws cognito-identity list-identity-pools --max-results 20
```

Output:

```
{
  "IdentityPools": [
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
          "IdentityPoolName": "MyIdentityPool"
      },
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
          "IdentityPoolName": "AnotherIdentityPool"
      },
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
```

```
            "IdentityPoolName": "IdentityPoolRegionA"
        }
    ]
}
```

- Per i dettagli sull'API, consulta ListIdentityPools 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.cognitoidentity.CognitoIdentityClient;
import
 software.amazon.awssdk.services.cognitoidentity.model.ListIdentityPoolsRequest;
import
 software.amazon.awssdk.services.cognitoidentity.model.ListIdentityPoolsResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc

/**
 * 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 ListIdentityPools {
    public static void main(String[] args) {
        CognitoIdentityClient cognitoClient = CognitoIdentityClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listIdPools(cognitoClient);
```

```
        cognitoClient.close();
    }

    public static void listIdPools(CognitoIdentityClient cognitoClient) {
        try {
            ListIdentityPoolsRequest poolsRequest =
 ListIdentityPoolsRequest.builder()
                    .maxResults(15)
                    .build();

            ListIdentityPoolsResponse response =
 cognitoClient.listIdentityPools(poolsRequest);
            response.identityPools().forEach(pool -> {
                System.out.println("Pool ID: " + pool.identityPoolId());
                System.out.println("Pool name: " + pool.identityPoolName());
            });

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListIdentityPoolssezione AWS SDK for Java 2.x API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: recupera un elenco di pool di identità esistenti.

```
Get-CGIIdentityPoolList
```

Output:

```
IdentityPoolId
 IdentityPoolName
--------------
 ----------------
```

```
us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1                  CommonTests1
us-east-1:118d242d-204e-4b88-b803-EXAMPLEGUID2                  Tests2
us-east-1:15d49393-ab16-431a-b26e-EXAMPLEGUID3                  CommonTests13
```

- Per i dettagli sull'API, vedere ListIdentityPoolsin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: recupera un elenco di pool di identità esistenti.

```
Get-CGIIdentityPoolList
```

Output:

```
IdentityPoolId
 IdentityPoolName
--------------
 ----------------
us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1                  CommonTests1
us-east-1:118d242d-204e-4b88-b803-EXAMPLEGUID2                  Tests2
us-east-1:15d49393-ab16-431a-b26e-EXAMPLEGUID3                  CommonTests13
```

- Per i dettagli sull'API, vedere ListIdentityPoolsin AWS Strumenti per PowerShell Cmdlet Reference (V5).

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 AWSCognitoIdentity


    /// Return the ID of the identity pool with the specified name.
    ///
```

```
    /// - Parameters:
    ///   - name: The name of the identity pool whose ID should be returned.
    ///
    /// - Returns: A string containing the ID of the specified identity pool
    ///   or `nil` on error or if not found.
    ///
    func getIdentityPoolID(name: String) async throws -> String? {
        let listPoolsInput = ListIdentityPoolsInput(maxResults: 25)
        // Use "Paginated" to get all the objects.
        // This lets the SDK handle the 'nextToken' field in
"ListIdentityPoolsOutput".
        let pages = cognitoIdentityClient.listIdentityPoolsPaginated(input:
listPoolsInput)

        do {
            for try await page in pages {
                guard let identityPools = page.identityPools else {
                    print("ERROR: listIdentityPoolsPaginated returned nil
contents.")
                    continue
                }

                /// Read pages of identity pools from Cognito until one is found
                /// whose name matches the one specified in the `name` parameter.
                /// Return the matching pool's ID.

                for pool in identityPools {
                    if pool.identityPoolName == name {
                        return pool.identityPoolId!
                    }
                }
            }
        } catch {
            print("ERROR: getIdentityPoolID:", dump(error))
            throw error
        }

        return nil
    }
```

Ottieni l’ID di un pool di identità esistente o crealo se non esiste.

```
import AWSCognitoIdentity


    /// Return the ID of the identity pool with the specified name.
    ///
    /// - Parameters:
    ///   - name: The name of the identity pool whose ID should be returned
    ///
    /// - Returns: A string containing the ID of the specified identity pool.
    ///   Returns `nil` if there's an error or if the pool isn't found.
    ///
    public func getOrCreateIdentityPoolID(name: String) async throws -> String? {
        // See if the pool already exists. If it doesn't, create it.

        do {
            guard let poolId = try await getIdentityPoolID(name: name) else {
                return try await createIdentityPool(name: name)
            }

            return poolId
        } catch {
            print("ERROR: getOrCreateIdentityPoolID:", dump(error))
            throw error
        }
    }
```

- Per ulteriori informazioni, consulta AWS SDK for Swift developer guide (Guida per gli sviluppatori di AWS SDK per Swift).
- Per i dettagli sull'API, consulta la ListIdentityPoolsguida 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 **`SetIdentityPoolRoles`** con una CLI

Gli esempi di codice seguenti mostrano come utilizzare `SetIdentityPoolRoles`.

CLI

AWS CLI

Come ottenere i ruoli del pool di identità

L'esempio `set-identity-pool-roles` seguente imposta il ruolo di un pool di identità.

```
aws cognito-identity set-identity-pool-roles \
    --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111" \
    --roles authenticated="arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolAuth_Role"
```

- Per i dettagli sull'API, consulta SetIdentityPoolRoles AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: configura un pool di identità specifico in modo che abbia un ruolo IAM non autenticato.

```
Set-CGIIdentityPoolRole -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1 -Role @{ "unauthenticated" = "arn:aws:iam::123456789012:role/
CommonTests1Role" }
```

- Per i dettagli sull'API, vedere SetIdentityPoolRolesin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: configura un pool di identità specifico in modo che abbia un ruolo IAM non autenticato.

```
Set-CGIIdentityPoolRole -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1 -Role @{ "unauthenticated" = "arn:aws:iam::123456789012:role/
CommonTests1Role" }
```

- Per i dettagli sull'API, vedere SetIdentityPoolRolesin 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 **`UpdateIdentityPool`** con una CLI

Gli esempi di codice seguenti mostrano come utilizzare `UpdateIdentityPool`.

CLI

AWS CLI

Come aggiornare un pool di identità

Questo esempio aggiorna un pool di identità. Imposta il nome su. MyIdentityPool Aggiunge Cognito come gestore dell'identità digitale. Non ammette identità non autenticate.

Comando:

```
aws cognito-identity update-identity-pool --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111" --identity-pool-
name "MyIdentityPool" --no-allow-unauthenticated-identities --cognito-
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Output:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- Per i dettagli sull'API, vedere UpdateIdentityPoolin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: aggiorna alcune proprietà del pool di identità (in questo caso il nome del pool di identità).

```
Update-CGIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1 -IdentityPoolName NewPoolName
```

Output:

```
LoggedAt                       : 8/12/2015 4:53:33 PM
AllowUnauthenticatedIdentities : False
DeveloperProviderName          :
IdentityPoolId                 : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
IdentityPoolName               : NewPoolName
OpenIdConnectProviderARNs      : {}
SupportedLoginProviders        : {}
ResponseMetadata               : Amazon.Runtime.ResponseMetadata
ContentLength                  : 135
HttpStatusCode                 : OK
```

- Per i dettagli sull'API, vedere UpdateIdentityPoolin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: aggiorna alcune proprietà del pool di identità (in questo caso il nome del pool di identità).

```
Update-CGIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1 -IdentityPoolName NewPoolName
```

Output:

```
LoggedAt                       : 8/12/2015 4:53:33 PM
AllowUnauthenticatedIdentities : False
DeveloperProviderName          :
IdentityPoolId                 : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
IdentityPoolName               : NewPoolName
OpenIdConnectProviderARNs      : {}
SupportedLoginProviders        : {}
```

```
ResponseMetadata                : Amazon.Runtime.ResponseMetadata
ContentLength                   : 135
HttpStatusCode                  : OK
```

- Per i dettagli sull'API, vedere UpdateIdentityPoolin 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.

Scenari per l'utilizzo di Amazon Cognito Identity AWS SDKs

I seguenti esempi di codice mostrano come implementare scenari comuni in Amazon Cognito Identity con. AWS SDKs Questi scenari illustrano come eseguire attività specifiche chiamando più funzioni all'interno dell'API di identità per Amazon Cognito o in combinazione 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

- Creazione di un'applicazione Amazon Textract explorer

Creazione di un'applicazione Amazon Textract explorer

Gli esempi di codice seguenti mostrano come esplorare l'output di Amazon Textract tramite un'applicazione interattiva.

JavaScript

SDK per JavaScript (v3)

Mostra come utilizzare per AWS SDK per JavaScript creare un'applicazione React che utilizza Amazon Textract per estrarre dati dall'immagine di un documento e visualizzarli in una pagina Web interattiva. Questo esempio viene eseguito in un browser Web e richiede, come credenziali, un'identità autenticata Amazon Cognito. Utilizza Amazon Simple Storage Service (Amazon S3) per l'archiviazione e per le notifiche esegue il polling di una coda Amazon

Simple Queue Service (Amazon SQS) sottoscritta a un argomento Amazon Simple Notification Service (Amazon SNS).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon Cognito Identity
- Simple Storage Service (Amazon S3)
- Amazon SNS
- Amazon SQS
- Amazon Textract

Python

SDK per Python (Boto3)

Mostra come utilizzarlo AWS SDK per Python (Boto3) con Amazon Textract per rilevare elementi di testo, moduli e tabelle nell'immagine di un documento. L'immagine di input e l'output di Amazon Textract sono mostrati in un'applicazione Tkinter che consente di esplorare gli elementi rilevati.

- Invia un'immagine del documento ad Amazon Textract ed esplora l'output degli elementi rilevati.
- Invia immagini direttamente ad Amazon Textract o tramite un bucket Amazon Simple Storage Service (Amazon S3).
- Utilizza asynchronous APIs per avviare un processo che pubblica una notifica su un argomento di Amazon Simple Notification Service (Amazon SNS) al termine del processo.
- Esegue il polling di una coda Amazon Simple Queue Service (Amazon SQS) per un messaggio di completamento del processo e visualizza i risultati.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Amazon Cognito Identity
- Simple Storage Service (Amazon S3)

- Amazon SNS
- Amazon SQS
- Amazon Textract

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 l'utilizzo di Amazon Cognito Identity Provider AWS SDK

I seguenti esempi di codice mostrano come utilizzare Amazon Cognito Identity Provider 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 di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestore dell'identità per Amazon Cognito

- Esempi di base per l'utilizzo di Amazon Cognito Identity Provider AWS SDK
  - Ciao Amazon Cognito
  - Azioni per l'utilizzo di Amazon Cognito Identity Provider AWS SDK
    - Da utilizzare AdminCreateUsercon un AWS SDK o CLI
    - Utilizzare AdminGetUsercon un AWS SDK o CLI
    - Utilizzare AdminInitiateAuthcon un AWS SDK o CLI
    - Utilizzare AdminRespondToAuthChallengecon un AWS SDK o CLI
    - Utilizzare AdminSetUserPasswordcon un AWS SDK o CLI
    - Utilizzare AssociateSoftwareTokencon un AWS SDK o CLI

- Utilizzare ConfirmDevicecon un AWS SDK o CLI
- Utilizzare ConfirmForgotPasswordcon un AWS SDK o CLI
- Utilizzare ConfirmSignUpcon un AWS SDK o CLI
- Utilizzare CreateUserPoolcon un AWS SDK o CLI
- Utilizzare CreateUserPoolClientcon un AWS SDK o CLI
- Utilizzare DeleteUsercon un AWS SDK o CLI
- Utilizzare DescribeTermsByClientcon un AWS SDK
- Utilizzare DescribeUserPoolClientcon un AWS SDK o CLI
- Utilizzare ForgotPasswordcon un AWS SDK o CLI
- Utilizzare GetClientTokencon un AWS SDK
- Utilizzare InitiateAuthcon un AWS SDK o CLI
- Utilizzare ListUserPoolscon un AWS SDK o CLI
- Utilizzare ListUserscon un AWS SDK o CLI
- Utilizzare ResendConfirmationCodecon un AWS SDK o CLI
- Utilizzare RespondToAuthChallengecon un AWS SDK o CLI
- Utilizzare SignUpcon un AWS SDK o CLI
- Utilizzare UpdateUserPoolcon un AWS SDK o CLI
- Utilizzare VerifySoftwareTokencon un AWS SDK o CLI

- Scenari per l'utilizzo di Amazon Cognito Identity Provider AWS SDK
  - Conferma automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda utilizzando un AWS SDK
  - Esegui automaticamente la migrazione di utenti Amazon Cognito noti con una funzione Lambda utilizzando un AWS SDK
  - Ottieni token client M2M e scopri i termini di Amazon Cognito utilizzando un AWS SDK
  - Nozioni di base sui bacini d'utenza
  - Registra un utente con un pool di utenti Amazon Cognito che richiede l'autenticazione a più fattori utilizzando un AWS SDK
  - Usa i pool di identità e i flussi di autenticazione di Amazon Cognito
  - Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito utilizzando un AWS SDK

Esempi di base per l'utilizzo di Amazon Cognito Identity Provider AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare le nozioni di base del gestore dell'identità per Amazon Cognito con gli SDK AWS .

Esempi

- Ciao Amazon Cognito
- Azioni per l'utilizzo di Amazon Cognito Identity Provider AWS SDK
  - Da utilizzare AdminCreateUsercon un AWS SDK o CLI
  - Utilizzare AdminGetUsercon un AWS SDK o CLI
  - Utilizzare AdminInitiateAuthcon un AWS SDK o CLI
  - Utilizzare AdminRespondToAuthChallengecon un AWS SDK o CLI
  - Utilizzare AdminSetUserPasswordcon un AWS SDK o CLI
  - Utilizzare AssociateSoftwareTokencon un AWS SDK o CLI
  - Utilizzare ConfirmDevicecon un AWS SDK o CLI
  - Utilizzare ConfirmForgotPasswordcon un AWS SDK o CLI
  - Utilizzare ConfirmSignUpcon un AWS SDK o CLI
  - Utilizzare CreateUserPoolcon un AWS SDK o CLI
  - Utilizzare CreateUserPoolClientcon un AWS SDK o CLI
  - Utilizzare DeleteUsercon un AWS SDK o CLI
  - Utilizzare DescribeTermsByClientcon un AWS SDK
  - Utilizzare DescribeUserPoolClientcon un AWS SDK o CLI
  - Utilizzare ForgotPasswordcon un AWS SDK o CLI
  - Utilizzare GetClientTokencon un AWS SDK
  - Utilizzare InitiateAuthcon un AWS SDK o CLI
  - Utilizzare ListUserPoolscon un AWS SDK o CLI
  - Utilizzare ListUserscon un AWS SDK o CLI
  - Utilizzare ResendConfirmationCodecon un AWS SDK o CLI
  - Utilizzare RespondToAuthChallengecon un AWS SDK o CLI
  - Utilizzare SignUpcon un AWS SDK o CLI
  - Utilizzare UpdateUserPoolcon un AWS SDK o CLI
  - Utilizzare VerifySoftwareTokencon un AWS SDK o CLI

Ciao Amazon Cognito

Gli esempi di codice seguenti mostrano come iniziare a utilizzare Amazon Cognito.

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 cognito-idp)

# Set this project's name.
project("hello_cognito")

# 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_cognito.cpp)

target_link_libraries(${PROJECT_NAME}
        ${AWSSDK_LINK_LIBRARIES})
```

Codice per il file origine hello_cognito.cpp.

```
#include <aws/core/Aws.h>
#include <aws/cognito-idp/CognitoIdentityProviderClient.h>
#include <aws/cognito-idp/model/ListUserPoolsRequest.h>
#include <iostream>

/*
 *  A "Hello Cognito" starter application which initializes an Amazon Cognito
 client and lists the Amazon Cognito
 *  user pools.
 *
 *  main function
 *
 *  Usage: 'hello_cognito'
 *
 */

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;
```

```
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
cognitoClient(clientConfig);

        Aws::String nextToken; // Used for pagination.
        std::vector<Aws::String> userPools;

        do {
            Aws::CognitoIdentityProvider::Model::ListUserPoolsRequest
listUserPoolsRequest;
            if (!nextToken.empty()) {
                listUserPoolsRequest.SetNextToken(nextToken);
            }

            Aws::CognitoIdentityProvider::Model::ListUserPoolsOutcome
listUserPoolsOutcome =
                    cognitoClient.ListUserPools(listUserPoolsRequest);

            if (listUserPoolsOutcome.IsSuccess()) {
                for (auto &userPool:
listUserPoolsOutcome.GetResult().GetUserPools()) {

                    userPools.push_back(userPool.GetName());
                }

                nextToken = listUserPoolsOutcome.GetResult().GetNextToken();
            } else {
                std::cerr << "ListUserPools error: " <<
listUserPoolsOutcome.GetError().GetMessage() << std::endl;
                result = 1;
                break;
            }


        } while (!nextToken.empty());
        std::cout << userPools.size() << " user pools found." << std::endl;
        for (auto &userPool: userPools) {
            std::cout << "   user pool: " << userPool << std::endl;
        }
    }
```

```
    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Per i dettagli sull'API, vedi ListUserPoolsin 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"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Notification
 Service
// (Amazon SNS) client and list the topics 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
}
cognitoClient := cognitoidentityprovider.NewFromConfig(sdkConfig)
fmt.Println("Let's list the user pools for your account.")
var pools []types.UserPoolDescriptionType
paginator := cognitoidentityprovider.NewListUserPoolsPaginator(
 cognitoClient, &cognitoidentityprovider.ListUserPoolsInput{MaxResults:
aws.Int32(10)})
for paginator.HasMorePages() {
 output, err := paginator.NextPage(ctx)
 if err != nil {
  log.Printf("Couldn't get user pools. Here's why: %v\n", err)
 } else {
  pools = append(pools, output.UserPools...)
 }
}
if len(pools) == 0 {
 fmt.Println("You don't have any user pools!")
} else {
 for _, pool := range pools {
  fmt.Printf("\t%v: %v\n", *pool.Name, *pool.Id)
 }
}
}
```

- Per i dettagli sull'API, ListUserPoolsconsulta 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.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsRequest;

/**
 * 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 ListUserPools {
    public static void main(String[] args) {
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllUserPools(cognitoClient);
        cognitoClient.close();
    }

    public static void listAllUserPools(CognitoIdentityProviderClient
 cognitoClient) {
        try {
            ListUserPoolsRequest request = ListUserPoolsRequest.builder()
                    .maxResults(10)
                    .build();

            ListUserPoolsResponse response =
 cognitoClient.listUserPools(request);
            response.userPools().forEach(userpool -> {
                System.out.println("User pool " + userpool.name() + ", User ID "
 + userpool.id());
            });
```

```
        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, ListUserPoolsconsulta 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 {
  paginateListUserPools,
  CognitoIdentityProviderClient,
} from "@aws-sdk/client-cognito-identity-provider";

const client = new CognitoIdentityProviderClient({});

export const helloCognito = async () => {
  const paginator = paginateListUserPools({ client }, {});

  const userPoolNames = [];

  for await (const page of paginator) {
    const names = page.UserPools.map((pool) => pool.Name);
    userPoolNames.push(...names);
  }

  console.log("User pool names: ");
  console.log(userPoolNames.join("\n"));
  return userPoolNames;
};
```

- Per i dettagli sull'API, ListUserPoolsconsulta 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

# Create a Cognito Identity Provider client
cognitoidp = boto3.client("cognito-idp")

# Initialize a paginator for the list_user_pools operation
paginator = cognitoidp.get_paginator("list_user_pools")

# Create a PageIterator from the paginator
page_iterator = paginator.paginate(MaxResults=10)

# Initialize variables for pagination
user_pools = []

# Handle pagination
for page in page_iterator:
    user_pools.extend(page.get("UserPools", []))

# Print the list of user pools
print("User Pools for the account:")
if user_pools:
    for pool in user_pools:
        print(f"Name: {pool['Name']}, ID: {pool['Id']}")
else:
    print("No user pools found.")
```

- Per i dettagli sull'API, consulta ListUserPools 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-cognitoidentityprovider'
require 'logger'

# CognitoManager is a class responsible for managing AWS Cognito operations
# such as listing all user pools in the current AWS account.
class CognitoManager
  def initialize(client)
    @client = client
    @logger = Logger.new($stdout)
  end

  # Lists and prints all user pools associated with the AWS account.
  def list_user_pools
    paginator = @client.list_user_pools(max_results: 10)
    user_pools = []
    paginator.each_page do |page|
      user_pools.concat(page.user_pools)
    end

    if user_pools.empty?
      @logger.info('No Cognito user pools found.')
    else
      user_pools.each do |user_pool|
        @logger.info("User pool ID: #{user_pool.id}")
        @logger.info("User pool name: #{user_pool.name}")
        @logger.info("User pool status: #{user_pool.status}")
        @logger.info('---')
```

```
      end
    end
  end
end

if $PROGRAM_NAME == __FILE__
  cognito_client = Aws::CognitoIdentityProvider::Client.new
  manager = CognitoManager.new(cognito_client)
  manager.list_user_pools
end
```

- Per i dettagli sull'API, ListUserPoolsconsulta 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.

Azioni per l'utilizzo di Amazon Cognito Identity Provider AWS SDK

I seguenti esempi di codice mostrano come eseguire singole azioni di Amazon Cognito Identity Provider con AWS gli 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 Amazon Cognito Identity Provider 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 Amazon Cognito Identity Provider AWS SDK.

Gli esempi seguenti includono solo le azioni più comunemente utilizzate. Per l'elenco completo, consulta la documentazione di riferimento dell'API del provider di identità di Amazon Cognito.

Esempi

- Da utilizzare AdminCreateUsercon un AWS SDK o CLI
- Utilizzare AdminGetUsercon un AWS SDK o CLI
- Utilizzare AdminInitiateAuthcon un AWS SDK o CLI
- Utilizzare AdminRespondToAuthChallengecon un AWS SDK o CLI
- Utilizzare AdminSetUserPasswordcon un AWS SDK o CLI

- Utilizzare AssociateSoftwareTokencon un AWS SDK o CLI
- Utilizzare ConfirmDevicecon un AWS SDK o CLI
- Utilizzare ConfirmForgotPasswordcon un AWS SDK o CLI
- Utilizzare ConfirmSignUpcon un AWS SDK o CLI
- Utilizzare CreateUserPoolcon un AWS SDK o CLI
- Utilizzare CreateUserPoolClientcon un AWS SDK o CLI
- Utilizzare DeleteUsercon un AWS SDK o CLI
- Utilizzare DescribeTermsByClientcon un AWS SDK
- Utilizzare DescribeUserPoolClientcon un AWS SDK o CLI
- Utilizzare ForgotPasswordcon un AWS SDK o CLI
- Utilizzare GetClientTokencon un AWS SDK
- Utilizzare InitiateAuthcon un AWS SDK o CLI
- Utilizzare ListUserPoolscon un AWS SDK o CLI
- Utilizzare ListUserscon un AWS SDK o CLI
- Utilizzare ResendConfirmationCodecon un AWS SDK o CLI
- Utilizzare RespondToAuthChallengecon un AWS SDK o CLI
- Utilizzare SignUpcon un AWS SDK o CLI
- Utilizzare UpdateUserPoolcon un AWS SDK o CLI
- Utilizzare VerifySoftwareTokencon un AWS SDK o CLI

Da utilizzare **`AdminCreateUser`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`AdminCreateUser`.

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 bacini d'utenza
- Scrivere i dati di attività personalizzate con una funzione Lambda dopo l’autenticazione utente di Amazon Cognito tramite un SDK

CLI

AWS CLI

Come creare un utente

L'esempio `admin-create-user` seguente crea un utente con le impostazioni specificate per l'indirizzo e-mail e il numero di telefono.

```
aws cognito-idp admin-create-user \
    --user-pool-id us-west-2_aaaaaaaaa \
    --username diego \
    --user-attributes Name=email,Value=diego@example.com
 Name=phone_number,Value="+15555551212" \
    --message-action SUPPRESS
```

Output:

```
{
    "User": {
        "Username": "diego",
        "Attributes": [
            {
                "Name": "sub",
                "Value": "7325c1de-b05b-4f84-b321-9adc6e61f4a2"
            },
            {
                "Name": "phone_number",
                "Value": "+15555551212"
            },
            {
                "Name": "email",
                "Value": "diego@example.com"
            }
        ],
        "UserCreateDate": 1548099495.428,
        "UserLastModifiedDate": 1548099495.428,
        "Enabled": true,
        "UserStatus": "FORCE_CHANGE_PASSWORD"
    }
}
```

- Per i dettagli sull'API, consulta AdminCreateUser 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"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// AdminCreateUser uses administrator credentials to add a user to a user pool.
 This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(ctx context.Context, userPoolId
 string, userName string, userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(ctx,
 &cognitoidentityprovider.AdminCreateUserInput{
  UserPoolId:     aws.String(userPoolId),
  Username:       aws.String(userName),
  MessageAction:  types.MessageActionTypeSuppress,
  UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
 aws.String(userEmail)}},
 })
 if err != nil {
  var userExists *types.UsernameExistsException
```

```
  if errors.As(err, &userExists) {
   log.Printf("User %v already exists in the user pool.", userName)
   err = nil
  } else {
   log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
  }
 }
 return err
}
```

- Per i dettagli sull'API, AdminCreateUserconsulta AWS SDK per Go 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.

Utilizzare **`AdminGetUser`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`AdminGetUser`.

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:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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 the specified user from an Amazon Cognito user pool with
 administrator access.
```

```
    /// </summary>
    /// <param name="userName">The name of the user.</param>
    /// <param name="poolId">The Id of the Amazon Cognito user pool.</param>
    /// <returns>Async task.</returns>
    public async Task<UserStatusType> GetAdminUserAsync(string userName, string
poolId)
    {
        AdminGetUserRequest userRequest = new AdminGetUserRequest
        {
            Username = userName,
            UserPoolId = poolId,
        };

        var response = await _cognitoService.AdminGetUserAsync(userRequest);

        Console.WriteLine($"User status {response.UserStatus}");
        return response.UserStatus;
    }
```

- Per i dettagli sull'API, AdminGetUserconsulta 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.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

    Aws::CognitoIdentityProvider::Model::AdminGetUserRequest request;
    request.SetUsername(userName);
```

```
    request.SetUserPoolId(userPoolID);

    Aws::CognitoIdentityProvider::Model::AdminGetUserOutcome outcome =
            client.AdminGetUser(request);

    if (outcome.IsSuccess()) {
        std::cout << "The status for " << userName << " is " <<

 Aws::CognitoIdentityProvider::Model::UserStatusTypeMapper::GetNameForUserStatusType(
                          outcome.GetResult().GetUserStatus()) << std::endl;
        std::cout << "Enabled is " << outcome.GetResult().GetEnabled() <<
 std::endl;
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::AdminGetUser. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
```

- Per i dettagli sull'API, AdminGetUserconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come ottenere un utente

Questo esempio mostra come ottenere informazioni sul nome utente jane@example.com.

Comando:

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

Output:

```
{
  "Username": "4320de44-2322-4620-999b-5e2e1c8df013",
  "Enabled": true,
  "UserStatus": "FORCE_CHANGE_PASSWORD",
  "UserCreateDate": 1548108509.537,
```

```
  "UserAttributes": [
      {
          "Name": "sub",
          "Value": "4320de44-2322-4620-999b-5e2e1c8df013"
      },
      {
          "Name": "email_verified",
          "Value": "true"
      },
      {
          "Name": "phone_number_verified",
          "Value": "true"
      },
      {
          "Name": "phone_number",
          "Value": "+01115551212"
      },
      {
          "Name": "email",
          "Value": "jane@example.com"
      }
  ],
  "UserLastModifiedDate": 1548108509.537
}
```

- Per i dettagli sull'API, consulta AdminGetUser 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.

```
   public static void getAdminUser(CognitoIdentityProviderClient
identityProviderClient, String userName,
           String poolId) {
       try {
           AdminGetUserRequest userRequest = AdminGetUserRequest.builder()
```

```
                    .username(userName)
                    .userPoolId(poolId)
                    .build();

            AdminGetUserResponse response =
 identityProviderClient.adminGetUser(userRequest);
            System.out.println("User status " + response.userStatusAsString());

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, AdminGetUserconsulta 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.

```
const adminGetUser = ({ userPoolId, username }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new AdminGetUserCommand({
    UserPoolId: userPoolId,
    Username: username,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, AdminGetUserconsulta 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 getAdminUser(
    userNameVal: String?,
    poolIdVal: String?,
) {
    val userRequest =
        AdminGetUserRequest {
            username = userNameVal
            userPoolId = poolIdVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.adminGetUser(userRequest)
        println("User status ${response.userStatus}")
    }
}
```

- Per i dettagli sull'API, AdminGetUserconsulta AWS SDK for Kotlin 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.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def sign_up_user(self, user_name, password, user_email):
        """
        Signs up a new user with Amazon Cognito. This action prompts Amazon
 Cognito
        to send an email to the specified email address. The email contains a
 code that
        can be used to confirm the user.

        When the user already exists, the user status is checked to determine
 whether
        the user has been confirmed.

        :param user_name: The user name that identifies the new user.
        :param password: The password for the new user.
        :param user_email: The email address for the new user.
        :return: True when the user is already confirmed with Amazon Cognito.
                 Otherwise, false.
        """
        try:
            kwargs = {
                "ClientId": self.client_id,
                "Username": user_name,
                "Password": password,
                "UserAttributes": [{"Name": "email", "Value": user_email}],
```

```
            }
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            response = self.cognito_idp_client.sign_up(**kwargs)
            confirmed = response["UserConfirmed"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "UsernameExistsException":
                response = self.cognito_idp_client.admin_get_user(
                    UserPoolId=self.user_pool_id, Username=user_name
                )
                logger.warning(
                    "User %s exists and is %s.", user_name,
response["UserStatus"]
                )
                confirmed = response["UserStatus"] == "CONFIRMED"
            else:
                logger.error(
                    "Couldn't sign up %s. Here's why: %s: %s",
                    user_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        return confirmed
```

- Per i dettagli sull'API, consulta AdminGetUser AWSSDK for Python (Boto3) 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 AWSClientRuntime
import AWSCognitoIdentityProvider
```

```
    /// Get information about a specific user in a user pool.
    ///
    /// - Parameters:
    ///   - cipClient: The Amazon Cognito Identity Provider client to use.
    ///   - userName: The user to retrieve information about.
    ///   - userPoolId: The user pool to search for the specified user.
    ///
    /// - Returns: `true` if the user's information was successfully
    ///   retrieved. Otherwise returns `false`.
    func adminGetUser(cipClient: CognitoIdentityProviderClient, userName: String,
                      userPoolId: String) async -> Bool {
        do {
            let output = try await cipClient.adminGetUser(
                input: AdminGetUserInput(
                    userPoolId: userPoolId,
                    username: userName
                )
            )

            guard let userStatus = output.userStatus else {
                print("*** Unable to get the user's status.")
                return false
            }

            print("User status: \(userStatus)")
            return true
        } catch {
            return false
        }
    }
```

- Per i dettagli sull'API, consulta la AdminGetUserguida 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 **`AdminInitiateAuth`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`AdminInitiateAuth`.

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:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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>
    /// Initiate an admin auth request.
    /// </summary>
    /// <param name="clientId">The client ID to use.</param>
    /// <param name="userPoolId">The ID of the user pool.</param>
    /// <param name="userName">The username to authenticate.</param>
    /// <param name="password">The user's password.</param>
    /// <returns>The session to use in challenge-response.</returns>
    public async Task<string> AdminInitiateAuthAsync(string clientId, string
userPoolId, string userName, string password)
    {
        var authParameters = new Dictionary<string, string>();
        authParameters.Add("USERNAME", userName);
        authParameters.Add("PASSWORD", password);

        var request = new AdminInitiateAuthRequest
        {
            ClientId = clientId,
            UserPoolId = userPoolId,
            AuthParameters = authParameters,
            AuthFlow = AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
        };

        var response = await _cognitoService.AdminInitiateAuthAsync(request);
        return response.Session;
    }
```

- Per i dettagli sull'API, AdminInitiateAuthconsulta 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.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

    Aws::CognitoIdentityProvider::Model::AdminInitiateAuthRequest request;
    request.SetClientId(clientID);
    request.SetUserPoolId(userPoolID);
    request.AddAuthParameters("USERNAME", userName);
    request.AddAuthParameters("PASSWORD", password);
    request.SetAuthFlow(

Aws::CognitoIdentityProvider::Model::AuthFlowType::ADMIN_USER_PASSWORD_AUTH);


    Aws::CognitoIdentityProvider::Model::AdminInitiateAuthOutcome outcome =
            client.AdminInitiateAuth(request);

    if (outcome.IsSuccess()) {
        std::cout << "Call to AdminInitiateAuth was successful." << std::endl;
        sessionResult = outcome.GetResult().GetSession();
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::AdminInitiateAuth. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
```

```
    }
```

- Per i dettagli sull'API, AdminInitiateAuthconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come accedere come utente admin

L'esempio `admin-initiate-auth` seguente accede all'utente diego@example.com. Questo esempio include anche metadati per la protezione dalle minacce e ClientMetadata per i trigger Lambda. L'utente è configurato per l'autenticazione a più fattori (MFA) basata su password monouso e riceve una richiesta di verifica che richiede l'immissione di un codice nell'app di autenticazione in uso prima di poter completare l'autenticazione.

```
aws cognito-idp admin-initiate-auth \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --auth-flow ADMIN_USER_PASSWORD_AUTH \
    --auth-parameters USERNAME=diego@example.com,PASSWORD="My@Example
$Password3!",SECRET_HASH=ExampleEncodedClientIdSecretAndUsername= \
    --context-data="{\"EncodedData\":\"abc123example\",\"HttpHeaders\":
[{\"headerName\":\"UserAgent\",\"headerValue\":\"Mozilla/5.0 (Windows NT
 6.1; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0\"}],\"IpAddress\":
\"192.0.2.1\",\"ServerName\":\"example.com\",\"ServerPath\":\"/login\"}" \
    --client-metadata="{\"MyExampleKey\": \"MyExampleValue\"}"
```

Output:

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeExample...",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "MyAuthenticatorApp",
        "USER_ID_FOR_SRP": "diego@example.com"
    }
}
```

Per ulteriori informazioni, consulta Flusso di autenticazione amministratore nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta Command AdminInitiateAuthReference 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.

```
   public static AdminInitiateAuthResponse
initiateAuth(CognitoIdentityProviderClient identityProviderClient,
           String clientId, String userName, String password, String userPoolId)
{
       try {
           Map<String, String> authParameters = new HashMap<>();
           authParameters.put("USERNAME", userName);
           authParameters.put("PASSWORD", password);

           AdminInitiateAuthRequest authRequest =
AdminInitiateAuthRequest.builder()
                   .clientId(clientId)
                   .userPoolId(userPoolId)
                   .authParameters(authParameters)
                   .authFlow(AuthFlowType.ADMIN_USER_PASSWORD_AUTH)
                   .build();

           AdminInitiateAuthResponse response =
identityProviderClient.adminInitiateAuth(authRequest);
           System.out.println("Result Challenge is : " +
response.challengeName());
           return response;

       } catch (CognitoIdentityProviderException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
```

```
        return null;
    }
```

- Per i dettagli sull'API, AdminInitiateAuthconsulta 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.

```
const adminInitiateAuth = ({ clientId, userPoolId, username, password }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new AdminInitiateAuthCommand({
    ClientId: clientId,
    UserPoolId: userPoolId,
    AuthFlow: AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
    AuthParameters: { USERNAME: username, PASSWORD: password },
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, AdminInitiateAuthconsulta 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 checkAuthMethod(
    clientIdVal: String,
    userNameVal: String,
    passwordVal: String,
    userPoolIdVal: String,
): AdminInitiateAuthResponse {
    val authParas = mutableMapOf<String, String>()
    authParas["USERNAME"] = userNameVal
    authParas["PASSWORD"] = passwordVal

    val authRequest =
        AdminInitiateAuthRequest {
            clientId = clientIdVal
            userPoolId = userPoolIdVal
            authParameters = authParas
            authFlow = AuthFlowType.AdminUserPasswordAuth
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.adminInitiateAuth(authRequest)
        println("Result Challenge is ${response.challengeName}")
        return response
    }
}
```

- Per i dettagli sull'API, AdminInitiateAuthconsulta AWS SDK for Kotlin 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.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def start_sign_in(self, user_name, password):
        """
        Starts the sign-in process for a user by using administrator credentials.
        This method of signing in is appropriate for code running on a secure
 server.

        If the user pool is configured to require MFA and this is the first sign-
in
        for the user, Amazon Cognito returns a challenge response to set up an
        MFA application. When this occurs, this function gets an MFA secret from
        Amazon Cognito and returns it to the caller.

        :param user_name: The name of the user to sign in.
```

```
        :param password: The user's password.
        :return: The result of the sign-in attempt. When sign-in is successful,
this
                 returns an access token that can be used to get AWS credentials.
Otherwise,
                 Amazon Cognito returns a challenge to set up an MFA application,
                 or a challenge to enter an MFA code from a registered MFA
application.
        """
        try:
            kwargs = {
                "UserPoolId": self.user_pool_id,
                "ClientId": self.client_id,
                "AuthFlow": "ADMIN_USER_PASSWORD_AUTH",
                "AuthParameters": {"USERNAME": user_name, "PASSWORD": password},
            }
            if self.client_secret is not None:
                kwargs["AuthParameters"]["SECRET_HASH"] =
self._secret_hash(user_name)
            response = self.cognito_idp_client.admin_initiate_auth(**kwargs)
            challenge_name = response.get("ChallengeName", None)
            if challenge_name == "MFA_SETUP":
                if (
                    "SOFTWARE_TOKEN_MFA"
                    in response["ChallengeParameters"]["MFAS_CAN_SETUP"]
                ):
                    response.update(self.get_mfa_secret(response["Session"]))
                else:
                    raise RuntimeError(
                        "The user pool requires MFA setup, but the user pool is
not "
                        "configured for TOTP MFA. This example requires TOTP
MFA."
                    )
        except ClientError as err:
            logger.error(
                "Couldn't start sign in for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
```

```
            return response
```

- Per i dettagli sull'API, consulta AdminInitiateAuth 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.
        " Set up authentication parameters
        DATA(lt_auth_params) = VALUE /aws1/
cl_cgpauthparamstype_w=>tt_authparameterstype(
          ( VALUE /aws1/cl_cgpauthparamstype_w=>ts_authparameterstype_maprow(
              key = 'USERNAME'
              value = NEW /aws1/cl_cgpauthparamstype_w( iv_user_name ) ) )
          ( VALUE /aws1/cl_cgpauthparamstype_w=>ts_authparameterstype_maprow(
              key = 'PASSWORD'
              value = NEW /aws1/cl_cgpauthparamstype_w( iv_password ) ) )
        ).

        " Add SECRET_HASH if provided
        IF iv_secret_hash IS NOT INITIAL.
          INSERT VALUE #(
            key = 'SECRET_HASH'
            value = NEW /aws1/cl_cgpauthparamstype_w( iv_secret_hash )
          ) INTO TABLE lt_auth_params.
        ENDIF.

        oo_result = lo_cgp->admininitiateauth(
          iv_userpoolid = iv_user_pool_id
          iv_clientid = iv_client_id
          iv_authflow = 'ADMIN_USER_PASSWORD_AUTH'
```

```
          it_authparameters = lt_auth_params
        ).

        DATA(lv_challenge) = oo_result->get_challengename( ).

        IF lv_challenge IS INITIAL.
          MESSAGE 'User successfully signed in.' TYPE 'I'.
        ELSE.
          MESSAGE |Authentication challenge required: { lv_challenge }.| TYPE
 'I'.
        ENDIF.

      CATCH /aws1/cx_cgpusernotfoundex INTO DATA(lo_user_ex).
        MESSAGE |User { iv_user_name } not found.| TYPE 'E'.

      CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
        MESSAGE 'Not authorized. Check credentials.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, AdminInitiateAuthconsulta 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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Begin an authentication session.
    ///
    /// - Parameters:
    ///   - cipClient: The `CongitoIdentityProviderClient` to use.
    ///   - clientId: The app client ID to use.
    ///   - userName: The username to check.
    ///   - password: The user's password.
```

```
    ///   - userPoolId: The user pool to use.
    ///
    /// - Returns: The session token associated with this authentication
    ///   session.
    func initiateAuth(cipClient: CognitoIdentityProviderClient, clientId: String,
                      userName: String, password: String,
                      userPoolId: String) async -> String? {
        var authParams: [String: String] = [:]

        authParams["USERNAME"] = userName
        authParams["PASSWORD"] = password

        do {
            let output = try await cipClient.adminInitiateAuth(
                input: AdminInitiateAuthInput(
                    authFlow:
CognitoIdentityProviderClientTypes.AuthFlowType.adminUserPasswordAuth,
                    authParameters: authParams,
                    clientId: clientId,
                    userPoolId: userPoolId
                )
            )

            guard let challengeName = output.challengeName else {
                print("*** Invalid response from the auth service.")
                return nil
            }

            print("=====> Response challenge is \(challengeName)")

            return output.session
        } catch _ as UserNotFoundException {
            print("*** The specified username, \(userName), doesn't exist.")
            return nil
        } catch _ as UserNotConfirmedException {
            print("*** The user \(userName) has not been confirmed.")
            return nil
        } catch {
            print("*** An unexpected error occurred.")
            return nil
        }
    }
```

- Per i dettagli sull'API, consulta la AdminInitiateAuthguida 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 **`AdminRespondToAuthChallenge`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`AdminRespondToAuthChallenge`.

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:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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>
/// Respond to an admin authentication challenge.
/// </summary>
/// <param name="userName">The name of the user.</param>
/// <param name="clientId">The client ID.</param>
/// <param name="mfaCode">The multi-factor authentication code.</param>
/// <param name="session">The current application session.</param>
/// <param name="clientId">The user pool ID.</param>
/// <returns>The result of the authentication response.</returns>
public async Task<AuthenticationResultType> AdminRespondToAuthChallengeAsync(
    string userName,
    string clientId,
    string mfaCode,
    string session,
```

```
        string userPoolId)
    {
        Console.WriteLine("SOFTWARE_TOKEN_MFA challenge is generated");

        var challengeResponses = new Dictionary<string, string>();
        challengeResponses.Add("USERNAME", userName);
        challengeResponses.Add("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

        var respondToAuthChallengeRequest = new
AdminRespondToAuthChallengeRequest
        {
            ChallengeName = ChallengeNameType.SOFTWARE_TOKEN_MFA,
            ClientId = clientId,
            ChallengeResponses = challengeResponses,
            Session = session,
            UserPoolId = userPoolId,
        };

        var response = await
_cognitoService.AdminRespondToAuthChallengeAsync(respondToAuthChallengeRequest);
        Console.WriteLine($"Response to Authentication
{response.AuthenticationResult.TokenType}");
        return response.AuthenticationResult;
    }
```

- Per i dettagli sull'API, AdminRespondToAuthChallengeconsulta 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.

```
        Aws::Client::ClientConfiguration clientConfig;
```

```
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

        Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeRequest
request;
        request.AddChallengeResponses("USERNAME", userName);
        request.AddChallengeResponses("SOFTWARE_TOKEN_MFA_CODE", mfaCode);
        request.SetChallengeName(

Aws::CognitoIdentityProvider::Model::ChallengeNameType::SOFTWARE_TOKEN_MFA);
        request.SetClientId(clientID);
        request.SetUserPoolId(userPoolID);
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeOutcome
outcome =
                client.AdminRespondToAuthChallenge(request);

        if (outcome.IsSuccess()) {
            std::cout << "Here is the response to the challenge.\n" <<

outcome.GetResult().GetAuthenticationResult().Jsonize().View().WriteReadable()
                      << std::endl;

            accessToken =
outcome.GetResult().GetAuthenticationResult().GetAccessToken();
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::AdminRespondToAuthChallenge. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Per i dettagli sull'API, AdminRespondToAuthChallengeconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come rispondere a una richiesta di autenticazione

Esistono molti modi per rispondere alle diverse richieste di autenticazione, a seconda del flusso di autenticazione, della configurazione del pool di utenti e delle impostazioni utente. L'esempio `admin-respond-to-auth-challenge` seguente fornisce un codice MFA TOTP per diego@example.com e completa l'accesso. In questo pool di utenti è attivata la memorizzazione dei dispositivi, quindi il risultato dell'autenticazione restituisce anche una nuova chiave del dispositivo.

```
aws cognito-idp admin-respond-to-auth-challenge \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --challenge-name SOFTWARE_TOKEN_MFA \
    --challenge-
responses USERNAME=diego@example.com,SOFTWARE_TOKEN_MFA_CODE=000000 \
    --session AYABeExample...
```

Output:

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-ExAmPlE1"
        }
    }
}
```

Per ulteriori informazioni, consulta Flusso di autenticazione amministratore nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta AdminRespondToAuthChallenge 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.

```
   // Respond to an authentication challenge.
   public static void adminRespondToAuthChallenge(CognitoIdentityProviderClient
identityProviderClient,
           String userName, String clientId, String mfaCode, String session) {
       System.out.println("SOFTWARE_TOKEN_MFA challenge is generated");
       Map<String, String> challengeResponses = new HashMap<>();

       challengeResponses.put("USERNAME", userName);
       challengeResponses.put("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

       AdminRespondToAuthChallengeRequest respondToAuthChallengeRequest =
AdminRespondToAuthChallengeRequest.builder()
               .challengeName(ChallengeNameType.SOFTWARE_TOKEN_MFA)
               .clientId(clientId)
               .challengeResponses(challengeResponses)
               .session(session)
               .build();

       AdminRespondToAuthChallengeResponse respondToAuthChallengeResult =
identityProviderClient
               .adminRespondToAuthChallenge(respondToAuthChallengeRequest);

System.out.println("respondToAuthChallengeResult.getAuthenticationResult()"
               + respondToAuthChallengeResult.authenticationResult());
   }
```

- Per i dettagli sull'API, AdminRespondToAuthChallengeconsulta 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.

```
const adminRespondToAuthChallenge = ({
  userPoolId,
  clientId,
  username,
  totp,
  session,
}) => {
  const client = new CognitoIdentityProviderClient({});
  const command = new AdminRespondToAuthChallengeCommand({
    ChallengeName: ChallengeNameType.SOFTWARE_TOKEN_MFA,
    ChallengeResponses: {
      SOFTWARE_TOKEN_MFA_CODE: totp,
      USERNAME: username,
    },
    ClientId: clientId,
    UserPoolId: userPoolId,
    Session: session,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, AdminRespondToAuthChallengeconsulta 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.

```
// Respond to an authentication challenge.
suspend fun adminRespondToAuthChallenge(
    userName: String,
    clientIdVal: String?,
    mfaCode: String,
    sessionVal: String?,
) {
    println("SOFTWARE_TOKEN_MFA challenge is generated")
    val challengeResponsesOb = mutableMapOf<String, String>()
    challengeResponsesOb["USERNAME"] = userName
    challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

    val adminRespondToAuthChallengeRequest =
        AdminRespondToAuthChallengeRequest {
            challengeName = ChallengeNameType.SoftwareTokenMfa
            clientId = clientIdVal
            challengeResponses = challengeResponsesOb
            session = sessionVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val respondToAuthChallengeResult =
 identityProviderClient.adminRespondToAuthChallenge(adminRespondToAuthChallengeRequest)
        println("respondToAuthChallengeResult.getAuthenticationResult()
 ${respondToAuthChallengeResult.authenticationResult}")
    }
}
```

- Per i dettagli sull'API, AdminRespondToAuthChallengeconsulta AWS SDK for Kotlin 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.

Rispondi a una richiesta di autenticazione MFA fornendo un codice generato da un'applicazione MFA associata.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def respond_to_mfa_challenge(self, user_name, session, mfa_code):
        """
        Responds to a challenge for an MFA code. This completes the second step
 of
        a two-factor sign-in. When sign-in is successful, it returns an access
 token
        that can be used to get AWS credentials from Amazon Cognito.

        :param user_name: The name of the user who is signing in.
```

```
        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :param mfa_code: A code generated by the associated MFA application.
        :return: The result of the authentication. When successful, this contains
an
                 access token for the user.
        """
        try:
            kwargs = {
                "UserPoolId": self.user_pool_id,
                "ClientId": self.client_id,
                "ChallengeName": "SOFTWARE_TOKEN_MFA",
                "Session": session,
                "ChallengeResponses": {
                    "USERNAME": user_name,
                    "SOFTWARE_TOKEN_MFA_CODE": mfa_code,
                },
            }
            if self.client_secret is not None:
                kwargs["ChallengeResponses"]["SECRET_HASH"] = self._secret_hash(
                    user_name
                )
            response =
self.cognito_idp_client.admin_respond_to_auth_challenge(**kwargs)
            auth_result = response["AuthenticationResult"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "ExpiredCodeException":
                logger.warning(
                    "Your MFA code has expired or has been used already. You
might have "
                    "to wait a few seconds until your app shows you a new code."
                )
            else:
                logger.error(
                    "Couldn't respond to mfa challenge for %s. Here's why: %s:
%s",
                    user_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return auth_result
```

- Per i dettagli sull'API, consulta AdminRespondToAuthChallenge 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.
        " Build challenge responses
        DATA(lt_challenge_responses) = VALUE /aws1/
cl_cgpchallengerspstyp00=>tt_challengeresponsestype(
          ( VALUE /aws1/cl_cgpchallengerspstyp00=>ts_challengerspstype_maprow(
              key = 'USERNAME'
              value = NEW /aws1/cl_cgpchallengerspstyp00( iv_user_name ) ) )
          ( VALUE /aws1/cl_cgpchallengerspstyp00=>ts_challengerspstype_maprow(
              key = 'SOFTWARE_TOKEN_MFA_CODE'
              value = NEW /aws1/cl_cgpchallengerspstyp00( iv_mfa_code ) ) )
        ).

        " Add SECRET_HASH if provided
        IF iv_secret_hash IS NOT INITIAL.
          INSERT VALUE #(
            key = 'SECRET_HASH'
            value = NEW /aws1/cl_cgpchallengerspstyp00( iv_secret_hash )
          ) INTO TABLE lt_challenge_responses.
        ENDIF.

        DATA(lo_result) = lo_cgp->adminrespondtoauthchallenge(
          iv_userpoolid = iv_user_pool_id
          iv_clientid = iv_client_id
          iv_challengename = 'SOFTWARE_TOKEN_MFA'
          it_challengeresponses = lt_challenge_responses
```

```
          iv_session = iv_session
        ).

        oo_auth_result = lo_result->get_authenticationresult( ).

        IF oo_auth_result IS BOUND.
          MESSAGE 'MFA challenge completed successfully.' TYPE 'I'.
        ELSE.
          " Another challenge might be required
          DATA(lv_next_challenge) = lo_result->get_challengename( ).
          MESSAGE |Additional challenge required: { lv_next_challenge }.| TYPE
'I'.
        ENDIF.

      CATCH /aws1/cx_cgpcodemismatchex INTO DATA(lo_code_ex).
        MESSAGE 'Invalid MFA code provided.' TYPE 'E'.

      CATCH /aws1/cx_cgpexpiredcodeex INTO DATA(lo_expired_ex).
        MESSAGE 'MFA code has expired.' TYPE 'E'.

      CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
        MESSAGE 'Not authorized. Check MFA configuration.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, AdminRespondToAuthChallengeconsulta 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 AWSClientRuntime
import AWSCognitoIdentityProvider
```

```
    /// Respond to the authentication challenge received from Cognito after
    /// initiating an authentication session. This involves sending a current
    /// MFA code to the service.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - userName: The user's username.
    ///   - clientId: The app client ID.
    ///   - userPoolId: The user pool to sign into.
    ///   - mfaCode: The 6-digit MFA code currently displayed by the user's
    ///     authenticator.
    ///   - session: The authentication session to continue processing.
    func adminRespondToAuthChallenge(cipClient: CognitoIdentityProviderClient,
userName: String,
                                     clientId: String, userPoolId: String,
mfaCode: String,
                                     session: String) async {
        print("=====> SOFTWARE_TOKEN_MFA challenge is generated...")

        var challengeResponsesOb: [String: String] = [:]
        challengeResponsesOb["USERNAME"] = userName
        challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

        do {
            let output = try await cipClient.adminRespondToAuthChallenge(
                input: AdminRespondToAuthChallengeInput(
                    challengeName:
CognitoIdentityProviderClientTypes.ChallengeNameType.softwareTokenMfa,
                    challengeResponses: challengeResponsesOb,
                    clientId: clientId,
                    session: session,
                    userPoolId: userPoolId
                )
            )

            guard let authenticationResult = output.authenticationResult else {
                print("*** Unable to get authentication result.")
                return
            }

            print("=====> Authentication result (JWTs are redacted):")
            print(authenticationResult)
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
```

```
            return
        } catch _ as CodeMismatchException {
            print("*** The specified MFA code doesn't match the expected value.")
            return
        } catch _ as UserNotFoundException {
            print("*** The specified username, \(userName), doesn't exist.")
            return
        } catch _ as UserNotConfirmedException {
            print("*** The user \(userName) has not been confirmed.")
            return
        } catch let error as NotAuthorizedException {
            print("*** Unauthorized access. Reason: \(error.properties.message ??
"<unknown>")")
        } catch {
            print("*** Error responding to the MFA challenge.")
            return
        }
    }
```

- Per i dettagli sull'API, consulta la AdminRespondToAuthChallengeguida 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 **`AdminSetUserPassword`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`AdminSetUserPassword`.

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:

- Scrivere i dati di attività personalizzate con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito tramite un SDK

CLI

AWS CLI

Come impostare una password utente come amministratore

L'esempio `admin-set-user-password` seguente imposta in modo permanente la password per diego@example.com.

```
aws cognito-idp admin-set-user-password \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com \
    --password MyExamplePassword1! \
    --permanent
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Password, recupero delle password e policy delle password nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta AdminSetUserPassword 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"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}
```

```
// AdminSetUserPassword uses administrator credentials to set a password for a
 user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(ctx context.Context, userPoolId
 string, userName string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(ctx,
 &cognitoidentityprovider.AdminSetUserPasswordInput{
  Password:   aws.String(password),
  UserPoolId: aws.String(userPoolId),
  Username:   aws.String(userName),
  Permanent:  true,
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
 err)
  }
 }
 return err
}
```

- Per i dettagli sull'API, AdminSetUserPasswordconsulta AWS SDK per Go 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.

Utilizzare **`AssociateSoftwareToken`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`AssociateSoftwareToken`.

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:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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 an MFA token to authenticate the user with the authenticator.
    /// </summary>
    /// <param name="session">The session name.</param>
    /// <returns>The session name.</returns>
    public async Task<string> AssociateSoftwareTokenAsync(string session)
    {
        var softwareTokenRequest = new AssociateSoftwareTokenRequest
        {
            Session = session,
        };

        var tokenResponse = await
_cognitoService.AssociateSoftwareTokenAsync(softwareTokenRequest);
        var secretCode = tokenResponse.SecretCode;

        Console.WriteLine($"Use the following secret code to set up the
authenticator: {secretCode}");

        return tokenResponse.Session;
    }
```

- Per i dettagli sull'API, AssociateSoftwareTokenconsulta 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.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
 client(clientConfig);

        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenRequest
 request;
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenOutcome
 outcome =
                client.AssociateSoftwareToken(request);

        if (outcome.IsSuccess()) {
            std::cout
                    << "Enter this setup key into an authenticator app, for
 example Google Authenticator."
                    << std::endl;
            std::cout << "Setup key: " << outcome.GetResult().GetSecretCode()
                      << std::endl;
#ifdef USING_QR
            printAsterisksLine();
            std::cout << "\nOr scan the QR code in the file '" << QR_CODE_PATH <<
 "."
                      << std::endl;

            saveQRCode(std::string("otpauth://totp/") + userName + "?secret=" +
                       outcome.GetResult().GetSecretCode());
#endif // USING_QR
            session = outcome.GetResult().GetSession();
```

```
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::AssociateSoftwareToken. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Per i dettagli sull'API, AssociateSoftwareTokenconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come generare una chiave segreta per un'app di autenticazione MFA

L'esempio `associate-software-token` seguente genera una chiave privata TOTP per un utente che ha effettuato l'accesso e ha ricevuto un token di accesso. La chiave privata risultante può essere immessa manualmente in un'app di autenticazione oppure le applicazioni possono visualizzarla come codice QR che l'utente può scansionare.

```
aws cognito-idp associate-software-token \
    --access-token eyJra456defEXAMPLE
```

Output:

```
{
    "SecretCode": "QWERTYUIOP123456EXAMPLE"
}
```

Per ulteriori informazioni, consulta Autenticazione MFA con token di software TOTP nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta AssociateSoftwareToken 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.

```
   public static String getSecretForAppMFA(CognitoIdentityProviderClient
identityProviderClient, String session) {
       AssociateSoftwareTokenRequest softwareTokenRequest =
AssociateSoftwareTokenRequest.builder()
               .session(session)
               .build();

       AssociateSoftwareTokenResponse tokenResponse = identityProviderClient
               .associateSoftwareToken(softwareTokenRequest);
       String secretCode = tokenResponse.secretCode();
       System.out.println("Enter this token into Google Authenticator");
       System.out.println(secretCode);
       return tokenResponse.session();
   }
```

- Per i dettagli sull'API, AssociateSoftwareTokenconsulta 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.

```
const associateSoftwareToken = (session) => {
```

```
  const client = new CognitoIdentityProviderClient({});
  const command = new AssociateSoftwareTokenCommand({
    Session: session,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, AssociateSoftwareTokenconsulta 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 getSecretForAppMFA(sessionVal: String?): String? {
    val softwareTokenRequest =
        AssociateSoftwareTokenRequest {
            session = sessionVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val tokenResponse =
 identityProviderClient.associateSoftwareToken(softwareTokenRequest)
        val secretCode = tokenResponse.secretCode
        println("Enter this token into Google Authenticator")
        println(secretCode)
        return tokenResponse.session
    }
}
```

- Per i dettagli sull'API, AssociateSoftwareTokenconsulta AWS SDK for Kotlin 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.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def get_mfa_secret(self, session):
        """
        Gets a token that can be used to associate an MFA application with the
 user.

        :param session: Session information returned from a previous call to
 initiate
                        authentication.
        :return: An MFA token that can be used to set up an MFA application.
        """
        try:
            response =
 self.cognito_idp_client.associate_software_token(Session=session)
```

```
        except ClientError as err:
            logger.error(
                "Couldn't get MFA secret. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response
```

- Per i dettagli sull'API, consulta AssociateSoftwareToken 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.
    DATA(lo_result) = lo_cgp->associatesoftwaretoken(
      iv_session = iv_session
    ).

    ov_secret_code = lo_result->get_secretcode( ).

    MESSAGE 'MFA secret code generated successfully.' TYPE 'I'.

  CATCH /aws1/cx_cgpresourcenotfoundex INTO DATA(lo_ex).
    MESSAGE 'Session not found or expired.' TYPE 'E'.

  CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
    MESSAGE 'Not authorized to associate software token.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, AssociateSoftwareTokenconsulta 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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Request and display an MFA secret token that the user should enter
    /// into their authenticator to set it up for the user account.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - authSession: The authentication session to request an MFA secret
    ///     for.
    ///
    /// - Returns: A string containing the MFA secret token that should be
    ///   entered into the authenticator software.
    func getSecretForAppMFA(cipClient: CognitoIdentityProviderClient,
 authSession: String?) async -> String? {
        do {
            let output = try await cipClient.associateSoftwareToken(
                input: AssociateSoftwareTokenInput(
                    session: authSession
                )
            )

            guard let secretCode = output.secretCode else {
                print("*** Unable to get the secret code")
                return nil
            }
```

```
            print("=====> Enter this token into Google Authenticator:
\(secretCode)")
            return output.session
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return nil
        } catch {
            print("*** An unexpected error occurred getting the secret for the
app's MFA.")
            return nil
        }
    }
```

- Per i dettagli sull'API, consulta la AssociateSoftwareTokenguida 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 **`ConfirmDevice`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ConfirmDevice`.

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:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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>
    /// Initiates and confirms tracking of the device.
    /// </summary>
    /// <param name="accessToken">The user's access token.</param>
    /// <param name="deviceKey">The key of the device from Amazon Cognito.</
param>
    /// <param name="deviceName">The device name.</param>
    /// <returns></returns>
    public async Task<bool> ConfirmDeviceAsync(string accessToken, string
 deviceKey, string deviceName)
    {
        var request = new ConfirmDeviceRequest
        {
            AccessToken = accessToken,
            DeviceKey = deviceKey,
            DeviceName = deviceName
        };

        var response = await _cognitoService.ConfirmDeviceAsync(request);
        return response.UserConfirmationNecessary;
    }
```

- Per i dettagli sull'API, ConfirmDeviceconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come confermare un dispositivo utente

L'esempio `confirm-device` seguente aggiunge un nuovo dispositivo memorizzato per l'utente corrente.

```
aws cognito-idp confirm-device \
   --access-token eyJra456defEXAMPLE \
   --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
   --device-secret-verifier-
config PasswordVerifier=TXlWZXJpZmllclN0cmluZw,Salt=TXlTUlBTYWx0
```

Output:

```
{
     "UserConfirmationNecessary": false
}
```

Per ulteriori informazioni, consulta Utilizzo dei dispositivi utente nel pool di utenti nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta ConfirmDevice 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 confirmDevice = ({ deviceKey, accessToken, passwordVerifier, salt }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmDeviceCommand({
    DeviceKey: deviceKey,
    AccessToken: accessToken,
    DeviceSecretVerifierConfig: {
      PasswordVerifier: passwordVerifier,
      Salt: salt,
    },
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, ConfirmDeviceconsulta 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.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def confirm_mfa_device(
        self,
        user_name,
        device_key,
        device_group_key,
        device_password,
        access_token,
        aws_srp,
    ):
        """
        Confirms an MFA device to be tracked by Amazon Cognito. When a device is
        tracked, its key and password can be used to sign in without requiring a
 new
```

```
        MFA code from the MFA application.

        :param user_name: The user that is associated with the device.
        :param device_key: The key of the device, returned by Amazon Cognito.
        :param device_group_key: The group key of the device, returned by Amazon
Cognito.
        :param device_password: The password that is associated with the device.
        :param access_token: The user's access token.
        :param aws_srp: A class that helps with Secure Remote Password (SRP)
                        calculations. The scenario associated with this example
uses
                        the warrant package.
        :return: True when the user must confirm the device. Otherwise, False.
When
                 False, the device is automatically confirmed and tracked.
        """
        srp_helper = aws_srp.AWSSRP(
            username=user_name,
            password=device_password,
            pool_id="_",
            client_id=self.client_id,
            client_secret=None,
            client=self.cognito_idp_client,
        )
        device_and_pw = f"{device_group_key}{device_key}:{device_password}"
        device_and_pw_hash = aws_srp.hash_sha256(device_and_pw.encode("utf-8"))
        salt = aws_srp.pad_hex(aws_srp.get_random(16))
        x_value = aws_srp.hex_to_long(aws_srp.hex_hash(salt +
device_and_pw_hash))
        verifier = aws_srp.pad_hex(pow(srp_helper.val_g, x_value,
srp_helper.big_n))
        device_secret_verifier_config = {
            "PasswordVerifier": base64.standard_b64encode(
                bytearray.fromhex(verifier)
            ).decode("utf-8"),
            "Salt":
base64.standard_b64encode(bytearray.fromhex(salt)).decode("utf-8"),
        }
        try:
            response = self.cognito_idp_client.confirm_device(
                AccessToken=access_token,
                DeviceKey=device_key,
                DeviceSecretVerifierConfig=device_secret_verifier_config,
            )
```

```
            user_confirm = response["UserConfirmationNecessary"]
        except ClientError as err:
            logger.error(
                "Couldn't confirm mfa device %s. Here's why: %s: %s",
                device_key,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return user_confirm
```

- Per i dettagli sull'API, consulta ConfirmDevice 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.

Utilizzare **`ConfirmForgotPassword`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ConfirmForgotPassword`.

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:

- Migrare automaticamente gli utenti noti con una funzione Lambda

CLI

AWS CLI

Come confermare una password dimenticata

Questo esempio conferma una password dimenticata per diego@example.com.

Comando:

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- Per i dettagli sull'API, consulta ConfirmForgotPassword 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"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// ConfirmForgotPassword confirms a user with a confirmation code and a new
 password.
func (actor CognitoActions) ConfirmForgotPassword(ctx context.Context, clientId
 string, code string, userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(ctx,
 &cognitoidentityprovider.ConfirmForgotPasswordInput{
  ClientId:         aws.String(clientId),
  ConfirmationCode: aws.String(code),
  Password:         aws.String(password),
  Username:         aws.String(userName),
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
```

```
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
  }
 }
 return err
}
```

- Per i dettagli sull'API, ConfirmForgotPasswordconsulta AWS SDK per Go 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.

Utilizzare **`ConfirmSignUp`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ConfirmSignUp`.

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:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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>
    /// Confirm that the user has signed up.
    /// </summary>
    /// <param name="clientId">The Id of this application.</param>
```

```
    /// <param name="code">The confirmation code sent to the user.</param>
    /// <param name="userName">The username.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ConfirmSignupAsync(string clientId, string code,
string userName)
    {
        var signUpRequest = new ConfirmSignUpRequest
        {
            ClientId = clientId,
            ConfirmationCode = code,
            Username = userName,
        };

        var response = await _cognitoService.ConfirmSignUpAsync(signUpRequest);
        if (response.HttpStatusCode == HttpStatusCode.OK)
        {
            Console.WriteLine($"{userName} was confirmed");
            return true;
        }
        return false;
    }
```

- Per i dettagli sull'API, ConfirmSignUpconsulta 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.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);
```

```
        Aws::CognitoIdentityProvider::Model::ConfirmSignUpRequest request;
        request.SetClientId(clientID);
        request.SetConfirmationCode(confirmationCode);
        request.SetUsername(userName);

        Aws::CognitoIdentityProvider::Model::ConfirmSignUpOutcome outcome =
                client.ConfirmSignUp(request);

        if (outcome.IsSuccess()) {
            std::cout << "ConfirmSignup was Successful."
                      << std::endl;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::ConfirmSignUp. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Per i dettagli sull'API, ConfirmSignUpconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come confermare la registrazione

Questo esempio conferma la registrazione per il nome utente diego@example.com.

Comando:

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --confirmation-code CONF_CODE
```

- Per i dettagli sull'API, consulta ConfirmSignUp 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.

```
   public static void confirmSignUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String code,
           String userName) {
       try {
           ConfirmSignUpRequest signUpRequest = ConfirmSignUpRequest.builder()
                   .clientId(clientId)
                   .confirmationCode(code)
                   .username(userName)
                   .build();

           identityProviderClient.confirmSignUp(signUpRequest);
           System.out.println(userName + " was confirmed");

       } catch (CognitoIdentityProviderException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Per i dettagli sull'API, ConfirmSignUpconsulta 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.

```
const confirmSignUp = ({ clientId, username, code }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmSignUpCommand({
    ClientId: clientId,
    Username: username,
    ConfirmationCode: code,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, ConfirmSignUpconsulta 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 confirmSignUp(
    clientIdVal: String?,
    codeVal: String?,
    userNameVal: String?,
) {
    val signUpRequest =
        ConfirmSignUpRequest {
            clientId = clientIdVal
            confirmationCode = codeVal
            username = userNameVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        identityProviderClient.confirmSignUp(signUpRequest)
        println("$userNameVal  was confirmed")
```

```
    }
}
```

- Per i dettagli sull'API, ConfirmSignUpconsulta AWS SDK for Kotlin 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.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def confirm_user_sign_up(self, user_name, confirmation_code):
        """
        Confirms a previously created user. A user must be confirmed before they
        can sign in to Amazon Cognito.

        :param user_name: The name of the user to confirm.
```

```
        :param confirmation_code: The confirmation code sent to the user's
registered
                                  email address.
        :return: True when the confirmation succeeds.
        """
        try:
            kwargs = {
                "ClientId": self.client_id,
                "Username": user_name,
                "ConfirmationCode": confirmation_code,
            }
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            self.cognito_idp_client.confirm_sign_up(**kwargs)
        except ClientError as err:
            logger.error(
                "Couldn't confirm sign up for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return True
```

- Per i dettagli sull'API, consulta ConfirmSignUp AWSSDK for Python (Boto3) 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 AWSClientRuntime
import AWSCognitoIdentityProvider
```

```
    /// Submit a confirmation code for the specified user. This is the code as
    /// entered by the user after they've received it by email or text
    /// message.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The app client ID the user is signing up for.
    ///   - userName: The username of the user whose code is being sent.
    ///   - code: The user's confirmation code.
    ///
    /// - Returns: `true` if the code was successfully confirmed; otherwise
`false`.
    func confirmSignUp(cipClient: CognitoIdentityProviderClient, clientId:
String,
                       userName: String, code: String) async -> Bool {
        do {
            _ = try await cipClient.confirmSignUp(
                input: ConfirmSignUpInput(
                    clientId: clientId,
                    confirmationCode: code,
                    username: userName
                )
            )

            print("=====> \(userName) has been confirmed.")
            return true
        } catch {
            print("=====> \(userName)'s code was entered incorrectly.")
            return false
        }
    }
```

- Per i dettagli sull'API, consulta la ConfirmSignUpguida 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 **`CreateUserPool`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateUserPool`.

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:

- Nozioni di base sui bacini d'utenza

CLI

AWS CLI

Come creare un pool di utenti con configurazione minima

Questo esempio crea un pool di utenti denominato MyUserPool utilizzando valori predefiniti. Non ci sono attributi obbligatori né client di applicazioni. MFA e sicurezza avanzata sono disattivate.

Comando:

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

Output:

```
{
  "UserPool": {
      "SchemaAttributes": [
          {
              "Name": "sub",
              "StringAttributeConstraints": {
                  "MinLength": "1",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": true,
              "AttributeDataType": "String",
              "Mutable": false
          },
          {
              "Name": "name",
              "StringAttributeConstraints": {
```

```
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "given_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "family_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "middle_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "nickname",
            "StringAttributeConstraints": {
```

```
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "preferred_username",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "profile",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "picture",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "website",
            "StringAttributeConstraints": {
```

```
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "email",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "email_verified",
            "Mutable": true
        },
        {
            "Name": "gender",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "birthdate",
            "StringAttributeConstraints": {
                "MinLength": "10",
                "MaxLength": "10"
            },
            "DeveloperOnlyAttribute": false,
```

```
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "zoneinfo",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "locale",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
```

```
            {
                "Name": "address",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "updated_at",
                "NumberAttributeConstraints": {
                    "MinValue": "0"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "Number",
                "Mutable": true
            }
        ],
        "MfaConfiguration": "OFF",
        "Name": "MyUserPool",
        "LastModifiedDate": 1547833345.777,
        "AdminCreateUserConfig": {
            "UnusedAccountValidityDays": 7,
            "AllowAdminCreateUserOnly": false
        },
        "EmailConfiguration": {},
        "Policies": {
            "PasswordPolicy": {
                "RequireLowercase": true,
                "RequireSymbols": true,
                "RequireNumbers": true,
                "MinimumLength": 8,
                "RequireUppercase": true
            }
        },
        "CreationDate": 1547833345.777,
        "EstimatedNumberOfUsers": 0,
        "Id": "us-west-2_aaaaaaaaa",
        "LambdaConfig": {}
    }
```

```
}
```

Come creare un pool di utenti con due attributi richiesti

Questo esempio crea un pool di utenti MyUserPool. Il pool è configurato per accettare l'e-mail come attributo del nome utente. Inoltre, imposta l'indirizzo e-mail di origine su un indirizzo convalidato utilizzando Amazon Simple Email Service (Amazon SES).

Comando:

```
aws cognito-idp create-user-pool --pool-name MyUserPool --username-
attributes "email" --email-configuration=SourceArn="arn:aws:ses:us-
east-1:111111111111:identity/
jane@example.com",ReplyToEmailAddress="jane@example.com"
```

Output:

```
{
  "UserPool": {
      "SchemaAttributes": [
          {
              "Name": "sub",
              "StringAttributeConstraints": {
                  "MinLength": "1",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": true,
              "AttributeDataType": "String",
              "Mutable": false
          },
          {
              "Name": "name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
          {
```

```
                "Name": "given_name",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "family_name",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "middle_name",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "nickname",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
```

```
            "Name": "preferred_username",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "profile",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "picture",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "website",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
```

```
            "Name": "email",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "email_verified",
            "Mutable": true
        },
        {
            "Name": "gender",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "birthdate",
            "StringAttributeConstraints": {
                "MinLength": "10",
                "MaxLength": "10"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "zoneinfo",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
```

```
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "locale",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
        {
            "Name": "address",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
```

```
                "Mutable": true
            },
            {
                "Name": "updated_at",
                "NumberAttributeConstraints": {
                    "MinValue": "0"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "Number",
                "Mutable": true
            }
        ],
        "MfaConfiguration": "OFF",
        "Name": "MyUserPool",
        "LastModifiedDate": 1547837788.189,
        "AdminCreateUserConfig": {
            "UnusedAccountValidityDays": 7,
            "AllowAdminCreateUserOnly": false
        },
        "EmailConfiguration": {
            "ReplyToEmailAddress": "jane@example.com",
            "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
jane@example.com"
        },
        "Policies": {
            "PasswordPolicy": {
                "RequireLowercase": true,
                "RequireSymbols": true,
                "RequireNumbers": true,
                "MinimumLength": 8,
                "RequireUppercase": true
            }
        },
        "UsernameAttributes": [
            "email"
        ],
        "CreationDate": 1547837788.189,
        "EstimatedNumberOfUsers": 0,
        "Id": "us-west-2_aaaaaaaaa",
        "LambdaConfig": {}
    }
}
```

- Per i dettagli sull'API, consulta CreateUserPool 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.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolResponse;

/**
 * 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 CreateUserPool {
    public static void main(String[] args) {

        final String usage = """

                Usage:
                    <userPoolName>\s

                Where:
                    userPoolName - The name to give your user pool when it's
 created.
```

```
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String userPoolName = args[0];
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String id = createPool(cognitoClient, userPoolName);
        System.out.println("User pool ID: " + id);
        cognitoClient.close();
    }

    public static String createPool(CognitoIdentityProviderClient cognitoClient,
 String userPoolName) {
        try {
            CreateUserPoolRequest request = CreateUserPoolRequest.builder()
                    .poolName(userPoolName)
                    .build();

            CreateUserPoolResponse response =
 cognitoClient.createUserPool(request);
            return response.userPool().id();

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Per i dettagli sull'API, CreateUserPoolconsulta AWS SDK for Java 2.x 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.

Utilizzare **`CreateUserPoolClient`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateUserPoolClient`.

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:

- Nozioni di base sui bacini d'utenza

CLI

AWS CLI

Come creare un client del pool di utenti

L'`create-user-pool-client`esempio seguente crea un nuovo client con pool di utenti con un client secret, attributi di lettura e scrittura espliciti, accesso con flussi username-password e SRP, accesso con tre, accesso a un sottoinsieme di ambiti OAuth IdPs, analisi e una validità estesa della sessione di autenticazione. PinPoint

```
aws cognito-idp create-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-name MyTestClient \
    --generate-secret \
    --refresh-token-validity 10 \
    --access-token-validity 60 \
    --id-token-validity 60 \
    --token-validity-units AccessToken=minutes,IdToken=minutes,RefreshToken=days
 \
    --read-attributes email phone_number email_verified phone_number_verified \
    --write-attributes email phone_number \
    --explicit-auth-
flows ALLOW_USER_PASSWORD_AUTH ALLOW_USER_SRP_AUTH ALLOW_REFRESH_TOKEN_AUTH \
    --supported-identity-providers Google Facebook MyOIDC \
    --callback-urls https://www.amazon.com https://example.com http://
localhost:8001 myapp://example \
    --allowed-o-auth-flows code implicit \
```

```
    --allowed-o-auth-scopes openid profile aws.cognito.signin.user.admin solar-
system-data/asteroids.add \
    --allowed-o-auth-flows-user-pool-client \
    --analytics-configuration ApplicationArn=arn:aws:mobiletargeting:us-
west-2:767671399759:apps/thisisanexamplepinpointapplicationid,UserDataShared=TRUE
 \
    --prevent-user-existence-errors ENABLED \
    --enable-token-revocation \
    --enable-propagate-additional-user-context-data \
    --auth-session-validity 4
```

Output:

```
{
    "UserPoolClient": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientName": "MyTestClient",
        "ClientId": "123abc456defEXAMPLE",
        "ClientSecret": "this1234is5678my91011example1213client1415secret",
        "LastModifiedDate": 1726788459.464,
        "CreationDate": 1726788459.464,
        "RefreshTokenValidity": 10,
        "AccessTokenValidity": 60,
        "IdTokenValidity": 60,
        "TokenValidityUnits": {
            "AccessToken": "minutes",
            "IdToken": "minutes",
            "RefreshToken": "days"
        },
        "ReadAttributes": [
            "email_verified",
            "phone_number_verified",
            "phone_number",
            "email"
        ],
        "WriteAttributes": [
            "phone_number",
            "email"
        ],
        "ExplicitAuthFlows": [
            "ALLOW_USER_PASSWORD_AUTH",
            "ALLOW_USER_SRP_AUTH",
            "ALLOW_REFRESH_TOKEN_AUTH"
```

```
        ],
        "SupportedIdentityProviders": [
            "Google",
            "MyOIDC",
            "Facebook"
        ],
        "CallbackURLs": [
            "https://example.com",
            "https://www.amazon.com",
            "myapp://example",
            "http://localhost:8001"
        ],
        "AllowedOAuthFlows": [
            "implicit",
            "code"
        ],
        "AllowedOAuthScopes": [
            "aws.cognito.signin.user.admin",
            "openid",
            "profile",
            "solar-system-data/asteroids.add"
        ],
        "AllowedOAuthFlowsUserPoolClient": true,
        "AnalyticsConfiguration": {
            "ApplicationArn": "arn:aws:mobiletargeting:us-
west-2:123456789012:apps/thisisanexamplepinpointapplicationid",
            "RoleArn": "arn:aws:iam::123456789012:role/aws-service-role/cognito-
idp.amazonaws.com/AWSServiceRoleForAmazonCognitoIdp",
            "UserDataShared": true
        },
        "PreventUserExistenceErrors": "ENABLED",
        "EnableTokenRevocation": true,
        "EnablePropagateAdditionalUserContextData": true,
        "AuthSessionValidity": 4
    }
}
```

Per ulteriori informazioni, consulta Application-specific le impostazioni con i client di app nella Amazon Cognito Developer Guide.

- Per i dettagli sull'API, consulta CreateUserPoolClient 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.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolClientReques
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolClientRespon

/**
 * A user pool client app is an application that authenticates with Amazon
 * Cognito user pools.
 * When you create a user pool, you can configure app clients that allow mobile
 * or web applications
 * to call API operations to authenticate users, manage user attributes and
 * profiles,
 * and implement sign-up and sign-in flows.
 *
 * 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 CreateUserPoolClient {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <clientName> <userPoolId>\s
```

```
                Where:
                    clientName - The name for the user pool client to create.
                    userPoolId - The ID for the user pool.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String clientName = args[0];
        String userPoolId = args[1];
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createPoolClient(cognitoClient, clientName, userPoolId);
        cognitoClient.close();
    }

    public static void createPoolClient(CognitoIdentityProviderClient
 cognitoClient, String clientName,
            String userPoolId) {
        try {
            CreateUserPoolClientRequest request =
 CreateUserPoolClientRequest.builder()
                    .clientName(clientName)
                    .userPoolId(userPoolId)
                    .build();

            CreateUserPoolClientResponse response =
 cognitoClient.createUserPoolClient(request);
            System.out.println("User pool " +
 response.userPoolClient().clientName() + " created. ID: "
                    + response.userPoolClient().clientId());

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, CreateUserPoolClientconsulta AWS SDK for Java 2.x 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.

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:

- Confermare automaticamente gli utenti noti con una funzione Lambda
- Migrare automaticamente gli utenti noti con una funzione Lambda
- Scrivere i dati di attività personalizzate con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito tramite un SDK

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::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

        Aws::CognitoIdentityProvider::Model::DeleteUserRequest request;
        request.SetAccessToken(accessToken);
```

```
        Aws::CognitoIdentityProvider::Model::DeleteUserOutcome outcome =
                client.DeleteUser(request);

        if (outcome.IsSuccess()) {
            std::cout << "The user " << userName << " was deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::DeleteUser. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
```

- Per i dettagli sull'API, DeleteUserconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come eliminare un utente

Questo esempio elimina un utente.

Comando:

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- 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"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(ctx context.Context, userAccessToken
 string) error {
 _, err := actor.CognitoClient.DeleteUser(ctx,
 &cognitoidentityprovider.DeleteUserInput{
  AccessToken: aws.String(userAccessToken),
 })
 if err != nil {
  log.Printf("Couldn't delete user. Here's why: %v\n", err)
 }
 return err
}
```

- Per i dettagli sull'API, DeleteUserconsulta 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.

```
/**
 * Delete the signed-in user. Useful for allowing a user to delete their
 * own profile.
 * @param {{ region: string, accessToken: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").DeleteUserCommandOutput | null, unknown]>}
 */
export const deleteUser = async ({ region, accessToken }) => {
  try {
    const client = new CognitoIdentityProviderClient({ region });
    const response = await client.send(
      new DeleteUserCommand({ AccessToken: accessToken }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};
```

- Per i dettagli sull'API, DeleteUserconsulta 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.

Utilizzare **`DescribeTermsByClient`**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `DescribeTermsByClient`.

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:

- Token client e scoperta dei termini

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 describe_terms_by_client(
        self, user_pool_id: str, client_id: str, terms_name: str
    ) -> dict[str, Any]:
        """
        Looks up a Terms document associated with an app client by client ID,
        user pool ID, and terms name.

        :param user_pool_id: The ID of the user pool that contains the Terms
documents.
        :param client_id: The ID of the app client associated with the Terms.
        :param terms_name: The name of the terms document ('terms-of-use' or
                           'privacy-policy').
        :return: A dictionary with the Terms document details.
        """
        try:
            response = self.cognito_idp_client.describe_terms_by_client(
                UserPoolId=user_pool_id,
                ClientId=client_id,
                TermsName=terms_name,
            )
            terms = response.get("Terms", dict())
            logger.info(
                "Described terms '%s' for client %s in user pool %s.",
                terms_name,
                client_id,
                user_pool_id,
            )
            return terms
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error(
                    "No terms documents found for client %s, user pool %s, "
```

```
                    "and terms name '%s'.",
                    client_id,
                    user_pool_id,
                    terms_name,
                )
            raise
```

- Per i dettagli sull'API, consulta DescribeTermsByClient 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.

Utilizzare **`DescribeUserPoolClient`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeUserPoolClient`.

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:

- Token client e scoperta dei termini
- Nozioni di base sui bacini d'utenza

CLI

AWS CLI

Come descrivere un client del pool di utenti

Questo esempio descrive un client del pool di utenti.

Comando:

```
aws cognito-idp describe-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --
client-id 38fjsnc484p94kpqsnet7mpld0
```

Output:

```
{
  "UserPoolClient": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ClientName": "MyApp",
      "ClientId": "38fjsnc484p94kpqsnet7mpld0",
      "ClientSecret": "CLIENT_SECRET",
      "LastModifiedDate": 1548108676.163,
      "CreationDate": 1548108676.163,
      "RefreshTokenValidity": 30,
      "ReadAttributes": [
          "address",
          "birthdate",
          "custom:CustomAttr1",
          "custom:CustomAttr2",
          "email",
          "email_verified",
          "family_name",
          "gender",
          "given_name",
          "locale",
          "middle_name",
          "name",
          "nickname",
          "phone_number",
          "phone_number_verified",
          "picture",
          "preferred_username",
          "profile",
          "updated_at",
          "website",
          "zoneinfo"
      ],
      "WriteAttributes": [
          "address",
          "birthdate",
          "custom:CustomAttr1",
          "custom:CustomAttr2",
          "email",
          "family_name",
          "gender",
          "given_name",
          "locale",
          "middle_name",
```

```
            "name",
            "nickname",
            "phone_number",
            "picture",
            "preferred_username",
            "profile",
            "updated_at",
            "website",
            "zoneinfo"
        ],
        "ExplicitAuthFlows": [
            "ADMIN_NO_SRP_AUTH",
            "USER_PASSWORD_AUTH"
        ],
        "AllowedOAuthFlowsUserPoolClient": false
    }
}
```

- Per i dettagli sull'API, consulta DescribeUserPoolClient AWS CLICommand 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.

```
    def describe_user_pool_client(
        self, user_pool_id: str, client_id: str
    ) -> dict[str, Any]:
        """
        Describes an app client in a user pool, including the client secret.

        :param user_pool_id: The ID of the user pool that contains the app
 client.
        :param client_id: The ID of the app client to describe.
        :return: A dictionary with the app client configuration.
        """
        try:
```

```
            response = self.cognito_idp_client.describe_user_pool_client(
                UserPoolId=user_pool_id, ClientId=client_id
            )
            client_info = response.get("UserPoolClient", dict())
            logger.info(
                "Described app client %s in user pool %s.",
                client_id,
                user_pool_id,
            )
            return client_info
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error(
                    "The user pool %s or client %s does not exist. "
                    "Verify the stack deployed successfully and the outputs are
correct.",
                    user_pool_id,
                    client_id,
                )
            raise
```

- Per i dettagli sull'API, consulta DescribeUserPoolClient 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.

Utilizzare **`ForgotPassword`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ForgotPassword`.

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:

- Migrare automaticamente gli utenti noti con una funzione Lambda

CLI

AWS CLI

Come forzare la modifica della password

L'esempio `forgot-password` seguente invia un messaggio a jane@example.com per la modifica della password.

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpqsnet7mpld0 --
username jane@example.com
```

Output:

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

- Per i dettagli sull'API, consulta ForgotPassword 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"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
```

```
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// ForgotPassword starts a password recovery flow for a user. This flow typically
 sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(ctx context.Context, clientId string,
 userName string) (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(ctx,
 &cognitoidentityprovider.ForgotPasswordInput{
  ClientId: aws.String(clientId),
  Username: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't start password reset for user '%v'. Here;s why: %v\n",
 userName, err)
 }
 return output.CodeDeliveryDetails, err
}
```

- Per i dettagli sull'API, ForgotPasswordconsulta AWS SDK per Go 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.

Utilizzare **`GetClientToken`**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `GetClientToken`.

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:

- Token client e scoperta dei termini

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_client_token(
        self,
        client_id: str,
        client_secret: str,
        scopes: Optional[list[str]] = None,
    ) -> dict[str, Any]:
        """
        Obtains an M2M access token for an app client using its client ID
        and secret, without end-user sign-in.

        :param client_id: The ID of the app client.
        :param client_secret: The client secret for the app client.
        :param scopes: Optional list of scopes in the format
                       'resource-server-identifier/scope-name'. If omitted,
                       all scopes configured for the app client are authorized.
        :return: A dictionary containing the ClientAuthenticationResult with
                 AccessToken, ExpiresIn, and TokenType.
        """
        try:
            params = dict()
            params["ClientId"] = client_id
            params["Secret"] = client_secret
            if scopes is not None:
                params["Scopes"] = scopes

            response = self.cognito_idp_client.get_client_token(**params)
            auth_result = response.get("ClientAuthenticationResult", dict())
            logger.info(
                "Obtained M2M access token for client %s. Token type: %s, expires
in: %s seconds.",
                client_id,
                auth_result.get("TokenType", "Unknown"),
```

```
                auth_result.get("ExpiresIn", 0),
            )
            return auth_result
        except ClientError as err:
            if err.response["Error"]["Code"] == "NotAuthorizedException":
                logger.error(
                    "Not authorized for client %s. Verify that the client ID and
"
                    "secret are correct, and that the ALLOW_CLIENT_TOKEN_AUTH
flow "
                    "is enabled for the app client.",
                    client_id,
                )
            raise
```

- Per i dettagli sull'API, consulta GetClientToken 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.

Utilizzare **`InitiateAuth`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`InitiateAuth`.

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:

- Confermare automaticamente gli utenti noti con una funzione Lambda
- Migrare automaticamente gli utenti noti con una funzione Lambda
- Nozioni di base sui bacini d'utenza
- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA
- Scrivere i dati di attività personalizzate con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito tramite un SDK

.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>
    /// Initiate authorization.
    /// </summary>
    /// <param name="clientId">The client Id of the application.</param>
    /// <param name="userName">The name of the user who is authenticating.</
param>
    /// <param name="password">The password for the user who is authenticating.</
param>
    /// <returns>The response from the initiate auth request.</returns>
    public async Task<InitiateAuthResponse> InitiateAuthAsync(string clientId,
 string userName, string password)
    {
        var authParameters = new Dictionary<string, string>();
        authParameters.Add("USERNAME", userName);
        authParameters.Add("PASSWORD", password);

        var authRequest = new InitiateAuthRequest

        {
            ClientId = clientId,
            AuthParameters = authParameters,
            AuthFlow = AuthFlowType.USER_PASSWORD_AUTH,
        };

        var response = await _cognitoService.InitiateAuthAsync(authRequest);
        Console.WriteLine($"Result Challenge is : {response.ChallengeName}");

        return response;
    }
```

- Per i dettagli sull'API, InitiateAuthconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come effettuare l'accesso per un utente

L'esempio `initiate-auth` seguente accede a un utente con il flusso nome utente-password di base e senza richieste di verifica aggiuntive.

```
aws cognito-idp initiate-auth \
    --auth-flow USER_PASSWORD_AUTH \
    --client-id 1example23456789 \
    --analytics-metadata AnalyticsEndpointId=d70b2ba36a8c4dc5a04a0451aEXAMPLE \
    --auth-parameters USERNAME=testuser,PASSWORD=[Password] --user-context-
data EncodedData=mycontextdata --client-metadata MyTestKey=MyTestValue
```

Output:

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
        }
    }
}
```

Per ulteriori informazioni, consulta Autenticazione nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta InitiateAuth 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"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// SignIn signs in a user to Amazon Cognito using a username and password
 authentication flow.
func (actor CognitoActions) SignIn(ctx context.Context, clientId string, userName
 string, password string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(ctx,
 &cognitoidentityprovider.InitiateAuthInput{
  AuthFlow:       "USER_PASSWORD_AUTH",
  ClientId:       aws.String(clientId),
  AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
 })
 if err != nil {
  var resetRequired *types.PasswordResetRequiredException
  if errors.As(err, &resetRequired) {
   log.Println(*resetRequired.Message)
```

```
  } else {
   log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
  }
 } else {
  authResult = output.AuthenticationResult
 }
 return authResult, err
}
```

- Per i dettagli sull'API, InitiateAuthconsulta 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.

```
const initiateAuth = ({ username, password, clientId }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new InitiateAuthCommand({
    AuthFlow: AuthFlowType.USER_PASSWORD_AUTH,
    AuthParameters: {
      USERNAME: username,
      PASSWORD: password,
    },
    ClientId: clientId,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, InitiateAuthconsulta 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.

Questo esempio mostra come avviare l'autenticazione con un dispositivo monitorato. Per completare l'accesso, il client deve rispondere correttamente alle richieste di autenticazione SRP (Secure Remote Password).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def sign_in_with_tracked_device(
        self,
        user_name,
        password,
        device_key,
        device_group_key,
        device_password,
        aws_srp,
    ):
```

```
        """
        Signs in to Amazon Cognito as a user who has a tracked device. Signing in
        with a tracked device lets a user sign in without entering a new MFA
 code.

        Signing in with a tracked device requires that the client respond to the
 SRP
        protocol. The scenario associated with this example uses the warrant
 package
        to help with SRP calculations.

        For more information on SRP, see https://en.wikipedia.org/wiki/
Secure_Remote_Password_protocol.

        :param user_name: The user that is associated with the device.
        :param password: The user's password.
        :param device_key: The key of a tracked device.
        :param device_group_key: The group key of a tracked device.
        :param device_password: The password that is associated with the device.
        :param aws_srp: A class that helps with SRP calculations. The scenario
                        associated with this example uses the warrant package.
        :return: The result of the authentication. When successful, this contains
 an
                 access token for the user.
        """
        try:
            srp_helper = aws_srp.AWSSRP(
                username=user_name,
                password=device_password,
                pool_id="_",
                client_id=self.client_id,
                client_secret=None,
                client=self.cognito_idp_client,
            )

            response_init = self.cognito_idp_client.initiate_auth(
                ClientId=self.client_id,
                AuthFlow="USER_PASSWORD_AUTH",
                AuthParameters={
                    "USERNAME": user_name,
                    "PASSWORD": password,
                    "DEVICE_KEY": device_key,
                },
            )
```

```
            if response_init["ChallengeName"] != "DEVICE_SRP_AUTH":
                raise RuntimeError(
                    f"Expected DEVICE_SRP_AUTH challenge but got
{response_init['ChallengeName']}."
                )

            auth_params = srp_helper.get_auth_params()
            auth_params["DEVICE_KEY"] = device_key
            response_auth = self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_SRP_AUTH",
                ChallengeResponses=auth_params,
            )
            if response_auth["ChallengeName"] != "DEVICE_PASSWORD_VERIFIER":
                raise RuntimeError(
                    f"Expected DEVICE_PASSWORD_VERIFIER challenge but got "
                    f"{response_init['ChallengeName']}."
                )

            challenge_params = response_auth["ChallengeParameters"]
            challenge_params["USER_ID_FOR_SRP"] = device_group_key + device_key
            cr = srp_helper.process_challenge(challenge_params, {"USERNAME":
user_name})
            cr["USERNAME"] = user_name
            cr["DEVICE_KEY"] = device_key
            response_verifier =
self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_PASSWORD_VERIFIER",
                ChallengeResponses=cr,
            )
            auth_tokens = response_verifier["AuthenticationResult"]
        except ClientError as err:
            logger.error(
                "Couldn't start client sign in for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return auth_tokens
```

- Per i dettagli sull'API, consulta InitiateAuth 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.

Utilizzare **`ListUserPools`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListUserPools`.

.NET

SDK per .NET (v4)

> **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.

```
    /// <summary>
    /// List the Amazon Cognito user pools for an account.
    /// </summary>
    /// <returns>A list of UserPoolDescriptionType objects.</returns>
    public async Task<List<UserPoolDescriptionType>> ListUserPoolsAsync()
    {
        var userPools = new List<UserPoolDescriptionType>();

        var userPoolsPaginator = _cognitoService.Paginators.ListUserPools(new
ListUserPoolsRequest());

        await foreach (var response in userPoolsPaginator.Responses)
        {
            userPools.AddRange(response.UserPools);
        }

        return userPools;
    }
```

- Per i dettagli sull'API, ListUserPoolsconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come elencare i pool di utenti

L'`list-user-pools`esempio seguente elenca 3 dei pool di utenti disponibili nell' AWS account delle credenziali CLI correnti.

```
aws cognito-idp list-user-pools \
    --max-results 3
```

Output:

```
{
    "NextToken": "[Pagination token]",
    "UserPools": [
        {
            "CreationDate": 1681502497.741,
            "Id": "us-west-2_EXAMPLE1",
            "LambdaConfig": {
                "CustomMessage": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreSignUp": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGeneration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGenerationConfig": {
                    "LambdaArn": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                    "LambdaVersion": "V1_0"
                }
            },
            "LastModifiedDate": 1681502497.741,
            "Name": "user pool 1"
        },
        {
            "CreationDate": 1686064178.717,
```

```
            "Id": "us-west-2_EXAMPLE2",
            "LambdaConfig": {
            },
            "LastModifiedDate": 1686064178.873,
            "Name": "user pool 2"
        },
        {
            "CreationDate": 1627681712.237,
            "Id": "us-west-2_EXAMPLE3",
            "LambdaConfig": {
                "UserMigration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction"
            },
            "LastModifiedDate": 1678486942.479,
            "Name": "user pool 3"
        }
    ]
}
```

Per ulteriori informazioni, consulta Pool di utenti di Amazon Cognito nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, vedere ListUserPoolsin 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.

```
package main

import (
 "context"
 "fmt"
 "log"
```

```
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Notification
 Service
// (Amazon SNS) client and list the topics 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
 }
 cognitoClient := cognitoidentityprovider.NewFromConfig(sdkConfig)
 fmt.Println("Let's list the user pools for your account.")
 var pools []types.UserPoolDescriptionType
 paginator := cognitoidentityprovider.NewListUserPoolsPaginator(
  cognitoClient, &cognitoidentityprovider.ListUserPoolsInput{MaxResults:
 aws.Int32(10)})
 for paginator.HasMorePages() {
  output, err := paginator.NextPage(ctx)
  if err != nil {
   log.Printf("Couldn't get user pools. Here's why: %v\n", err)
  } else {
   pools = append(pools, output.UserPools...)
  }
 }
 if len(pools) == 0 {
  fmt.Println("You don't have any user pools!")
 } else {
  for _, pool := range pools {
   fmt.Printf("\t%v: %v\n", *pool.Name, *pool.Id)
  }
 }
}
```

- Per i dettagli sull'API, ListUserPoolsconsulta 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.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsRequest;

/**
 * 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 ListUserPools {
    public static void main(String[] args) {
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllUserPools(cognitoClient);
        cognitoClient.close();
```

```
    }

    public static void listAllUserPools(CognitoIdentityProviderClient
 cognitoClient) {
        try {
            ListUserPoolsRequest request = ListUserPoolsRequest.builder()
                    .maxResults(10)
                    .build();

            ListUserPoolsResponse response =
 cognitoClient.listUserPools(request);
            response.userPools().forEach(userpool -> {
                System.out.println("User pool " + userpool.name() + ", User ID "
 + userpool.id());
            });

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, ListUserPoolsconsulta AWS SDK for Java 2.x 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 show_pools(client: &Client) -> Result<(), Error> {
    let response = client.list_user_pools().max_results(10).send().await?;
    let pools = response.user_pools();
    println!("User pools:");
    for pool in pools {
        println!("  ID:             {}", pool.id().unwrap_or_default());
```

```
        println!("  Name:            {}", pool.name().unwrap_or_default());
        println!("  Lambda Config:   {:?}", pool.lambda_config().unwrap());
        println!(
            "  Last modified:   {}",
            pool.last_modified_date().unwrap().to_chrono_utc()?
        );
        println!(
            "  Creation date:   {:?}",
            pool.creation_date().unwrap().to_chrono_utc()
        );
        println!();
    }
    println!("Next token: {}", response.next_token().unwrap_or_default());

    Ok(())
}
```

- Per i dettagli sulle API, consulta il riferimento ListUserPoolsall'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.

Utilizzare **`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 nei seguenti esempi di codice:

- Nozioni di base sui bacini d'utenza
- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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 a list of users for the Amazon Cognito user pool.
/// </summary>
/// <param name="userPoolId">The user pool ID.</param>
/// <returns>A list of users.</returns>
public async Task<List<UserType>> ListUsersAsync(string userPoolId)
{
    var request = new ListUsersRequest
    {
        UserPoolId = userPoolId
    };

    var users = new List<UserType>();

    var usersPaginator = _cognitoService.Paginators.ListUsers(request);
    await foreach (var response in usersPaginator.Responses)
    {
        users.AddRange(response.Users);
    }

    return users;
}
```

- Per i dettagli sull'API, ListUsersconsulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Esempio 1: come restituire l'elenco degli utenti con un filtro lato server

L'esempio `list-users` seguente elenca 3 utenti nel pool di utenti richiesto i cui indirizzi e-mail iniziano con `testuser`.

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
    --filter email^=\"testuser\" \
    --max-items 3
```

Output:

```
{
    "PaginationToken": "efgh5678EXAMPLE",
    "Users": [
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "eaad0219-2117-439f-8d46-4db20e59268f"
                },
                {
                    "Name": "email",
                    "Value": "testuser@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1682955829.578,
            "UserLastModifiedDate": 1689030181.63,
            "UserStatus": "CONFIRMED",
            "Username": "testuser"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "3b994cfd-0b07-4581-be46-3c82f9a70c90"
                },
                {
                    "Name": "email",
                    "Value": "testuser2@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427979.201,
```

```
            "UserLastModifiedDate": 1684427979.201,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser2"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "5929e0d1-4c34-42d1-9b79-a5ecacfe66f7"
                },
                {
                    "Name": "email",
                    "Value": "testuser3@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427823.641,
            "UserLastModifiedDate": 1684427823.641,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser3@example.com"
        }
    ]
}
```

Per ulteriori informazioni, consulta Gestione e ricerca degli utenti nella Guida per gli sviluppatori di Amazon Cognito.

Esempio 2: come restituire l'elenco degli utenti con un filtro lato client

L'esempio `list-users` seguente elenca gli attributi di tre utenti che dispongono di un attributo (in questo caso l'indirizzo e-mail contenente il dominio e-mail "@example .com"). Verranno visualizzati anche eventuali altri attributi contenuti in questa stringa. Il secondo utente non dispone di attributi corrispondenti alla query e pertanto è escluso dall'output visualizzato, ma non dalla risposta del server.

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
    --max-items 3
    --query Users\[\*\].Attributes\[\?Value\.contains\(\@\,\'@example.com\'\)\]
```

Output:

```
[
    [
        {
            "Name": "email",
            "Value": "admin@example.com"
        }
    ],
    [],
    [
        {
            "Name": "email",
            "Value": "operator@example.com"
        }
    ]
]
```

Per ulteriori informazioni, consulta Gestione e ricerca degli utenti nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta ListUsers 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.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUsersRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUsersResponse;
```

```
/**
 * 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) {

        final String usage = """

                Usage:
                    <userPoolId>\s

                Where:
                    userPoolId - The ID given to your user pool when it's
 created.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String userPoolId = args[0];
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllUsers(cognitoClient, userPoolId);
        listUsersFilter(cognitoClient, userPoolId);
        cognitoClient.close();
    }

    public static void listAllUsers(CognitoIdentityProviderClient cognitoClient,
 String userPoolId) {
        try {
            ListUsersRequest usersRequest = ListUsersRequest.builder()
                    .userPoolId(userPoolId)
                    .build();
```

```
            ListUsersResponse response = cognitoClient.listUsers(usersRequest);
            response.users().forEach(user -> {
                System.out.println("User " + user.username() + " Status " +
 user.userStatus() + " Created "
                        + user.userCreateDate());
            });

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Shows how to list users by using a filter.
    public static void listUsersFilter(CognitoIdentityProviderClient
 cognitoClient, String userPoolId) {

        try {
            String filter = "email = \"tblue@noserver.com\"";
            ListUsersRequest usersRequest = ListUsersRequest.builder()
                    .userPoolId(userPoolId)
                    .filter(filter)
                    .build();

            ListUsersResponse response = cognitoClient.listUsers(usersRequest);
            response.users().forEach(user -> {
                System.out.println("User with filter applied " + user.username()
 + " Status " + user.userStatus()
                        + " Created " + user.userCreateDate());
            });

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, ListUsersconsulta 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.

```
const listUsers = ({ userPoolId }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ListUsersCommand({
    UserPoolId: userPoolId,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, ListUsersconsulta 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(userPoolId: String) {
    val request =
        ListUsersRequest {
            this.userPoolId = userPoolId
        }
```

```
    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { cognitoClient ->
        val response = cognitoClient.listUsers(request)
        response.users?.forEach { user ->
            println("The user name is ${user.username}")
        }
    }
}
```

- Per i dettagli sull'API, ListUsersconsulta AWS SDK for Kotlin 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.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def list_users(self):
```

```
        """
        Returns a list of the users in the current user pool.

        :return: The list of users.
        """
        try:
            response =
self.cognito_idp_client.list_users(UserPoolId=self.user_pool_id)
            users = response["Users"]
        except ClientError as err:
            logger.error(
                "Couldn't list users for %s. Here's why: %s: %s",
                self.user_pool_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return users
```

- Per i dettagli sull'API, consulta ListUsers 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.
        DATA(lo_result) = lo_cgp->listusers(
          iv_userpoolid = iv_user_pool_id
        ).

        ot_users = lo_result->get_users( ).

        MESSAGE |Found { lines( ot_users ) } users in the pool.| TYPE 'I'.
```

```
    CATCH /aws1/cx_cgpresourcenotfoundex INTO DATA(lo_ex).
      MESSAGE |User pool { iv_user_pool_id } not found.| TYPE 'E'.

    CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
      MESSAGE 'Not authorized to list 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.

```
    do {
        let output = try await cognitoClient.listUsers(
            input: ListUsersInput(
                userPoolId: poolId
            )
        )

        guard let users = output.users else {
            print("No users found.")
            return
        }

        print("\(users.count) user(s) found.")
        for user in users {
            print("  \(user.username ?? "<unknown>")")
        }
    } catch _ as NotAuthorizedException {
        print("*** Please authenticate with AWS before using this command.")
        return
    } catch _ as ResourceNotFoundException {
        print("*** The specified User Pool was not found.")
```

```
            return
        } catch {
            print("*** An unexpected type of error occurred.")
            return
        }
```

- 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.

Utilizzare **`ResendConfirmationCode`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ResendConfirmationCode`.

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:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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>
    /// Send a new confirmation code to a user.
    /// </summary>
    /// <param name="clientId">The Id of the client application.</param>
    /// <param name="userName">The username of user who will receive the code.</
param>
    /// <returns>The delivery details.</returns>
    public async Task<CodeDeliveryDetailsType> ResendConfirmationCodeAsync(string
 clientId, string userName)
```

```
    {
        var codeRequest = new ResendConfirmationCodeRequest
        {
            ClientId = clientId,
            Username = userName,
        };

        var response = await
_cognitoService.ResendConfirmationCodeAsync(codeRequest);

        Console.WriteLine($"Method of delivery is
{response.CodeDeliveryDetails.DeliveryMedium}");

        return response.CodeDeliveryDetails;
    }
```

- Per i dettagli sull'API, ResendConfirmationCodeconsulta 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.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

        Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeRequest
request;
        request.SetUsername(userName);
```

```
        request.SetClientId(clientID);

        Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeOutcome
outcome =
                client.ResendConfirmationCode(request);

        if (outcome.IsSuccess()) {
            std::cout
                    << "CognitoIdentityProvider::ResendConfirmationCode was
successful."
                    << std::endl;
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::ResendConfirmationCode. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Per i dettagli sull'API, ResendConfirmationCodeconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come inviare nuovamente un codice di conferma

L'esempio `resend-confirmation-code` seguente invia un codice di conferma all'utente `jane`.

```
aws cognito-idp resend-confirmation-code \
    --client-id 12a3b456c7de890f11g123hijk \
    --username jane
```

Output:

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
```

```
        "AttributeName": "email"
    }
}
```

Per ulteriori informazioni, consulta Registrazione e conferma degli account utente nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta ResendConfirmationCode 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.

```
   public static void resendConfirmationCode(CognitoIdentityProviderClient
identityProviderClient, String clientId,
           String userName) {
       try {
           ResendConfirmationCodeRequest codeRequest =
ResendConfirmationCodeRequest.builder()
                   .clientId(clientId)
                   .username(userName)
                   .build();

           ResendConfirmationCodeResponse response =
identityProviderClient.resendConfirmationCode(codeRequest);
           System.out.println("Method of delivery is " +
response.codeDeliveryDetails().deliveryMediumAsString());

       } catch (CognitoIdentityProviderException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Per i dettagli sull'API, ResendConfirmationCodeconsulta 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.

```
const resendConfirmationCode = ({ clientId, username }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ResendConfirmationCodeCommand({
    ClientId: clientId,
    Username: username,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, ResendConfirmationCodeconsulta 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 resendConfirmationCode(
```

```
    clientIdVal: String?,
    userNameVal: String?,
) {
    val codeRequest =
        ResendConfirmationCodeRequest {
            clientId = clientIdVal
            username = userNameVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.resendConfirmationCode(codeRequest)
        println("Method of delivery is " +
 (response.codeDeliveryDetails?.deliveryMedium))
    }
}
```

- Per i dettagli sull'API, ResendConfirmationCodeconsulta AWS SDK for Kotlin 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.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
```

```
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def resend_confirmation(self, user_name):
        """
        Prompts Amazon Cognito to resend an email with a new confirmation code.

        :param user_name: The name of the user who will receive the email.
        :return: Delivery information about where the email is sent.
        """
        try:
            kwargs = {"ClientId": self.client_id, "Username": user_name}
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            response = self.cognito_idp_client.resend_confirmation_code(**kwargs)
            delivery = response["CodeDeliveryDetails"]
        except ClientError as err:
            logger.error(
                "Couldn't resend confirmation to %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return delivery
```

- Per i dettagli sull'API, consulta ResendConfirmationCode AWSSDK for Python (Boto3) 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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Requests a new confirmation code be sent to the given user's contact
    /// method.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The application client ID.
    ///   - userName: The user to resend a code for.
    ///
    /// - Returns: `true` if a new code was sent successfully, otherwise
    ///   `false`.
    func resendConfirmationCode(cipClient: CognitoIdentityProviderClient,
 clientId: String,
                                userName: String) async -> Bool {
        do {
            let output = try await cipClient.resendConfirmationCode(
                input: ResendConfirmationCodeInput(
                    clientId: clientId,
                    username: userName
                )
            )

            guard let deliveryMedium = output.codeDeliveryDetails?.deliveryMedium
 else {
                print("*** Unable to get the delivery method for the resent
 code.")
                return false
            }

            print("=====> A new code has been sent by \(deliveryMedium)")
```

```
            return true
        } catch {
            print("*** Unable to resend the confirmation code to user
\(userName).")
            return false
        }
    }
```

- Per i dettagli sull'API, consulta la ResendConfirmationCodeguida 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 **`RespondToAuthChallenge`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`RespondToAuthChallenge`.

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:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

CLI

AWS CLI

Esempio 1: come rispondere a una richiesta di verifica NEW_PASSWORD_REQUIRED

L'esempio `respond-to-auth-challenge` seguente risponde alla richiesta di verifica NEW_PASSWORD_REQUIRED restituita da initiate-auth. Imposta una password per l'utente `jane@example.com`.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --challenge-name NEW_PASSWORD_REQUIRED \
    --challenge-responses USERNAME=jane@example.com,NEW_PASSWORD=[Password] \
    --session AYABeEv5HklEXAMPLE
```

Output:

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "ACCESS_TOKEN",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "REFRESH_TOKEN",
        "IdToken": "ID_TOKEN",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-wt2ha1Zd"
        }
    }
}
```

Per ulteriori informazioni, consulta Autenticazione nella Guida per gli sviluppatori di Amazon Cognito.

Esempio 2: come rispondere a una richiesta di verifica SELECT_MFA_TYPE

L'esempio `respond-to-auth-challenge` seguente sceglie TOTP MFA come opzione MFA per l'utente corrente. All'utente viene richiesto di selezionare un tipo di autenticazione MFA e quindi di inserire il proprio codice MFA.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789
    --session AYABeEv5HklEXAMPLE
    --challenge-name SELECT_MFA_TYPE
    --challenge-responses USERNAME=testuser,ANSWER=SOFTWARE_TOKEN_MFA
```

Output:

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeEv5HklEXAMPLE",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "transparent"
    }
}
```

Per ulteriori informazioni, consulta Aggiunta di MFA nella Guida per gli sviluppatori di Amazon Cognito.

Esempio 3: come rispondere a una richiesta di verifica SOFTWARE_TOKEN_MFA

L'esempio `respond-to-auth-challenge` seguente fornisce un codice MFA TOTP e completa l'accesso.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --session AYABeEv5HklEXAMPLE \
    --challenge-name SOFTWARE_TOKEN_MFA \
    --challenge-responses USERNAME=testuser,SOFTWARE_TOKEN_MFA_CODE=123456
```

Output:

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
        }
    }
}
```

Per ulteriori informazioni, consulta Aggiunta di MFA nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta RespondToAuthChallenge 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 respondToAuthChallenge = ({
  clientId,
  username,
  session,
  userPoolId,
  code,
}) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new RespondToAuthChallengeCommand({
    ChallengeName: ChallengeNameType.SOFTWARE_TOKEN_MFA,
    ChallengeResponses: {
      SOFTWARE_TOKEN_MFA_CODE: code,
      USERNAME: username,
    },
    ClientId: clientId,
    UserPoolId: userPoolId,
    Session: session,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, RespondToAuthChallengeconsulta 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.

Accedi con un dispositivo monitorato. Per completare l'accesso, il client deve rispondere correttamente alle richieste di autenticazione SRP (Secure Remote Password).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def sign_in_with_tracked_device(
        self,
        user_name,
        password,
        device_key,
        device_group_key,
        device_password,
        aws_srp,
    ):
        """
```

```
        Signs in to Amazon Cognito as a user who has a tracked device. Signing in
        with a tracked device lets a user sign in without entering a new MFA
 code.

        Signing in with a tracked device requires that the client respond to the
 SRP
        protocol. The scenario associated with this example uses the warrant
 package
        to help with SRP calculations.

        For more information on SRP, see https://en.wikipedia.org/wiki/
Secure_Remote_Password_protocol.

        :param user_name: The user that is associated with the device.
        :param password: The user's password.
        :param device_key: The key of a tracked device.
        :param device_group_key: The group key of a tracked device.
        :param device_password: The password that is associated with the device.
        :param aws_srp: A class that helps with SRP calculations. The scenario
                        associated with this example uses the warrant package.
        :return: The result of the authentication. When successful, this contains
 an
                 access token for the user.
        """
        try:
            srp_helper = aws_srp.AWSSRP(
                username=user_name,
                password=device_password,
                pool_id="_",
                client_id=self.client_id,
                client_secret=None,
                client=self.cognito_idp_client,
            )

            response_init = self.cognito_idp_client.initiate_auth(
                ClientId=self.client_id,
                AuthFlow="USER_PASSWORD_AUTH",
                AuthParameters={
                    "USERNAME": user_name,
                    "PASSWORD": password,
                    "DEVICE_KEY": device_key,
                },
            )
            if response_init["ChallengeName"] != "DEVICE_SRP_AUTH":
```

```
                raise RuntimeError(
                    f"Expected DEVICE_SRP_AUTH challenge but got
{response_init['ChallengeName']}."
                )

            auth_params = srp_helper.get_auth_params()
            auth_params["DEVICE_KEY"] = device_key
            response_auth = self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_SRP_AUTH",
                ChallengeResponses=auth_params,
            )
            if response_auth["ChallengeName"] != "DEVICE_PASSWORD_VERIFIER":
                raise RuntimeError(
                    f"Expected DEVICE_PASSWORD_VERIFIER challenge but got "
                    f"{response_init['ChallengeName']}."
                )

            challenge_params = response_auth["ChallengeParameters"]
            challenge_params["USER_ID_FOR_SRP"] = device_group_key + device_key
            cr = srp_helper.process_challenge(challenge_params, {"USERNAME":
user_name})
            cr["USERNAME"] = user_name
            cr["DEVICE_KEY"] = device_key
            response_verifier =
self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_PASSWORD_VERIFIER",
                ChallengeResponses=cr,
            )
            auth_tokens = response_verifier["AuthenticationResult"]
        except ClientError as err:
            logger.error(
                "Couldn't start client sign in for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return auth_tokens
```

- Per i dettagli sull'API, consulta RespondToAuthChallenge 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.

Utilizzare **`SignUp`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`SignUp`.

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:

- Confermare automaticamente gli utenti noti con una funzione Lambda
- Migrare automaticamente gli utenti noti con una funzione Lambda
- Nozioni di base sui bacini d'utenza
- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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>
    /// Sign up a new user.
    /// </summary>
    /// <param name="clientId">The client Id of the application.</param>
    /// <param name="userName">The username to use.</param>
    /// <param name="password">The user's password.</param>
    /// <param name="email">The email address of the user.</param>
    /// <returns>A Boolean value indicating whether the user was confirmed.</
returns>
```

```
    public async Task<bool> SignUpAsync(string clientId, string userName, string
password, string email)
    {
        var userAttrs = new AttributeType
        {
            Name = "email",
            Value = email,
        };

        var userAttrsList = new List<AttributeType>();

        userAttrsList.Add(userAttrs);

        var signUpRequest = new SignUpRequest
        {
            UserAttributes = userAttrsList,
            Username = userName,
            ClientId = clientId,
            Password = password
        };

        var response = await _cognitoService.SignUpAsync(signUpRequest);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, SignUpconsulta 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.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";
```

```
   Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

       Aws::CognitoIdentityProvider::Model::SignUpRequest request;
       request.AddUserAttributes(
               Aws::CognitoIdentityProvider::Model::AttributeType().WithName(
                       "email").WithValue(email));
       request.SetUsername(userName);
       request.SetPassword(password);
       request.SetClientId(clientID);
       Aws::CognitoIdentityProvider::Model::SignUpOutcome outcome =
               client.SignUp(request);

       if (outcome.IsSuccess()) {
           std::cout << "The signup request for " << userName << " was
successful."
                     << std::endl;
       }
       else if (outcome.GetError().GetErrorType() ==

Aws::CognitoIdentityProvider::CognitoIdentityProviderErrors::USERNAME_EXISTS) {
           std::cout
                   << "The username already exists. Please enter a different
username."
                   << std::endl;
           userExists = true;
       }
       else {
           std::cerr << "Error with CognitoIdentityProvider::SignUpRequest. "
                     << outcome.GetError().GetMessage()
                     << std::endl;
           return false;
       }
```

- Per i dettagli sull'API, SignUpconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come registrare un utente

In questo esempio viene registrato jane@example.com.

Comando:

```
aws cognito-idp sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username jane@example.com --password PASSWORD --user-attributes
 Name="email",Value="jane@example.com" Name="name",Value="Jane"
```

Output:

```
{
  "UserConfirmed": false,
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"
}
```

- Per i dettagli sull'API, consulta SignUp 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"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
```

```
}



// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(ctx context.Context, clientId string, userName
 string, password string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(ctx,
 &cognitoidentityprovider.SignUpInput{
  ClientId: aws.String(clientId),
  Password: aws.String(password),
  Username: aws.String(userName),
  UserAttributes: []types.AttributeType{
   {Name: aws.String("email"), Value: aws.String(userEmail)},
  },
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
  }
 } else {
  confirmed = output.UserConfirmed
 }
 return confirmed, err
}
```

- Per i dettagli sull'API, SignUpconsulta 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.

```
   public static void signUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String userName,
           String password, String email) {
       AttributeType userAttrs = AttributeType.builder()
               .name("email")
               .value(email)
               .build();

       List<AttributeType> userAttrsList = new ArrayList<>();
       userAttrsList.add(userAttrs);
       try {
           SignUpRequest signUpRequest = SignUpRequest.builder()
                   .userAttributes(userAttrsList)
                   .username(userName)
                   .clientId(clientId)
                   .password(password)
                   .build();

           identityProviderClient.signUp(signUpRequest);
           System.out.println("User has been signed up ");

       } catch (CognitoIdentityProviderException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Per i dettagli sull'API, SignUpconsulta 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.

```
const signUp = ({ clientId, username, password, email }) => {
```

```
  const client = new CognitoIdentityProviderClient({});

  const command = new SignUpCommand({
    ClientId: clientId,
    Username: username,
    Password: password,
    UserAttributes: [{ Name: "email", Value: email }],
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, SignUpconsulta 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 signUp(
    clientIdVal: String?,
    userNameVal: String?,
    passwordVal: String?,
    emailVal: String?,
) {
    val userAttrs =
        AttributeType {
            name = "email"
            value = emailVal
        }

    val userAttrsList = mutableListOf<AttributeType>()
    userAttrsList.add(userAttrs)
    val signUpRequest =
        SignUpRequest {
            userAttributes = userAttrsList
```

```
            username = userNameVal
            clientId = clientIdVal
            password = passwordVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        identityProviderClient.signUp(signUpRequest)
        println("User has been signed up")
    }
}
```

- Per i dettagli sull'API, SignUpconsulta AWS SDK for Kotlin 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.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret
```

```
   def sign_up_user(self, user_name, password, user_email):
       """
       Signs up a new user with Amazon Cognito. This action prompts Amazon
Cognito
       to send an email to the specified email address. The email contains a
code that
       can be used to confirm the user.

       When the user already exists, the user status is checked to determine
whether
       the user has been confirmed.

       :param user_name: The user name that identifies the new user.
       :param password: The password for the new user.
       :param user_email: The email address for the new user.
       :return: True when the user is already confirmed with Amazon Cognito.
                Otherwise, false.
       """
       try:
           kwargs = {
               "ClientId": self.client_id,
               "Username": user_name,
               "Password": password,
               "UserAttributes": [{"Name": "email", "Value": user_email}],
           }
           if self.client_secret is not None:
               kwargs["SecretHash"] = self._secret_hash(user_name)
           response = self.cognito_idp_client.sign_up(**kwargs)
           confirmed = response["UserConfirmed"]
       except ClientError as err:
           if err.response["Error"]["Code"] == "UsernameExistsException":
               response = self.cognito_idp_client.admin_get_user(
                   UserPoolId=self.user_pool_id, Username=user_name
               )
               logger.warning(
                   "User %s exists and is %s.", user_name,
response["UserStatus"]
               )
               confirmed = response["UserStatus"] == "CONFIRMED"
           else:
               logger.error(
                   "Couldn't sign up %s. Here's why: %s: %s",
```

```
                    user_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        return confirmed
```

- Per i dettagli sull'API, consulta SignUp AWSSDK for Python (Boto3) 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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Create a new user in a user pool.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The ID of the app client to create a user for.
    ///   - userName: The username for the new user.
    ///   - password: The new user's password.
    ///   - email: The new user's email address.
    ///
    /// - Returns: `true` if successful; otherwise `false`.
    func signUp(cipClient: CognitoIdentityProviderClient, clientId: String,
 userName: String, password: String, email: String) async -> Bool {
        let emailAttr = CognitoIdentityProviderClientTypes.AttributeType(
            name: "email",
            value: email
        )

        let userAttrsList = [emailAttr]
```

```
        do {
            _ = try await cipClient.signUp(
                input: SignUpInput(
                    clientId: clientId,
                    password: password,
                    userAttributes: userAttrsList,
                    username: userName
                )

            )

            print("=====> User \(userName) signed up.")
        } catch _ as AWSCognitoIdentityProvider.UsernameExistsException {
            print("*** The username \(userName) already exists. Please use a
different one.")
            return false
        } catch let error as AWSCognitoIdentityProvider.InvalidPasswordException
{
            print("*** Error: The specified password is invalid. Reason:
\(error.properties.message ?? "<none available>").")
            return false
        } catch _ as AWSCognitoIdentityProvider.ResourceNotFoundException {
            print("*** Error: The specified client ID (\(clientId)) doesn't
exist.")
            return false
        } catch {
            print("*** Unexpected error: \(error)")
            return false
        }

        return true
    }
```

- Per i dettagli sull'API, consulta la SignUpguida 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 **`UpdateUserPool`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateUserPool`.

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:

- Confermare automaticamente gli utenti noti con una funzione Lambda
- Migrare automaticamente gli utenti noti con una funzione Lambda
- Scrivere i dati di attività personalizzate con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito tramite un SDK

CLI

AWS CLI

Come aggiornare un pool di utenti

L'esempio `update-user-pool` seguente modifica un pool di utenti con una sintassi di esempio per ciascuna delle opzioni di configurazione disponibili. Per aggiornare un pool di utenti, è necessario specificare tutte le opzioni configurate in precedenza. In caso contrario, verrà ripristinato il relativo valore predefinito.

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_EXAMPLE \
    --policies PasswordPolicy=
\{MinimumLength=6,RequireUppercase=true,RequireLowercase=true,RequireNumbers=true,Require
 \
    --deletion-protection ACTIVE \
    --lambda-config PreSignUp="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-presignup-
function",PreTokenGeneration="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-pretoken-function" \
    --auto-verified-attributes "phone_number" "email" \
    --verification-message-template \{\"SmsMessage\":\""Your code is
 {####}"\",\"EmailMessage\":\""Your code is {####}"\",\"EmailSubject\":\""Your
 verification code"\",\"EmailMessageByLink\":\""Click {##here##} to verify
 your email address."\",\"EmailSubjectByLink\":\""Your verification link"\",
\"DefaultEmailOption\":\"CONFIRM_WITH_LINK\"\} \
    --sms-authentication-message "Your code is {####}" \
    --user-attribute-update-settings
 AttributesRequireVerificationBeforeUpdate="email","phone_number" \
```

```
    --mfa-configuration "OPTIONAL" \
    --device-
configuration ChallengeRequiredOnNewDevice=true,DeviceOnlyRememberedOnUserPrompt=true
 \
    --email-configuration SourceArn="arn:aws:ses:us-
west-2:123456789012:identity/admin@example.com",ReplyToEmailAddress="amdin
+noreply@example.com",EmailSendingAccount=DEVELOPER,From="admin@amazon.com",Configuration
configuration-set" \
    --sms-configuration SnsCallerArn="arn:aws:iam::123456789012:role/service-
role/SNS-SMS-Role",ExternalId="12345",SnsRegion="us-west-2" \
    --admin-create-user-config
 AllowAdminCreateUserOnly=false,InviteMessageTemplate=\{SMSMessage=\""Welcome
 {username}. Your confirmation code is {####}"\",EmailMessage=\""Welcome
 {username}. Your confirmation code is {####}"\",EmailSubject=\""Welcome to
 MyMobileGame"\"\} \
    --user-pool-tags "Function"="MyMobileGame","Developers"="Berlin" \
    --admin-create-user-config
 AllowAdminCreateUserOnly=false,InviteMessageTemplate=\{SMSMessage=\""Welcome
 {username}. Your confirmation code is {####}"\",EmailMessage=\""Welcome
 {username}. Your confirmation code is {####}"\",EmailSubject=\""Welcome to
 MyMobileGame"\"\} \
    --user-pool-add-ons AdvancedSecurityMode="AUDIT" \
    --account-recovery-setting RecoveryMechanisms=
\[\{Priority=1,Name="verified_email"\},
\{Priority=2,Name="verified_phone_number"\}\]
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Aggiornamento della configurazione del pool di utenti nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta UpdateUserPool 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"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
 trigger.
type Trigger int

const (
 PreSignUp Trigger = iota
 UserMigration
 PostAuthentication
)

type TriggerInfo struct {
 Trigger    Trigger
 HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
 is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(ctx context.Context, userPoolId
 string, triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(ctx,
 &cognitoidentityprovider.DescribeUserPoolInput{
  UserPoolId: aws.String(userPoolId),
 })
 if err != nil {
```

```
  log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
 userPoolId, err)
  return err
 }
 lambdaConfig := output.UserPool.LambdaConfig
 for _, trigger := range triggers {
  switch trigger.Trigger {
  case PreSignUp:
   lambdaConfig.PreSignUp = trigger.HandlerArn
  case UserMigration:
   lambdaConfig.UserMigration = trigger.HandlerArn
  case PostAuthentication:
   lambdaConfig.PostAuthentication = trigger.HandlerArn
  }
 }
 _, err = actor.CognitoClient.UpdateUserPool(ctx,
 &cognitoidentityprovider.UpdateUserPoolInput{
  UserPoolId:   aws.String(userPoolId),
  LambdaConfig: lambdaConfig,
 })
 if err != nil {
  log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
 }
 return err
}
```

- Per i dettagli sull'API, UpdateUserPoolconsulta 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.

```
/**
 * Connect a Lambda function to the PreSignUp trigger for a Cognito user pool
```

```
 * @param {{ region: string, userPoolId: string, handlerArn: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").UpdateUserPoolCommandOutput | null, unknown]>}
 */
export const addPreSignUpHandler = async ({
  region,
  userPoolId,
  handlerArn,
}) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({
      region,
    });

    const command = new UpdateUserPoolCommand({
      UserPoolId: userPoolId,
      LambdaConfig: {
        PreSignUp: handlerArn,
      },
    });

    const response = await cognitoClient.send(command);
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};
```

- Per i dettagli sull'API, UpdateUserPoolconsulta 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.

Utilizzare **`VerifySoftwareToken`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`VerifySoftwareToken`.

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:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.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>
    /// Verify the TOTP and register for MFA.
    /// </summary>
    /// <param name="session">The name of the session.</param>
    /// <param name="code">The MFA code.</param>
    /// <returns>The status of the software token.</returns>
    public async Task<VerifySoftwareTokenResponseType>
VerifySoftwareTokenAsync(string session, string code)
    {
        var tokenRequest = new VerifySoftwareTokenRequest
        {
            UserCode = code,
            Session = session,
        };

        var verifyResponse = await
_cognitoService.VerifySoftwareTokenAsync(tokenRequest);

        return verifyResponse.Status;
    }
```

- Per i dettagli sull'API, VerifySoftwareTokenconsulta 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.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

        Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenRequest request;
        request.SetUserCode(userCode);
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenOutcome outcome =
                client.VerifySoftwareToken(request);

        if (outcome.IsSuccess()) {
            std::cout << "Verification of the code was successful."
                      << std::endl;
            session = outcome.GetResult().GetSession();
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::VerifySoftwareToken. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Per i dettagli sull'API, VerifySoftwareTokenconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come confermare la registrazione di un autenticatore TOTP

L'esempio `verify-software-token` seguente completa la registrazione TOTP per l'utente corrente.

```
aws cognito-idp verify-software-token \
    --access-token eyJra456defEXAMPLE \
    --user-code 123456
```

Output:

```
{
    "Status": "SUCCESS"
}
```

Per ulteriori informazioni, consulta Aggiunta dell'autenticazione MFA a un pool di utenti nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta VerifySoftwareToken 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.

```
   // Verify the TOTP and register for MFA.
   public static void verifyTOTP(CognitoIdentityProviderClient
identityProviderClient, String session, String code) {
       try {
           VerifySoftwareTokenRequest tokenRequest =
VerifySoftwareTokenRequest.builder()
                   .userCode(code)
```

```
                    .session(session)
                    .build();

            VerifySoftwareTokenResponse verifyResponse =
 identityProviderClient.verifySoftwareToken(tokenRequest);
            System.out.println("The status of the token is " +
 verifyResponse.statusAsString());

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, VerifySoftwareTokenconsulta 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.

```
const verifySoftwareToken = (totp) => {
  const client = new CognitoIdentityProviderClient({});

  // The 'Session' is provided in the response to 'AssociateSoftwareToken'.
  const session = process.env.SESSION;

  if (!session) {
    throw new Error(
      "Missing a valid Session. Did you run 'admin-initiate-auth'?",
    );
  }

  const command = new VerifySoftwareTokenCommand({
    Session: session,
    UserCode: totp,
```

```
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, VerifySoftwareTokenconsulta 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.

```
// Verify the TOTP and register for MFA.
suspend fun verifyTOTP(
    sessionVal: String?,
    codeVal: String?,
) {
    val tokenRequest =
        VerifySoftwareTokenRequest {
            userCode = codeVal
            session = sessionVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val verifyResponse =
 identityProviderClient.verifySoftwareToken(tokenRequest)
        println("The status of the token is ${verifyResponse.status}")
    }
}
```

- Per i dettagli sull'API, VerifySoftwareTokenconsulta AWS SDK for Kotlin 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.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def verify_mfa(self, session, user_code):
        """
        Verify a new MFA application that is associated with a user.

        :param session: Session information returned from a previous call to
 initiate
                        authentication.
        :param user_code: A code generated by the associated MFA application.
        :return: Status that indicates whether the MFA application is verified.
        """
        try:
            response = self.cognito_idp_client.verify_software_token(
                Session=session, UserCode=user_code
```

```
            )
        except ClientError as err:
            logger.error(
                "Couldn't verify MFA. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response
```

- Per i dettagli sull'API, consulta VerifySoftwareToken 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.
        DATA(lo_result) = lo_cgp->verifysoftwaretoken(
          iv_session = iv_session
          iv_usercode = iv_user_code
        ).

        ov_status = lo_result->get_status( ).

        IF ov_status = 'SUCCESS'.
          MESSAGE 'MFA token verified successfully.' TYPE 'I'.
        ELSE.
          MESSAGE |MFA verification status: { ov_status }.| TYPE 'I'.
        ENDIF.
```

```
      CATCH /aws1/cx_cgpcodemismatchex INTO DATA(lo_code_ex).
        MESSAGE 'Invalid MFA code provided.' TYPE 'E'.

      CATCH /aws1/cx_cgpenbsoftwaretokmf00 INTO DATA(lo_enabled_ex).
        MESSAGE 'Software token MFA is already enabled.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, VerifySoftwareTokenconsulta 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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Confirm that the user's TOTP authenticator is configured correctly by
    /// sending a code to it to check that it matches successfully.
    ///
    /// - Parameters:
    ///   - cipClient: The `CongnitoIdentityProviderClient` to use.
    ///   - session: An authentication session previously returned by an
    ///     `associateSoftwareToken()` call.
    ///   - mfaCode: The 6-digit code currently displayed by the user's
    ///     authenticator, as provided by the user.
    func verifyTOTP(cipClient: CognitoIdentityProviderClient, session: String?,
 mfaCode: String?) async {
        do {
            let output = try await cipClient.verifySoftwareToken(
                input: VerifySoftwareTokenInput(
                    session: session,
                    userCode: mfaCode
                )
```

```
            )

            guard let tokenStatus = output.status else {
                print("*** Unable to get the token's status.")
                return
            }
            print("=====> The token's status is: \(tokenStatus)")
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return
        } catch _ as CodeMismatchException {
            print("*** The specified MFA code doesn't match the expected value.")
            return
        } catch _ as UserNotFoundException {
            print("*** The specified username doesn't exist.")
            return
        } catch _ as UserNotConfirmedException {
            print("*** The user has not been confirmed.")
            return
        } catch {
            print("*** Error verifying the MFA token!")
            return
        }
    }
```

- Per i dettagli sull'API, consulta la VerifySoftwareTokenguida 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.

Scenari per l'utilizzo di Amazon Cognito Identity Provider AWS SDK

I seguenti esempi di codice mostrano come implementare scenari comuni in Amazon Cognito Identity Provider con AWS SDK. Questi scenari illustrano come eseguire attività specifiche chiamando più funzioni all'interno del gestore dell'identità per Amazon Cognito o in combinazione 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

- Conferma automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda utilizzando un AWS SDK
- Esegui automaticamente la migrazione di utenti Amazon Cognito noti con una funzione Lambda utilizzando un AWS SDK
- Ottieni token client M2M e scopri i termini di Amazon Cognito utilizzando un AWS SDK
- Nozioni di base sui bacini d'utenza
- Registra un utente con un pool di utenti Amazon Cognito che richiede l'autenticazione a più fattori utilizzando un AWS SDK
- Usa i pool di identità e i flussi di autenticazione di Amazon Cognito
- Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito utilizzando un AWS SDK

Conferma automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda utilizzando un AWS SDK

L'esempio di codice seguente mostra come confermare automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda.

- Configura un pool di utenti per chiamare una funzione Lambda per il trigger `PreSignUp`.
- Registra un utente con Amazon Cognito.
- La funzione Lambda esegue la scansione di una tabella DynamoDB e conferma automaticamente gli utenti noti.
- Accedi come nuovo utente, quindi elimina le risorse.

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.

Esegui uno scenario interattivo al prompt dei comandi.

```
import (
 "context"
 "errors"
 "log"
 "strings"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// AutoConfirm separates the steps of this scenario into individual functions so
 that
// they are simpler to read and understand.
type AutoConfirm struct {
 helper       IScenarioHelper
 questioner   demotools.IQuestioner
 resources    Resources
 cognitoActor *actions.CognitoActions
}

// NewAutoConfirm constructs a new auto confirm runner.
func NewAutoConfirm(sdkConfig aws.Config, questioner demotools.IQuestioner,
 helper IScenarioHelper) AutoConfirm {
 scenario := AutoConfirm{
  helper:       helper,
  questioner:   questioner,
  resources:    Resources{},
```

```
  cognitoActor: &actions.CognitoActions{CognitoClient:
 cognitoidentityprovider.NewFromConfig(sdkConfig)},
 }
 scenario.resources.init(scenario.cognitoActor, questioner)
 return scenario
}

// AddPreSignUpTrigger adds a Lambda handler as an invocation target for the
 PreSignUp trigger.
func (runner *AutoConfirm) AddPreSignUpTrigger(ctx context.Context, userPoolId
 string, functionArn string) {
 log.Printf("Let's add a Lambda function to handle the PreSignUp trigger from
 Cognito.\n" +
  "This trigger happens when a user signs up, and lets your function take action
 before the main Cognito\n" +
  "sign up processing occurs.\n")
 err := runner.cognitoActor.UpdateTriggers(
  ctx, userPoolId,
  actions.TriggerInfo{Trigger: actions.PreSignUp, HandlerArn:
 aws.String(functionArn)})
 if err != nil {
  panic(err)
 }
 log.Printf("Lambda function %v added to user pool %v to handle the PreSignUp
 trigger.\n",
  functionArn, userPoolId)
}

// SignUpUser signs up a user from the known user table with a password you
 specify.
func (runner *AutoConfirm) SignUpUser(ctx context.Context, clientId string,
 usersTable string) (string, string) {
 log.Println("Let's sign up a user to your Cognito user pool. When the user's
 email matches an email in the\n" +
  "DynamoDB known users table, it is automatically verified and the user is
 confirmed.")

 knownUsers, err := runner.helper.GetKnownUsers(ctx, usersTable)
 if err != nil {
  panic(err)
 }
 userChoice := runner.questioner.AskChoice("Which user do you want to use?\n",
 knownUsers.UserNameList())
 user := knownUsers.Users[userChoice]
```

```
 var signedUp bool
 var userConfirmed bool
 password := runner.questioner.AskPassword("Enter a password that has at least
 eight characters, uppercase, lowercase, numbers and symbols.\n"+
  "(the password will not display as you type):", 8)
 for !signedUp {
  log.Printf("Signing up user '%v' with email '%v' to Cognito.\n", user.UserName,
 user.UserEmail)
  userConfirmed, err = runner.cognitoActor.SignUp(ctx, clientId, user.UserName,
 password, user.UserEmail)
  if err != nil {
   var invalidPassword *types.InvalidPasswordException
   if errors.As(err, &invalidPassword) {
    password = runner.questioner.AskPassword("Enter another password:", 8)
   } else {
    panic(err)
   }
  } else {
   signedUp = true
  }
 }
 log.Printf("User %v signed up, confirmed = %v.\n", user.UserName, userConfirmed)

 log.Println(strings.Repeat("-", 88))

 return user.UserName, password
}

// SignInUser signs in a user.
func (runner *AutoConfirm) SignInUser(ctx context.Context, clientId string,
 userName string, password string) string {
 runner.questioner.Ask("Press Enter when you're ready to continue.")
 log.Printf("Let's sign in as %v...\n", userName)
 authResult, err := runner.cognitoActor.SignIn(ctx, clientId, userName, password)
 if err != nil {
  panic(err)
 }
 log.Printf("Successfully signed in. Your access token starts with: %v...\n",
 (*authResult.AccessToken)[:10])
 log.Println(strings.Repeat("-", 88))
 return *authResult.AccessToken
}
```

```
// Run runs the scenario.
func (runner *AutoConfirm) Run(ctx context.Context, stackName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   runner.resources.Cleanup(ctx)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Printf("Welcome\n")

 log.Println(strings.Repeat("-", 88))

 stackOutputs, err := runner.helper.GetStackOutputs(ctx, stackName)
 if err != nil {
  panic(err)
 }
 runner.resources.userPoolId = stackOutputs["UserPoolId"]
 runner.helper.PopulateUserTable(ctx, stackOutputs["TableName"])

 runner.AddPreSignUpTrigger(ctx, stackOutputs["UserPoolId"],
 stackOutputs["AutoConfirmFunctionArn"])
 runner.resources.triggers = append(runner.resources.triggers, actions.PreSignUp)
 userName, password := runner.SignUpUser(ctx, stackOutputs["UserPoolClientId"],
 stackOutputs["TableName"])
 runner.helper.ListRecentLogEvents(ctx, stackOutputs["AutoConfirmFunction"])
 runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
  runner.SignInUser(ctx, stackOutputs["UserPoolClientId"], userName, password))

 runner.resources.Cleanup(ctx)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Gestisci il trigger `PreSignUp` con una funzione Lambda.

```
import (
```

```
 "context"
 "log"
 "os"

 "github.com/aws/aws-lambda-go/events"
 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 dynamodbtypes "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

const TABLE_NAME = "TABLE_NAME"

// UserInfo defines structured user data that can be marshalled to a DynamoDB
 format.
type UserInfo struct {
 UserName  string `dynamodbav:"UserName"`
 UserEmail string `dynamodbav:"UserEmail"`
}

// GetKey marshals the user email value to a DynamoDB key format.
func (user UserInfo) GetKey() map[string]dynamodbtypes.AttributeValue {
 userEmail, err := attributevalue.Marshal(user.UserEmail)
 if err != nil {
  panic(err)
 }
 return map[string]dynamodbtypes.AttributeValue{"UserEmail": userEmail}
}

type handler struct {
 dynamoClient *dynamodb.Client
}

// HandleRequest handles the PreSignUp event by looking up a user in an Amazon
 DynamoDB table and
// specifying whether they should be confirmed and verified.
func (h *handler) HandleRequest(ctx context.Context, event
 events.CognitoEventUserPoolsPreSignup) (events.CognitoEventUserPoolsPreSignup,
 error) {
 log.Printf("Received presignup from %v for user '%v'", event.TriggerSource,
 event.UserName)
 if event.TriggerSource != "PreSignUp_SignUp" {
```

```
  // Other trigger sources, such as PreSignUp_AdminInitiateAuth, ignore the
 response from this handler.
  return event, nil
 }
 tableName := os.Getenv(TABLE_NAME)
 user := UserInfo{
  UserEmail: event.Request.UserAttributes["email"],
 }
 log.Printf("Looking up email %v in table %v.\n", user.UserEmail, tableName)
 output, err := h.dynamoClient.GetItem(ctx, &dynamodb.GetItemInput{
  Key:       user.GetKey(),
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Error looking up email %v.\n", user.UserEmail)
  return event, err
 }
 if output.Item == nil {
  log.Printf("Email %v not found. Email verification is required.\n",
 user.UserEmail)
  return event, err
 }

 err = attributevalue.UnmarshalMap(output.Item, &user)
 if err != nil {
  log.Printf("Couldn't unmarshal DynamoDB item. Here's why: %v\n", err)
  return event, err
 }

 if user.UserName != event.UserName {
  log.Printf("UserEmail %v found, but stored UserName '%v' does not match
 supplied UserName '%v'. Verification is required.\n",
   user.UserEmail, user.UserName, event.UserName)
 } else {
  log.Printf("UserEmail %v found with matching UserName %v. User is confirmed.
\n", user.UserEmail, user.UserName)
  event.Response.AutoConfirmUser = true
  event.Response.AutoVerifyEmail = true
 }

 return event, err
}

func main() {
```

```
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 if err != nil {
  log.Panicln(err)
 }
 h := handler{
  dynamoClient: dynamodb.NewFromConfig(sdkConfig),
 }
 lambda.Start(h.HandleRequest)
}
```

Crea una struttura che esegue operazioni comuni.

```
import (
 "context"
 "log"
 "strings"
 "time"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// IScenarioHelper defines common functions used by the workflows in this
 example.
type IScenarioHelper interface {
 Pause(secs int)
 GetStackOutputs(ctx context.Context, stackName string) (actions.StackOutputs,
 error)
 PopulateUserTable(ctx context.Context, tableName string)
 GetKnownUsers(ctx context.Context, tableName string) (actions.UserList, error)
 AddKnownUser(ctx context.Context, tableName string, user actions.User)
 ListRecentLogEvents(ctx context.Context, functionName string)
}
```

```
// ScenarioHelper contains AWS wrapper structs used by the workflows in this
 example.
type ScenarioHelper struct {
 questioner  demotools.IQuestioner
 dynamoActor *actions.DynamoActions
 cfnActor    *actions.CloudFormationActions
 cwlActor    *actions.CloudWatchLogsActions
 isTestRun   bool
}

// NewScenarioHelper constructs a new scenario helper.
func NewScenarioHelper(sdkConfig aws.Config, questioner demotools.IQuestioner)
 ScenarioHelper {
 scenario := ScenarioHelper{
  questioner:  questioner,
  dynamoActor: &actions.DynamoActions{DynamoClient:
 dynamodb.NewFromConfig(sdkConfig)},
  cfnActor:    &actions.CloudFormationActions{CfnClient:
 cloudformation.NewFromConfig(sdkConfig)},
  cwlActor:    &actions.CloudWatchLogsActions{CwlClient:
 cloudwatchlogs.NewFromConfig(sdkConfig)},
 }
 return scenario
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
 if !helper.isTestRun {
  time.Sleep(time.Duration(secs) * time.Second)
 }
}

// GetStackOutputs gets the outputs from the specified CloudFormation stack in a
 structured format.
func (helper ScenarioHelper) GetStackOutputs(ctx context.Context, stackName
 string) (actions.StackOutputs, error) {
 return helper.cfnActor.GetOutputs(ctx, stackName), nil
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(ctx context.Context, tableName
 string) {
 log.Printf("First, let's add some users to the DynamoDB %v table we'll use for
 this example.\n", tableName)
```

```
 err := helper.dynamoActor.PopulateTable(ctx, tableName)
 if err != nil {
  panic(err)
 }
}

// GetKnownUsers gets the users from the known users table in a structured
 format.
func (helper ScenarioHelper) GetKnownUsers(ctx context.Context, tableName string)
 (actions.UserList, error) {
 knownUsers, err := helper.dynamoActor.Scan(ctx, tableName)
 if err != nil {
  log.Printf("Couldn't get known users from table %v. Here's why: %v\n",
 tableName, err)
 }
 return knownUsers, err
}

// AddKnownUser adds a user to the known users table.
func (helper ScenarioHelper) AddKnownUser(ctx context.Context, tableName string,
 user actions.User) {
 log.Printf("Adding user '%v' with email '%v' to the DynamoDB known users
 table...\n",
  user.UserName, user.UserEmail)
 err := helper.dynamoActor.AddUser(ctx, tableName, user)
 if err != nil {
  panic(err)
 }
}

// ListRecentLogEvents gets the most recent log stream and events for the
 specified Lambda function and displays them.
func (helper ScenarioHelper) ListRecentLogEvents(ctx context.Context,
 functionName string) {
 log.Println("Waiting a few seconds to let Lambda write to CloudWatch Logs...")
 helper.Pause(10)
 log.Println("Okay, let's check the logs to find what's happened recently with
 your Lambda function.")
 logStream, err := helper.cwlActor.GetLatestLogStream(ctx, functionName)
 if err != nil {
  panic(err)
 }
 log.Printf("Getting some recent events from log stream %v\n",
 *logStream.LogStreamName)
```

```
 events, err := helper.cwlActor.GetLogEvents(ctx, functionName,
 *logStream.LogStreamName, 10)
 if err != nil {
  panic(err)
 }
 for _, event := range events {
  log.Printf("\t%v", *event.Message)
 }
 log.Println(strings.Repeat("-", 88))
}
```

Crea una struttura che racchiude le azioni di Amazon Cognito.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
 trigger.
type Trigger int

const (
 PreSignUp Trigger = iota
 UserMigration
 PostAuthentication
)

type TriggerInfo struct {
```

```
 Trigger    Trigger
 HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
 is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(ctx context.Context, userPoolId
 string, triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(ctx,
 &cognitoidentityprovider.DescribeUserPoolInput{
  UserPoolId: aws.String(userPoolId),
 })
 if err != nil {
  log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
 userPoolId, err)
  return err
 }
 lambdaConfig := output.UserPool.LambdaConfig
 for _, trigger := range triggers {
  switch trigger.Trigger {
  case PreSignUp:
   lambdaConfig.PreSignUp = trigger.HandlerArn
  case UserMigration:
   lambdaConfig.UserMigration = trigger.HandlerArn
  case PostAuthentication:
   lambdaConfig.PostAuthentication = trigger.HandlerArn
  }
 }
 _, err = actor.CognitoClient.UpdateUserPool(ctx,
 &cognitoidentityprovider.UpdateUserPoolInput{
  UserPoolId:   aws.String(userPoolId),
  LambdaConfig: lambdaConfig,
 })
 if err != nil {
  log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
 }
 return err
}



// SignUp signs up a user with Amazon Cognito.
```

```
func (actor CognitoActions) SignUp(ctx context.Context, clientId string, userName
 string, password string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(ctx,
 &cognitoidentityprovider.SignUpInput{
  ClientId: aws.String(clientId),
  Password: aws.String(password),
  Username: aws.String(userName),
  UserAttributes: []types.AttributeType{
   {Name: aws.String("email"), Value: aws.String(userEmail)},
  },
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
  }
 } else {
  confirmed = output.UserConfirmed
 }
 return confirmed, err
}



// SignIn signs in a user to Amazon Cognito using a username and password
 authentication flow.
func (actor CognitoActions) SignIn(ctx context.Context, clientId string, userName
 string, password string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(ctx,
 &cognitoidentityprovider.InitiateAuthInput{
  AuthFlow:       "USER_PASSWORD_AUTH",
  ClientId:       aws.String(clientId),
  AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
 })
 if err != nil {
  var resetRequired *types.PasswordResetRequiredException
  if errors.As(err, &resetRequired) {
   log.Println(*resetRequired.Message)
  } else {
   log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
```

```
  }
 } else {
  authResult = output.AuthenticationResult
 }
 return authResult, err
}



// ForgotPassword starts a password recovery flow for a user. This flow typically
 sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(ctx context.Context, clientId string,
 userName string) (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(ctx,
 &cognitoidentityprovider.ForgotPasswordInput{
  ClientId: aws.String(clientId),
  Username: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't start password reset for user '%v'. Here;s why: %v\n",
 userName, err)
 }
 return output.CodeDeliveryDetails, err
}



// ConfirmForgotPassword confirms a user with a confirmation code and a new
 password.
func (actor CognitoActions) ConfirmForgotPassword(ctx context.Context, clientId
 string, code string, userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(ctx,
 &cognitoidentityprovider.ConfirmForgotPasswordInput{
  ClientId:         aws.String(clientId),
  ConfirmationCode: aws.String(code),
  Password:         aws.String(password),
  Username:         aws.String(userName),
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
```

```
   log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
  }
 }
 return err
}



// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(ctx context.Context, userAccessToken
 string) error {
 _, err := actor.CognitoClient.DeleteUser(ctx,
 &cognitoidentityprovider.DeleteUserInput{
  AccessToken: aws.String(userAccessToken),
 })
 if err != nil {
  log.Printf("Couldn't delete user. Here's why: %v\n", err)
 }
 return err
}



// AdminCreateUser uses administrator credentials to add a user to a user pool.
 This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(ctx context.Context, userPoolId
 string, userName string, userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(ctx,
 &cognitoidentityprovider.AdminCreateUserInput{
  UserPoolId:     aws.String(userPoolId),
  Username:       aws.String(userName),
  MessageAction:  types.MessageActionTypeSuppress,
  UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
 aws.String(userEmail)}},
 })
 if err != nil {
  var userExists *types.UsernameExistsException
  if errors.As(err, &userExists) {
   log.Printf("User %v already exists in the user pool.", userName)
   err = nil
  } else {
   log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
  }
```

```
 }
 return err
}



// AdminSetUserPassword uses administrator credentials to set a password for a
 user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(ctx context.Context, userPoolId
 string, userName string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(ctx,
 &cognitoidentityprovider.AdminSetUserPasswordInput{
  Password:   aws.String(password),
  UserPoolId: aws.String(userPoolId),
  Username:   aws.String(userName),
  Permanent:  true,
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
 err)
  }
 }
 return err
}
```

Crea una struttura che racchiude le azioni di DynamoDB.

```
import (
 "context"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
```

```
 "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

// DynamoActions encapsulates the Amazon Simple Notification Service (Amazon SNS)
 actions
// used in the examples.
type DynamoActions struct {
 DynamoClient *dynamodb.Client
}

// User defines structured user data.
type User struct {
 UserName  string
 UserEmail string
 LastLogin *LoginInfo `dynamodbav:",omitempty"`
}

// LoginInfo defines structured custom login data.
type LoginInfo struct {
 UserPoolId string
 ClientId   string
 Time       string
}

// UserList defines a list of users.
type UserList struct {
 Users []User
}

// UserNameList returns the usernames contained in a UserList as a list of
 strings.
func (users *UserList) UserNameList() []string {
 names := make([]string, len(users.Users))
 for i := 0; i < len(users.Users); i++ {
  names[i] = users.Users[i].UserName
 }
 return names
}

// PopulateTable adds a set of test users to the table.
func (actor DynamoActions) PopulateTable(ctx context.Context, tableName string)
 error {
 var err error
 var item map[string]types.AttributeValue
```

```
 var writeReqs []types.WriteRequest
 for i := 1; i < 4; i++ {
  item, err = attributevalue.MarshalMap(User{UserName: fmt.Sprintf("test_user_
%v", i), UserEmail: fmt.Sprintf("test_email_%v@example.com", i)})
  if err != nil {
   log.Printf("Couldn't marshall user into DynamoDB format. Here's why: %v\n",
 err)
   return err
  }
  writeReqs = append(writeReqs, types.WriteRequest{PutRequest:
 &types.PutRequest{Item: item}})
 }
 _, err = actor.DynamoClient.BatchWriteItem(ctx, &dynamodb.BatchWriteItemInput{
  RequestItems: map[string][]types.WriteRequest{tableName: writeReqs},
 })
 if err != nil {
  log.Printf("Couldn't populate table %v with users. Here's why: %v\n",
 tableName, err)
 }
 return err
}

// Scan scans the table for all items.
func (actor DynamoActions) Scan(ctx context.Context, tableName string) (UserList,
 error) {
 var userList UserList
 output, err := actor.DynamoClient.Scan(ctx, &dynamodb.ScanInput{
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't scan table %v for items. Here's why: %v\n", tableName,
 err)
 } else {
  err = attributevalue.UnmarshalListOfMaps(output.Items, &userList.Users)
  if err != nil {
   log.Printf("Couldn't unmarshal items into users. Here's why: %v\n", err)
  }
 }
 return userList, err
}

// AddUser adds a user item to a table.
func (actor DynamoActions) AddUser(ctx context.Context, tableName string, user
 User) error {
```

```
 userItem, err := attributevalue.MarshalMap(user)
 if err != nil {
  log.Printf("Couldn't marshall user to item. Here's why: %v\n", err)
 }
 _, err = actor.DynamoClient.PutItem(ctx, &dynamodb.PutItemInput{
  Item:      userItem,
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't put item in table %v. Here's why: %v", tableName, err)
 }
 return err
}
```

Crea una struttura che racchiuda le azioni di CloudWatch Logs.

```
import (
 "context"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs/types"
)

type CloudWatchLogsActions struct {
 CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(ctx context.Context,
 functionName string) (types.LogStream, error) {
 var logStream types.LogStream
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.DescribeLogStreams(ctx,
 &cloudwatchlogs.DescribeLogStreamsInput{
  Descending:   aws.Bool(true),
  Limit:        aws.Int32(1),
  LogGroupName: aws.String(logGroupName),
```

```
  OrderBy:      types.OrderByLastEventTime,
 })
 if err != nil {
  log.Printf("Couldn't get log streams for log group %v. Here's why: %v\n",
 logGroupName, err)
 } else {
  logStream = output.LogStreams[0]
 }
 return logStream, err
}

// GetLogEvents gets the most recent eventCount events from the specified log
 stream.
func (actor CloudWatchLogsActions) GetLogEvents(ctx context.Context, functionName
 string, logStreamName string, eventCount int32) (
 []types.OutputLogEvent, error) {
 var events []types.OutputLogEvent
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.GetLogEvents(ctx,
 &cloudwatchlogs.GetLogEventsInput{
  LogStreamName: aws.String(logStreamName),
  Limit:         aws.Int32(eventCount),
  LogGroupName:  aws.String(logGroupName),
 })
 if err != nil {
  log.Printf("Couldn't get log event for log stream %v. Here's why: %v\n",
 logStreamName, err)
 } else {
  events = output.Events
 }
 return events, err
}
```

Crea una struttura che racchiuda le azioni. CloudFormation

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
```

```
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
)

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
 CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
 structured format.
func (actor CloudFormationActions) GetOutputs(ctx context.Context, stackName
 string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(ctx,
 &cloudformation.DescribeStacksInput{
  StackName: aws.String(stackName),
 })
 if err != nil || len(output.Stacks) == 0 {
  log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
 stackName, err)
 }
 stackOutputs := StackOutputs{}
 for _, out := range output.Stacks[0].Outputs {
  stackOutputs[*out.OutputKey] = *out.OutputValue
 }
 return stackOutputs
}
```

Eliminare le risorse.

```
import (
 "context"
 "log"
 "user_pools_and_lambda_triggers/actions"

 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// Resources keeps track of AWS resources created during an example and handles
```

```
// cleanup when the example finishes.
type Resources struct {
 userPoolId       string
 userAccessTokens []string
 triggers         []actions.Trigger

 cognitoActor *actions.CognitoActions
 questioner   demotools.IQuestioner
}

func (resources *Resources) init(cognitoActor *actions.CognitoActions, questioner
 demotools.IQuestioner) {
 resources.userAccessTokens = []string{}
 resources.triggers = []actions.Trigger{}
 resources.cognitoActor = cognitoActor
 resources.questioner = questioner
}

// Cleanup deletes all AWS resources created during an example.
func (resources *Resources) Cleanup(ctx context.Context) {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong during cleanup.\n%v\n", r)
   log.Println("Use the AWS Management Console to remove any remaining resources
 \n" +
    "that were created for this scenario.")
  }
 }()

 wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
 resources that were created "+
  "during this demo (y/n)?", "y")
 if wantDelete {
  for _, accessToken := range resources.userAccessTokens {
   err := resources.cognitoActor.DeleteUser(ctx, accessToken)
   if err != nil {
    log.Println("Couldn't delete user during cleanup.")
    panic(err)
   }
   log.Println("Deleted user.")
  }
  triggerList := make([]actions.TriggerInfo, len(resources.triggers))
  for i := 0; i < len(resources.triggers); i++ {
```

```
    triggerList[i] = actions.TriggerInfo{Trigger: resources.triggers[i],
  HandlerArn: nil}
   }
   err := resources.cognitoActor.UpdateTriggers(ctx, resources.userPoolId,
  triggerList...)
   if err != nil {
    log.Println("Couldn't update Cognito triggers during cleanup.")
    panic(err)
   }
   log.Println("Removed Cognito triggers from user pool.")
  } else {
   log.Println("Be sure to remove resources when you're done with them to avoid
  unexpected charges!")
  }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Go .
  - DeleteUser
  - InitiateAuth
  - SignUp
  - UpdateUserPool

JavaScript

SDK per (v3 JavaScript)

> **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.

Configura un'esecuzione interattiva dello "Scenario". Gli esempi JavaScript (v3) condividono uno Scenario runner per semplificare esempi complessi. Il codice sorgente completo è attivo. GitHub

```
import { AutoConfirm } from "./scenario-auto-confirm.js";

/**
 * The context is passed to every scenario. Scenario steps
 * will modify the context.
 */
const context = {
  errors: [],
  users: [
    {
      UserName: "test_user_1",
      UserEmail: "test_email_1@example.com",
    },
    {
      UserName: "test_user_2",
      UserEmail: "test_email_2@example.com",
    },
    {
      UserName: "test_user_3",
      UserEmail: "test_email_3@example.com",
    },
  ],
};

/**
 * Three Scenarios are created for the workflow. A Scenario is an orchestration
 class
 * that simplifies running a series of steps.
 */
export const scenarios = {
  // Demonstrate automatically confirming known users in a database.
  "auto-confirm": AutoConfirm(context),
};

// Call function if run directly
import { fileURLToPath } from "node:url";
import { parseScenarioArgs } from "@aws-doc-sdk-examples/lib/scenario/index.js";

if (process.argv[1] === fileURLToPath(import.meta.url)) {
  parseScenarioArgs(scenarios, {
    name: "Cognito user pools and triggers",
    description:
```

```
      "Demonstrate how to use the AWS SDKs to customize Amazon Cognito
 authentication behavior.",
  });
}
```

Questo scenario dimostra la conferma automatica di un utente noto. Orchestra i passaggi di esempio.

```
import { wait } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import {
  Scenario,
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
} from "@aws-doc-sdk-examples/lib/scenario/scenario.js";

import {
  getStackOutputs,
  logCleanUpReminder,
  promptForStackName,
  promptForStackRegion,
  skipWhenErrors,
} from "./steps-common.js";
import { populateTable } from "./actions/dynamodb-actions.js";
import {
  addPreSignUpHandler,
  deleteUser,
  getUser,
  signIn,
  signUpUser,
} from "./actions/cognito-actions.js";
import {
  getLatestLogStreamForLambda,
  getLogEvents,
} from "./actions/cloudwatch-logs-actions.js";

/**
 * @typedef {{
 *   errors: Error[],
 *   password: string,
 *   users: { UserName: string, UserEmail: string }[],
 *   selectedUser?: string,
```

```
 *   stackName?: string,
 *   stackRegion?: string,
 *   token?: string,
 *   confirmDeleteSignedInUser?: boolean,
 *   TableName?: string,
 *   UserPoolClientId?: string,
 *   UserPoolId?: string,
 *   UserPoolArn?: string,
 *   AutoConfirmHandlerArn?: string,
 *   AutoConfirmHandlerName?: string
 * }} State
 */

const greeting = new ScenarioOutput(
  "greeting",
  (/** @type {State} */ state) => `This demo will populate some users into the \
database created as part of the "${state.stackName}" stack. \
Then the AutoConfirmHandler will be linked to the PreSignUp \
trigger from Cognito. Finally, you will choose a user to sign up.`,
  { skipWhen: skipWhenErrors },
);

const logPopulatingUsers = new ScenarioOutput(
  "logPopulatingUsers",
  "Populating the DynamoDB table with some users.",
  { skipWhenErrors: skipWhenErrors },
);

const logPopulatingUsersComplete = new ScenarioOutput(
  "logPopulatingUsersComplete",
  "Done populating users.",
  { skipWhen: skipWhenErrors },
);

const populateUsers = new ScenarioAction(
  "populateUsers",
  async (/** @type {State} */ state) => {
    const [_, err] = await populateTable({
      region: state.stackRegion,
      tableName: state.TableName,
      items: state.users,
    });
    if (err) {
      state.errors.push(err);
```

```
    }
  },
  {
    skipWhen: skipWhenErrors,
  },
);

const logSetupSignUpTrigger = new ScenarioOutput(
  "logSetupSignUpTrigger",
  "Setting up the PreSignUp trigger for the Cognito User Pool.",
  { skipWhen: skipWhenErrors },
);

const setupSignUpTrigger = new ScenarioAction(
  "setupSignUpTrigger",
  async (/** @type {State} */ state) => {
    const [_, err] = await addPreSignUpHandler({
      region: state.stackRegion,
      userPoolId: state.UserPoolId,
      handlerArn: state.AutoConfirmHandlerArn,
    });
    if (err) {
      state.errors.push(err);
    }
  },
  {
    skipWhen: skipWhenErrors,
  },
);

const logSetupSignUpTriggerComplete = new ScenarioOutput(
  "logSetupSignUpTriggerComplete",
  (
    /** @type {State} */ state,
  ) => `The lambda function "${state.AutoConfirmHandlerName}" \
has been configured as the PreSignUp trigger handler for the user pool
 "${state.UserPoolId}".`,
  { skipWhen: skipWhenErrors },
);

const selectUser = new ScenarioInput(
  "selectedUser",
  "Select a user to sign up.",
  {
```

```
    type: "select",
    choices: (/** @type {State} */ state) => state.users.map((u) => u.UserName),
    skipWhen: skipWhenErrors,
    default: (/** @type {State} */ state) => state.users[0].UserName,
  },
);

const checkIfUserAlreadyExists = new ScenarioAction(
  "checkIfUserAlreadyExists",
  async (/** @type {State} */ state) => {
    const [user, err] = await getUser({
      region: state.stackRegion,
      userPoolId: state.UserPoolId,
      username: state.selectedUser,
    });

    if (err?.name === "UserNotFoundException") {
      // Do nothing. We're not expecting the user to exist before
      // sign up is complete.
      return;
    }

    if (err) {
      state.errors.push(err);
      return;
    }

    if (user) {
      state.errors.push(
        new Error(
          `The user "${state.selectedUser}" already exists in the user pool
 "${state.UserPoolId}".`,
        ),
      );
    }
  },
  {
    skipWhen: skipWhenErrors,
  },
);

const createPassword = new ScenarioInput(
  "password",
```

```
  "Enter a password that has at least eight characters, uppercase, lowercase,
 numbers and symbols.",
  { type: "password", skipWhen: skipWhenErrors, default: "Abcd1234!" },
);

const logSignUpExistingUser = new ScenarioOutput(
  "logSignUpExistingUser",
  (/** @type {State} */ state) => `Signing up user "${state.selectedUser}".`,
  { skipWhen: skipWhenErrors },
);

const signUpExistingUser = new ScenarioAction(
  "signUpExistingUser",
  async (/** @type {State} */ state) => {
    const signUp = (password) =>
      signUpUser({
        region: state.stackRegion,
        userPoolClientId: state.UserPoolClientId,
        username: state.selectedUser,
        email: state.users.find((u) => u.UserName === state.selectedUser)
          .UserEmail,
        password,
      });

    let [_, err] = await signUp(state.password);

    while (err?.name === "InvalidPasswordException") {
      console.warn("The password you entered was invalid.");
      await createPassword.handle(state);
      [_, err] = await signUp(state.password);
    }

    if (err) {
      state.errors.push(err);
    }
  },
  { skipWhen: skipWhenErrors },
);

const logSignUpExistingUserComplete = new ScenarioOutput(
  "logSignUpExistingUserComplete",
  (/** @type {State} */ state) =>
    `"${state.selectedUser} was signed up successfully.`,
  { skipWhen: skipWhenErrors },
```

```
);

const logLambdaLogs = new ScenarioAction(
  "logLambdaLogs",
  async (/** @type {State} */ state) => {
    console.log(
      "Waiting a few seconds to let Lambda write to CloudWatch Logs...\n",
    );
    await wait(10);

    const [logStream, logStreamErr] = await getLatestLogStreamForLambda({
      functionName: state.AutoConfirmHandlerName,
      region: state.stackRegion,
    });
    if (logStreamErr) {
      state.errors.push(logStreamErr);
      return;
    }

    console.log(
      `Getting some recent events from log stream "${logStream.logStreamName}"`,
    );
    const [logEvents, logEventsErr] = await getLogEvents({
      functionName: state.AutoConfirmHandlerName,
      region: state.stackRegion,
      eventCount: 10,
      logStreamName: logStream.logStreamName,
    });
    if (logEventsErr) {
      state.errors.push(logEventsErr);
      return;
    }

    console.log(logEvents.map((ev) => `\t${ev.message}`).join(""));
  },
  { skipWhen: skipWhenErrors },
);

const logSignInUser = new ScenarioOutput(
  "logSignInUser",
  (/** @type {State} */ state) => `Let's sign in as ${state.selectedUser}`,
  { skipWhen: skipWhenErrors },
);
```

```
const signInUser = new ScenarioAction(
  "signInUser",
  async (/** @type {State} */ state) => {
    const [response, err] = await signIn({
      region: state.stackRegion,
      clientId: state.UserPoolClientId,
      username: state.selectedUser,
      password: state.password,
    });

    if (err?.name === "PasswordResetRequiredException") {
      state.errors.push(new Error("Please reset your password."));
      return;
    }

    if (err) {
      state.errors.push(err);
      return;
    }

    state.token = response?.AuthenticationResult?.AccessToken;
  },
  { skipWhen: skipWhenErrors },
);

const logSignInUserComplete = new ScenarioOutput(
  "logSignInUserComplete",
  (/** @type {State} */ state) =>
    `Successfully signed in. Your access token starts with:
 ${state.token.slice(0, 11)}`,
  { skipWhen: skipWhenErrors },
);

const confirmDeleteSignedInUser = new ScenarioInput(
  "confirmDeleteSignedInUser",
  "Do you want to delete the currently signed in user?",
  { type: "confirm", skipWhen: skipWhenErrors },
);

const deleteSignedInUser = new ScenarioAction(
  "deleteSignedInUser",
  async (/** @type {State} */ state) => {
    const [_, err] = await deleteUser({
      region: state.stackRegion,
```

```
      accessToken: state.token,
    });

    if (err) {
      state.errors.push(err);
    }
  },
  {
    skipWhen: (/** @type {State} */ state) =>
      skipWhenErrors(state) || !state.confirmDeleteSignedInUser,
  },
);

const logErrors = new ScenarioOutput(
  "logErrors",
  (/** @type {State}*/ state) => {
    const errorList = state.errors
      .map((err) => ` - ${err.name}: ${err.message}`)
      .join("\n");
    return `Scenario errors found:\n${errorList}`;
  },
  {
    // Don't log errors when there aren't any!
    skipWhen: (/** @type {State} */ state) => state.errors.length === 0,
  },
);

export const AutoConfirm = (context) =>
  new Scenario(
    "AutoConfirm",
    [
      promptForStackName,
      promptForStackRegion,
      getStackOutputs,
      greeting,
      logPopulatingUsers,
      populateUsers,
      logPopulatingUsersComplete,
      logSetupSignUpTrigger,
      setupSignUpTrigger,
      logSetupSignUpTriggerComplete,
      selectUser,
      checkIfUserAlreadyExists,
      createPassword,
```

```
      logSignUpExistingUser,
      signUpExistingUser,
      logSignUpExistingUserComplete,
      logLambdaLogs,
      logSignInUser,
      signInUser,
      logSignInUserComplete,
      confirmDeleteSignedInUser,
      deleteSignedInUser,
      logCleanUpReminder,
      logErrors,
    ],
    context,
  );
```

Questi sono passaggi condivisi con altri scenari.

```
import {
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
} from "@aws-doc-sdk-examples/lib/scenario/scenario.js";
import { getCfnOutputs } from "@aws-doc-sdk-examples/lib/sdk/cfn-outputs.js";

export const skipWhenErrors = (state) => state.errors.length > 0;

export const getStackOutputs = new ScenarioAction(
  "getStackOutputs",
  async (state) => {
    if (!state.stackName || !state.stackRegion) {
      state.errors.push(
        new Error(
          "No stack name or region provided. The stack name and \
region are required to fetch CFN outputs relevant to this example.",
        ),
      );
      return;
    }

    const outputs = await getCfnOutputs(state.stackName, state.stackRegion);
    Object.assign(state, outputs);
  },
```

```
);

export const promptForStackName = new ScenarioInput(
  "stackName",
  "Enter the name of the stack you deployed earlier.",
  { type: "input", default: "PoolsAndTriggersStack" },
);

export const promptForStackRegion = new ScenarioInput(
  "stackRegion",
  "Enter the region of the stack you deployed earlier.",
  { type: "input", default: "us-east-1" },
);

export const logCleanUpReminder = new ScenarioOutput(
  "logCleanUpReminder",
  "All done. Remember to run 'cdk destroy' to teardown the stack.",
  { skipWhen: skipWhenErrors },
);
```

Un handler per il trigger `PreSignUp` con una funzione Lambda.

```
import type { PreSignUpTriggerEvent, Handler } from "aws-lambda";
import type { UserRepository } from "./user-repository";
import { DynamoDBUserRepository } from "./user-repository";

export class PreSignUpHandler {
  private userRepository: UserRepository;

  constructor(userRepository: UserRepository) {
    this.userRepository = userRepository;
  }

  private isPreSignUpTriggerSource(event: PreSignUpTriggerEvent): boolean {
    return event.triggerSource === "PreSignUp_SignUp";
  }

  private getEventUserEmail(event: PreSignUpTriggerEvent): string {
    return event.request.userAttributes.email;
  }

  async handlePreSignUpTriggerEvent(
```

```
    event: PreSignUpTriggerEvent,
  ): Promise<PreSignUpTriggerEvent> {
    console.log(
      `Received presignup from ${event.triggerSource} for user
 '${event.userName}'`,
    );

    if (!this.isPreSignUpTriggerSource(event)) {
      return event;
    }

    const eventEmail = this.getEventUserEmail(event);
    console.log(`Looking up email ${eventEmail}.`);
    const storedUserInfo =
      await this.userRepository.getUserInfoByEmail(eventEmail);

    if (!storedUserInfo) {
      console.log(
        `Email ${eventEmail} not found. Email verification is required.`,
      );
      return event;
    }

    if (storedUserInfo.UserName !== event.userName) {
      console.log(
        `UserEmail ${eventEmail} found, but stored UserName
 '${storedUserInfo.UserName}' does not match supplied UserName
 '${event.userName}'. Verification is required.`,
      );
    } else {
      console.log(
        `UserEmail ${eventEmail} found with matching UserName
 ${storedUserInfo.UserName}. User is confirmed.`,
      );
      event.response.autoConfirmUser = true;
      event.response.autoVerifyEmail = true;
    }
    return event;
  }
}

const createPreSignUpHandler = (): PreSignUpHandler => {
  const tableName = process.env.TABLE_NAME;
  if (!tableName) {
```

```
    throw new Error("TABLE_NAME environment variable is not set");
  }

  const userRepository = new DynamoDBUserRepository(tableName);
  return new PreSignUpHandler(userRepository);
};

export const handler: Handler = async (event: PreSignUpTriggerEvent) => {
  const preSignUpHandler = createPreSignUpHandler();
  return preSignUpHandler.handlePreSignUpTriggerEvent(event);
};
```

Modulo di azioni CloudWatch Logs.

```
import {
  CloudWatchLogsClient,
  GetLogEventsCommand,
  OrderBy,
  paginateDescribeLogStreams,
} from "@aws-sdk/client-cloudwatch-logs";

/**
 * Get the latest log stream for a Lambda function.
 * @param {{ functionName: string, region: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cloudwatch-logs").LogStream | null,
 unknown]>}
 */
export const getLatestLogStreamForLambda = async ({ functionName, region }) => {
  try {
    const logGroupName = `/aws/lambda/${functionName}`;
    const cwlClient = new CloudWatchLogsClient({ region });
    const paginator = paginateDescribeLogStreams(
      { client: cwlClient },
      {
        descending: true,
        limit: 1,
        orderBy: OrderBy.LastEventTime,
        logGroupName,
      },
    );
```

```
    for await (const page of paginator) {
      return [page.logStreams[0], null];
    }
  } catch (err) {
    return [null, err];
  }
};

/**
 * Get the log events for a Lambda function's log stream.
 * @param {{
 *   functionName: string,
 *   logStreamName: string,
 *   eventCount: number,
 *   region: string
 * }} config
 * @returns {Promise<[import("@aws-sdk/client-cloudwatch-logs").OutputLogEvent[]
 | null, unknown]>}
 */
export const getLogEvents = async ({
  functionName,
  logStreamName,
  eventCount,
  region,
}) => {
  try {
    const cwlClient = new CloudWatchLogsClient({ region });
    const logGroupName = `/aws/lambda/${functionName}`;
    const response = await cwlClient.send(
      new GetLogEventsCommand({
        logStreamName: logStreamName,
        limit: eventCount,
        logGroupName: logGroupName,
      }),
    );

    return [response.events, null];
  } catch (err) {
    return [null, err];
  }
};
```

Modulo di azioni Amazon Cognito.

```
import {
  AdminGetUserCommand,
  CognitoIdentityProviderClient,
  DeleteUserCommand,
  InitiateAuthCommand,
  SignUpCommand,
  UpdateUserPoolCommand,
} from "@aws-sdk/client-cognito-identity-provider";

/**
 * Connect a Lambda function to the PreSignUp trigger for a Cognito user pool
 * @param {{ region: string, userPoolId: string, handlerArn: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").UpdateUserPoolCommandOutput | null, unknown]>}
 */
export const addPreSignUpHandler = async ({
  region,
  userPoolId,
  handlerArn,
}) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({
      region,
    });

    const command = new UpdateUserPoolCommand({
      UserPoolId: userPoolId,
      LambdaConfig: {
        PreSignUp: handlerArn,
      },
    });

    const response = await cognitoClient.send(command);
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};

/**
 * Attempt to register a user to a user pool with a given username and password.
```

```
 * @param {{
 *   region: string,
 *   userPoolClientId: string,
 *   username: string,
 *   email: string,
 *   password: string
 * }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").SignUpCommandOutput | null, unknown]>}
 */
export const signUpUser = async ({
  region,
  userPoolClientId,
  username,
  email,
  password,
}) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({
      region,
    });

    const response = await cognitoClient.send(
      new SignUpCommand({
        ClientId: userPoolClientId,
        Username: username,
        Password: password,
        UserAttributes: [{ Name: "email", Value: email }],
      }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};

/**
 * Sign in a user to Amazon Cognito using a username and password authentication
 flow.
 * @param {{ region: string, clientId: string, username: string, password:
 string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").InitiateAuthCommandOutput | null, unknown]>}
 */
```

```
export const signIn = async ({ region, clientId, username, password }) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({ region });
    const response = await cognitoClient.send(
      new InitiateAuthCommand({
        AuthFlow: "USER_PASSWORD_AUTH",
        ClientId: clientId,
        AuthParameters: { USERNAME: username, PASSWORD: password },
      }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};

/**
 * Retrieve an existing user from a user pool.
 * @param {{ region: string, userPoolId: string, username: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").AdminGetUserCommandOutput | null, unknown]>}
 */
export const getUser = async ({ region, userPoolId, username }) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({ region });
    const response = await cognitoClient.send(
      new AdminGetUserCommand({
        UserPoolId: userPoolId,
        Username: username,
      }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};

/**
 * Delete the signed-in user. Useful for allowing a user to delete their
 * own profile.
 * @param {{ region: string, accessToken: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").DeleteUserCommandOutput | null, unknown]>}
 */
```

```
export const deleteUser = async ({ region, accessToken }) => {
  try {
    const client = new CognitoIdentityProviderClient({ region });
    const response = await client.send(
      new DeleteUserCommand({ AccessToken: accessToken }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};
```

Modulo di azioni DynamoDB.

```
import { DynamoDBClient } from "@aws-sdk/client-dynamodb";
import {
  BatchWriteCommand,
  DynamoDBDocumentClient,
} from "@aws-sdk/lib-dynamodb";

/**
 * Populate a DynamoDB table with provide items.
 * @param {{ region: string, tableName: string, items: Record<string,
 unknown>[] }} config
 * @returns {Promise<[import("@aws-sdk/lib-dynamodb").BatchWriteCommandOutput |
 null, unknown]>}
 */
export const populateTable = async ({ region, tableName, items }) => {
  try {
    const ddbClient = new DynamoDBClient({ region });
    const docClient = DynamoDBDocumentClient.from(ddbClient);
    const response = await docClient.send(
      new BatchWriteCommand({
        RequestItems: {
          [tableName]: items.map((item) => ({
            PutRequest: {
              Item: item,
            },
          })),
        },
      }),
```

```
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per JavaScript .
  - DeleteUser
  - InitiateAuth
  - SignUp
  - UpdateUserPool

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.

Esegui automaticamente la migrazione di utenti Amazon Cognito noti con una funzione Lambda utilizzando un AWS SDK

L'esempio di codice seguente mostra come migrare automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda.

- Configura un pool di utenti per chiamare una funzione Lambda per il trigger `MigrateUser`.
- Accedi ad Amazon Cognito con un nome utente e un indirizzo e-mail non incluso nel pool di utenti.
- La funzione Lambda esegue la scansione di una tabella DynamoDB e migra automaticamente gli utenti noti al pool di utenti.
- Esegui il flusso della password dimenticata per reimpostare la password per l'utente migrato.
- Accedi come nuovo utente, quindi elimina le risorse.

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.

Esegui uno scenario interattivo al prompt dei comandi.

```
import (
 "context"
 "errors"
 "fmt"
 "log"
 "strings"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// MigrateUser separates the steps of this scenario into individual functions so
 that
// they are simpler to read and understand.
type MigrateUser struct {
 helper       IScenarioHelper
 questioner   demotools.IQuestioner
 resources    Resources
 cognitoActor *actions.CognitoActions
}

// NewMigrateUser constructs a new migrate user runner.
func NewMigrateUser(sdkConfig aws.Config, questioner demotools.IQuestioner,
 helper IScenarioHelper) MigrateUser {
 scenario := MigrateUser{
  helper:       helper,
  questioner:   questioner,
```

```
  resources:    Resources{},
  cognitoActor: &actions.CognitoActions{CognitoClient:
 cognitoidentityprovider.NewFromConfig(sdkConfig)},
 }
 scenario.resources.init(scenario.cognitoActor, questioner)
 return scenario
}

// AddMigrateUserTrigger adds a Lambda handler as an invocation target for the
 MigrateUser trigger.
func (runner *MigrateUser) AddMigrateUserTrigger(ctx context.Context, userPoolId
 string, functionArn string) {
 log.Printf("Let's add a Lambda function to handle the MigrateUser trigger from
 Cognito.\n" +
  "This trigger happens when an unknown user signs in, and lets your function
 take action before Cognito\n" +
  "rejects the user.\n\n")
 err := runner.cognitoActor.UpdateTriggers(
  ctx, userPoolId,
  actions.TriggerInfo{Trigger: actions.UserMigration, HandlerArn:
 aws.String(functionArn)})
 if err != nil {
  panic(err)
 }
 log.Printf("Lambda function %v added to user pool %v to handle the MigrateUser
 trigger.\n",
  functionArn, userPoolId)

 log.Println(strings.Repeat("-", 88))
}

// SignInUser adds a new user to the known users table and signs that user in to
 Amazon Cognito.
func (runner *MigrateUser) SignInUser(ctx context.Context, usersTable string,
 clientId string) (bool, actions.User) {
 log.Println("Let's sign in a user to your Cognito user pool. When the username
 and email matches an entry in the\n" +
  "DynamoDB known users table, the email is automatically verified and the user
 is migrated to the Cognito user pool.")

 user := actions.User{}
 user.UserName = runner.questioner.Ask("\nEnter a username:")
 user.UserEmail = runner.questioner.Ask("\nEnter an email that you own. This
 email will be used to confirm user migration\n" +
```

```
  "during this example:")

 runner.helper.AddKnownUser(ctx, usersTable, user)

 var err error
 var resetRequired *types.PasswordResetRequiredException
 var authResult *types.AuthenticationResultType
 signedIn := false
 for !signedIn && resetRequired == nil {
  log.Printf("Signing in to Cognito as user '%v'. The expected result is a
 PasswordResetRequiredException.\n\n", user.UserName)
  authResult, err = runner.cognitoActor.SignIn(ctx, clientId, user.UserName, "_")
  if err != nil {
   if errors.As(err, &resetRequired) {
    log.Printf("\nUser '%v' is not in the Cognito user pool but was found in the
 DynamoDB known users table.\n"+
     "User migration is started and a password reset is required.",
 user.UserName)
   } else {
    panic(err)
   }
  } else {
   log.Printf("User '%v' successfully signed in. This is unexpected and probably
 means you have not\n"+
    "cleaned up a previous run of this scenario, so the user exist in the Cognito
 user pool.\n"+
    "You can continue this example and select to clean up resources, or manually
 remove\n"+
    "the user from your user pool and try again.", user.UserName)
   runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
 *authResult.AccessToken)
   signedIn = true
  }
 }

 log.Println(strings.Repeat("-", 88))
 return resetRequired != nil, user
}

// ResetPassword starts a password recovery flow.
func (runner *MigrateUser) ResetPassword(ctx context.Context, clientId string,
 user actions.User) {
 wantCode := runner.questioner.AskBool(fmt.Sprintf("In order to migrate the user
 to Cognito, you must be able to receive a confirmation\n"+
```

```
 "code by email at %v. Do you want to send a code (y/n)?", user.UserEmail), "y")
if !wantCode {
 log.Println("To complete this example and successfully migrate a user to
Cognito, you must enter an email\n" +
  "you own that can receive a confirmation code.")
 return
}
codeDelivery, err := runner.cognitoActor.ForgotPassword(ctx, clientId,
user.UserName)
if err != nil {
 panic(err)
}
log.Printf("\nA confirmation code has been sent to %v.",
*codeDelivery.Destination)
code := runner.questioner.Ask("Check your email and enter it here:")

confirmed := false
password := runner.questioner.AskPassword("\nEnter a password that has at least
eight characters, uppercase, lowercase, numbers and symbols.\n"+
 "(the password will not display as you type):", 8)
for !confirmed {
 log.Printf("\nConfirming password reset for user '%v'.\n", user.UserName)
 err = runner.cognitoActor.ConfirmForgotPassword(ctx, clientId, code,
user.UserName, password)
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   password = runner.questioner.AskPassword("\nEnter another password:", 8)
  } else {
   panic(err)
  }
 } else {
  confirmed = true
 }
}
log.Printf("User '%v' successfully confirmed and migrated.\n", user.UserName)
log.Println("Signing in with your username and password...")
authResult, err := runner.cognitoActor.SignIn(ctx, clientId, user.UserName,
password)
if err != nil {
 panic(err)
}
log.Printf("Successfully signed in. Your access token starts with: %v...\n",
(*authResult.AccessToken)[:10])
```

```
 runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
 *authResult.AccessToken)

 log.Println(strings.Repeat("-", 88))
}

// Run runs the scenario.
func (runner *MigrateUser) Run(ctx context.Context, stackName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   runner.resources.Cleanup(ctx)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Printf("Welcome\n")

 log.Println(strings.Repeat("-", 88))

 stackOutputs, err := runner.helper.GetStackOutputs(ctx, stackName)
 if err != nil {
  panic(err)
 }
 runner.resources.userPoolId = stackOutputs["UserPoolId"]

 runner.AddMigrateUserTrigger(ctx, stackOutputs["UserPoolId"],
 stackOutputs["MigrateUserFunctionArn"])
 runner.resources.triggers = append(runner.resources.triggers,
 actions.UserMigration)
 resetNeeded, user := runner.SignInUser(ctx, stackOutputs["TableName"],
 stackOutputs["UserPoolClientId"])
 if resetNeeded {
  runner.helper.ListRecentLogEvents(ctx, stackOutputs["MigrateUserFunction"])
  runner.ResetPassword(ctx, stackOutputs["UserPoolClientId"], user)
 }

 runner.resources.Cleanup(ctx)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Gestisci il trigger `MigrateUser` con una funzione Lambda.

```
import (
 "context"
 "log"
 "os"

 "github.com/aws/aws-lambda-go/events"
 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/expression"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
)

const TABLE_NAME = "TABLE_NAME"

// UserInfo defines structured user data that can be marshalled to a DynamoDB
 format.
type UserInfo struct {
 UserName  string `dynamodbav:"UserName"`
 UserEmail string `dynamodbav:"UserEmail"`
}

type handler struct {
 dynamoClient *dynamodb.Client
}

// HandleRequest handles the MigrateUser event by looking up a user in an Amazon
 DynamoDB table and
// specifying whether they should be migrated to the user pool.
func (h *handler) HandleRequest(ctx context.Context, event
 events.CognitoEventUserPoolsMigrateUser)
 (events.CognitoEventUserPoolsMigrateUser, error) {
 log.Printf("Received migrate trigger from %v for user '%v'",
 event.TriggerSource, event.UserName)
 if event.TriggerSource != "UserMigration_Authentication" {
  return event, nil
 }
```

```
tableName := os.Getenv(TABLE_NAME)
user := UserInfo{
 UserName: event.UserName,
}
log.Printf("Looking up user '%v' in table %v.\n", user.UserName, tableName)
filterEx := expression.Name("UserName").Equal(expression.Value(user.UserName))
expr, err := expression.NewBuilder().WithFilter(filterEx).Build()
if err != nil {
 log.Printf("Error building expression to query for user '%v'.\n",
user.UserName)
 return event, err
}
output, err := h.dynamoClient.Scan(ctx, &dynamodb.ScanInput{
 TableName:                 aws.String(tableName),
 FilterExpression:          expr.Filter(),
 ExpressionAttributeNames:  expr.Names(),
 ExpressionAttributeValues: expr.Values(),
})
if err != nil {
 log.Printf("Error looking up user '%v'.\n", user.UserName)
 return event, err
}
if len(output.Items) == 0 {
 log.Printf("User '%v' not found, not migrating user.\n", user.UserName)
 return event, err
}

var users []UserInfo
err = attributevalue.UnmarshalListOfMaps(output.Items, &users)
if err != nil {
 log.Printf("Couldn't unmarshal DynamoDB items. Here's why: %v\n", err)
 return event, err
}

user = users[0]
log.Printf("UserName '%v' found with email %v. User is migrated and must reset
password.\n", user.UserName, user.UserEmail)
event.CognitoEventUserPoolsMigrateUserResponse.UserAttributes =
map[string]string{
 "email":          user.UserEmail,
 "email_verified": "true", // email_verified is required for the forgot password
flow.
}
```

```
 event.CognitoEventUserPoolsMigrateUserResponse.FinalUserStatus =
 "RESET_REQUIRED"
 event.CognitoEventUserPoolsMigrateUserResponse.MessageAction = "SUPPRESS"

 return event, err
}

func main() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 if err != nil {
  log.Panicln(err)
 }
 h := handler{
  dynamoClient: dynamodb.NewFromConfig(sdkConfig),
 }
 lambda.Start(h.HandleRequest)
}
```

Crea una struttura che esegue operazioni comuni.

```
import (
 "context"
 "log"
 "strings"
 "time"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// IScenarioHelper defines common functions used by the workflows in this
 example.
type IScenarioHelper interface {
 Pause(secs int)
```

```
 GetStackOutputs(ctx context.Context, stackName string) (actions.StackOutputs,
 error)
 PopulateUserTable(ctx context.Context, tableName string)
 GetKnownUsers(ctx context.Context, tableName string) (actions.UserList, error)
 AddKnownUser(ctx context.Context, tableName string, user actions.User)
 ListRecentLogEvents(ctx context.Context, functionName string)
}

// ScenarioHelper contains AWS wrapper structs used by the workflows in this
 example.
type ScenarioHelper struct {
 questioner  demotools.IQuestioner
 dynamoActor *actions.DynamoActions
 cfnActor    *actions.CloudFormationActions
 cwlActor    *actions.CloudWatchLogsActions
 isTestRun   bool
}

// NewScenarioHelper constructs a new scenario helper.
func NewScenarioHelper(sdkConfig aws.Config, questioner demotools.IQuestioner)
 ScenarioHelper {
 scenario := ScenarioHelper{
  questioner:  questioner,
  dynamoActor: &actions.DynamoActions{DynamoClient:
 dynamodb.NewFromConfig(sdkConfig)},
  cfnActor:    &actions.CloudFormationActions{CfnClient:
 cloudformation.NewFromConfig(sdkConfig)},
  cwlActor:    &actions.CloudWatchLogsActions{CwlClient:
 cloudwatchlogs.NewFromConfig(sdkConfig)},
 }
 return scenario
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
 if !helper.isTestRun {
  time.Sleep(time.Duration(secs) * time.Second)
 }
}

// GetStackOutputs gets the outputs from the specified CloudFormation stack in a
 structured format.
func (helper ScenarioHelper) GetStackOutputs(ctx context.Context, stackName
 string) (actions.StackOutputs, error) {
```

```
 return helper.cfnActor.GetOutputs(ctx, stackName), nil
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(ctx context.Context, tableName
 string) {
 log.Printf("First, let's add some users to the DynamoDB %v table we'll use for
 this example.\n", tableName)
 err := helper.dynamoActor.PopulateTable(ctx, tableName)
 if err != nil {
  panic(err)
 }
}

// GetKnownUsers gets the users from the known users table in a structured
 format.
func (helper ScenarioHelper) GetKnownUsers(ctx context.Context, tableName string)
 (actions.UserList, error) {
 knownUsers, err := helper.dynamoActor.Scan(ctx, tableName)
 if err != nil {
  log.Printf("Couldn't get known users from table %v. Here's why: %v\n",
 tableName, err)
 }
 return knownUsers, err
}

// AddKnownUser adds a user to the known users table.
func (helper ScenarioHelper) AddKnownUser(ctx context.Context, tableName string,
 user actions.User) {
 log.Printf("Adding user '%v' with email '%v' to the DynamoDB known users
 table...\n",
  user.UserName, user.UserEmail)
 err := helper.dynamoActor.AddUser(ctx, tableName, user)
 if err != nil {
  panic(err)
 }
}

// ListRecentLogEvents gets the most recent log stream and events for the
 specified Lambda function and displays them.
func (helper ScenarioHelper) ListRecentLogEvents(ctx context.Context,
 functionName string) {
 log.Println("Waiting a few seconds to let Lambda write to CloudWatch Logs...")
 helper.Pause(10)
```

```
 log.Println("Okay, let's check the logs to find what's happened recently with
 your Lambda function.")
 logStream, err := helper.cwlActor.GetLatestLogStream(ctx, functionName)
 if err != nil {
  panic(err)
 }
 log.Printf("Getting some recent events from log stream %v\n",
 *logStream.LogStreamName)
 events, err := helper.cwlActor.GetLogEvents(ctx, functionName,
 *logStream.LogStreamName, 10)
 if err != nil {
  panic(err)
 }
 for _, event := range events {
  log.Printf("\t%v", *event.Message)
 }
 log.Println(strings.Repeat("-", 88))
}
```

Crea una struttura che racchiude le azioni di Amazon Cognito.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
 trigger.
type Trigger int
```

```
const (
 PreSignUp Trigger = iota
 UserMigration
 PostAuthentication
)

type TriggerInfo struct {
 Trigger    Trigger
 HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
 is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(ctx context.Context, userPoolId
 string, triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(ctx,
 &cognitoidentityprovider.DescribeUserPoolInput{
  UserPoolId: aws.String(userPoolId),
 })
 if err != nil {
  log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
 userPoolId, err)
  return err
 }
 lambdaConfig := output.UserPool.LambdaConfig
 for _, trigger := range triggers {
  switch trigger.Trigger {
  case PreSignUp:
   lambdaConfig.PreSignUp = trigger.HandlerArn
  case UserMigration:
   lambdaConfig.UserMigration = trigger.HandlerArn
  case PostAuthentication:
   lambdaConfig.PostAuthentication = trigger.HandlerArn
  }
 }
 _, err = actor.CognitoClient.UpdateUserPool(ctx,
 &cognitoidentityprovider.UpdateUserPoolInput{
  UserPoolId:   aws.String(userPoolId),
  LambdaConfig: lambdaConfig,
 })
 if err != nil {
  log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
```

```
 }
 return err
}



// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(ctx context.Context, clientId string, userName
 string, password string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(ctx,
 &cognitoidentityprovider.SignUpInput{
  ClientId: aws.String(clientId),
  Password: aws.String(password),
  Username: aws.String(userName),
  UserAttributes: []types.AttributeType{
   {Name: aws.String("email"), Value: aws.String(userEmail)},
  },
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
  }
 } else {
  confirmed = output.UserConfirmed
 }
 return confirmed, err
}



// SignIn signs in a user to Amazon Cognito using a username and password
 authentication flow.
func (actor CognitoActions) SignIn(ctx context.Context, clientId string, userName
 string, password string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(ctx,
 &cognitoidentityprovider.InitiateAuthInput{
  AuthFlow:       "USER_PASSWORD_AUTH",
  ClientId:       aws.String(clientId),
  AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
```

```
 })
 if err != nil {
  var resetRequired *types.PasswordResetRequiredException
  if errors.As(err, &resetRequired) {
   log.Println(*resetRequired.Message)
  } else {
   log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
  }
 } else {
  authResult = output.AuthenticationResult
 }
 return authResult, err
}



// ForgotPassword starts a password recovery flow for a user. This flow typically
 sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(ctx context.Context, clientId string,
 userName string) (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(ctx,
 &cognitoidentityprovider.ForgotPasswordInput{
  ClientId: aws.String(clientId),
  Username: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't start password reset for user '%v'. Here;s why: %v\n",
 userName, err)
 }
 return output.CodeDeliveryDetails, err
}



// ConfirmForgotPassword confirms a user with a confirmation code and a new
 password.
func (actor CognitoActions) ConfirmForgotPassword(ctx context.Context, clientId
 string, code string, userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(ctx,
 &cognitoidentityprovider.ConfirmForgotPasswordInput{
  ClientId:         aws.String(clientId),
  ConfirmationCode: aws.String(code),
  Password:         aws.String(password),
```

```
  Username:         aws.String(userName),
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
  }
 }
 return err
}



// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(ctx context.Context, userAccessToken
 string) error {
 _, err := actor.CognitoClient.DeleteUser(ctx,
 &cognitoidentityprovider.DeleteUserInput{
  AccessToken: aws.String(userAccessToken),
 })
 if err != nil {
  log.Printf("Couldn't delete user. Here's why: %v\n", err)
 }
 return err
}



// AdminCreateUser uses administrator credentials to add a user to a user pool.
 This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(ctx context.Context, userPoolId
 string, userName string, userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(ctx,
 &cognitoidentityprovider.AdminCreateUserInput{
  UserPoolId:     aws.String(userPoolId),
  Username:       aws.String(userName),
  MessageAction:  types.MessageActionTypeSuppress,
  UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
 aws.String(userEmail)}},
 })
 if err != nil {
```

```
  var userExists *types.UsernameExistsException
  if errors.As(err, &userExists) {
   log.Printf("User %v already exists in the user pool.", userName)
   err = nil
  } else {
   log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
  }
 }
 return err
}



// AdminSetUserPassword uses administrator credentials to set a password for a
 user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(ctx context.Context, userPoolId
 string, userName string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(ctx,
 &cognitoidentityprovider.AdminSetUserPasswordInput{
  Password:   aws.String(password),
  UserPoolId: aws.String(userPoolId),
  Username:   aws.String(userName),
  Permanent:  true,
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
 err)
  }
 }
 return err
}
```

Crea una struttura che racchiude le azioni di DynamoDB.

```
import (
```

```
 "context"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

// DynamoActions encapsulates the Amazon Simple Notification Service (Amazon SNS)
 actions
// used in the examples.
type DynamoActions struct {
 DynamoClient *dynamodb.Client
}

// User defines structured user data.
type User struct {
 UserName  string
 UserEmail string
 LastLogin *LoginInfo `dynamodbav:",omitempty"`
}

// LoginInfo defines structured custom login data.
type LoginInfo struct {
 UserPoolId string
 ClientId   string
 Time       string
}

// UserList defines a list of users.
type UserList struct {
 Users []User
}

// UserNameList returns the usernames contained in a UserList as a list of
 strings.
func (users *UserList) UserNameList() []string {
 names := make([]string, len(users.Users))
 for i := 0; i < len(users.Users); i++ {
  names[i] = users.Users[i].UserName
 }
 return names
```

```
}

// PopulateTable adds a set of test users to the table.
func (actor DynamoActions) PopulateTable(ctx context.Context, tableName string)
 error {
 var err error
 var item map[string]types.AttributeValue
 var writeReqs []types.WriteRequest
 for i := 1; i < 4; i++ {
  item, err = attributevalue.MarshalMap(User{UserName: fmt.Sprintf("test_user_
%v", i), UserEmail: fmt.Sprintf("test_email_%v@example.com", i)})
  if err != nil {
   log.Printf("Couldn't marshall user into DynamoDB format. Here's why: %v\n",
 err)
   return err
  }
  writeReqs = append(writeReqs, types.WriteRequest{PutRequest:
 &types.PutRequest{Item: item}})
 }
 _, err = actor.DynamoClient.BatchWriteItem(ctx, &dynamodb.BatchWriteItemInput{
  RequestItems: map[string][]types.WriteRequest{tableName: writeReqs},
 })
 if err != nil {
  log.Printf("Couldn't populate table %v with users. Here's why: %v\n",
 tableName, err)
 }
 return err
}

// Scan scans the table for all items.
func (actor DynamoActions) Scan(ctx context.Context, tableName string) (UserList,
 error) {
 var userList UserList
 output, err := actor.DynamoClient.Scan(ctx, &dynamodb.ScanInput{
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't scan table %v for items. Here's why: %v\n", tableName,
 err)
 } else {
  err = attributevalue.UnmarshalListOfMaps(output.Items, &userList.Users)
  if err != nil {
   log.Printf("Couldn't unmarshal items into users. Here's why: %v\n", err)
  }
```

```
 }
 return userList, err
}

// AddUser adds a user item to a table.
func (actor DynamoActions) AddUser(ctx context.Context, tableName string, user
 User) error {
 userItem, err := attributevalue.MarshalMap(user)
 if err != nil {
  log.Printf("Couldn't marshall user to item. Here's why: %v\n", err)
 }
 _, err = actor.DynamoClient.PutItem(ctx, &dynamodb.PutItemInput{
  Item:      userItem,
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't put item in table %v. Here's why: %v", tableName, err)
 }
 return err
}
```

Crea una struttura che racchiuda le azioni di CloudWatch Logs.

```
import (
 "context"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs/types"
)

type CloudWatchLogsActions struct {
 CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(ctx context.Context,
 functionName string) (types.LogStream, error) {
```

```
 var logStream types.LogStream
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.DescribeLogStreams(ctx,
 &cloudwatchlogs.DescribeLogStreamsInput{
  Descending:   aws.Bool(true),
  Limit:        aws.Int32(1),
  LogGroupName: aws.String(logGroupName),
  OrderBy:      types.OrderByLastEventTime,
 })
 if err != nil {
  log.Printf("Couldn't get log streams for log group %v. Here's why: %v\n",
 logGroupName, err)
 } else {
  logStream = output.LogStreams[0]
 }
 return logStream, err
}

// GetLogEvents gets the most recent eventCount events from the specified log
 stream.
func (actor CloudWatchLogsActions) GetLogEvents(ctx context.Context, functionName
 string, logStreamName string, eventCount int32) (
 []types.OutputLogEvent, error) {
 var events []types.OutputLogEvent
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.GetLogEvents(ctx,
 &cloudwatchlogs.GetLogEventsInput{
  LogStreamName: aws.String(logStreamName),
  Limit:         aws.Int32(eventCount),
  LogGroupName:  aws.String(logGroupName),
 })
 if err != nil {
  log.Printf("Couldn't get log event for log stream %v. Here's why: %v\n",
 logStreamName, err)
 } else {
  events = output.Events
 }
 return events, err
}
```

Crea una struttura che racchiuda le azioni. CloudFormation

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
)

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
 CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
 structured format.
func (actor CloudFormationActions) GetOutputs(ctx context.Context, stackName
 string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(ctx,
 &cloudformation.DescribeStacksInput{
  StackName: aws.String(stackName),
 })
 if err != nil || len(output.Stacks) == 0 {
  log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
 stackName, err)
 }
 stackOutputs := StackOutputs{}
 for _, out := range output.Stacks[0].Outputs {
  stackOutputs[*out.OutputKey] = *out.OutputValue
 }
 return stackOutputs
}
```

Eliminare le risorse.

```
import (
 "context"
 "log"
```

```
 "user_pools_and_lambda_triggers/actions"

 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// Resources keeps track of AWS resources created during an example and handles
// cleanup when the example finishes.
type Resources struct {
 userPoolId       string
 userAccessTokens []string
 triggers         []actions.Trigger

 cognitoActor *actions.CognitoActions
 questioner   demotools.IQuestioner
}

func (resources *Resources) init(cognitoActor *actions.CognitoActions, questioner
 demotools.IQuestioner) {
 resources.userAccessTokens = []string{}
 resources.triggers = []actions.Trigger{}
 resources.cognitoActor = cognitoActor
 resources.questioner = questioner
}

// Cleanup deletes all AWS resources created during an example.
func (resources *Resources) Cleanup(ctx context.Context) {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong during cleanup.\n%v\n", r)
   log.Println("Use the AWS Management Console to remove any remaining resources
 \n" +
    "that were created for this scenario.")
  }
 }()

 wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
 resources that were created "+
  "during this demo (y/n)?", "y")
 if wantDelete {
  for _, accessToken := range resources.userAccessTokens {
   err := resources.cognitoActor.DeleteUser(ctx, accessToken)
   if err != nil {
    log.Println("Couldn't delete user during cleanup.")
    panic(err)
```

```
    }
    log.Println("Deleted user.")
   }
   triggerList := make([]actions.TriggerInfo, len(resources.triggers))
   for i := 0; i < len(resources.triggers); i++ {
    triggerList[i] = actions.TriggerInfo{Trigger: resources.triggers[i],
  HandlerArn: nil}
   }
   err := resources.cognitoActor.UpdateTriggers(ctx, resources.userPoolId,
  triggerList...)
   if err != nil {
    log.Println("Couldn't update Cognito triggers during cleanup.")
    panic(err)
   }
   log.Println("Removed Cognito triggers from user pool.")
  } else {
   log.Println("Be sure to remove resources when you're done with them to avoid
  unexpected charges!")
  }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Go .
  - ConfirmForgotPassword
  - DeleteUser
  - ForgotPassword
  - InitiateAuth
  - SignUp
  - UpdateUserPool

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.

Ottieni token client M2M e scopri i termini di Amazon Cognito utilizzando un AWS SDK

Il seguente esempio di codice mostra come utilizzare l'autorizzazione da macchina a macchina (M2M) di Amazon Cognito e l'individuazione dei termini.

- Implementa uno CloudFormation stack con risorse Cognito (pool di utenti, server di risorse, app client, documenti Terms).
- Recupera il segreto del client dell'app con. DescribeUserPoolClient
- Ottieni token di accesso M2M con. GetClientToken
- Scopri i documenti Terms con. DescribeTermsByClient
- Pulisci eliminando lo CloudFormation stack.

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.

Esegui uno scenario interattivo al prompt dei comandi.

```
class ClientTokenAndTermsScenario:
    """
    Runs an interactive scenario that demonstrates the Amazon Cognito
    GetClientToken and DescribeTermsByClient APIs.
    """

    def __init__(
        self,
        cognito_wrapper: CognitoIdentityProviderWrapper,
        cfn_client: BaseClient,
    ):
        """
        :param cognito_wrapper: An instance of the CognitoIdentityProviderWrapper
 class.
        :param cfn_client: A Boto3 CloudFormation client.
        """
```

```
        self.cognito_wrapper = cognito_wrapper
        self.cfn_client = cfn_client
        self.stack_name = None
        self.user_pool_id = None
        self.client_id = None
        self.client_secret = None

    def run(self) -> None:
        """Runs the scenario from start to finish."""
        print(SEPARATOR)
        print("Welcome to the Amazon Cognito Client Token and Terms Scenario.")
        print(SEPARATOR)
        print("This example creates Cognito resources in a CloudFormation stack,
then")
        print("demonstrates the new GetClientToken and DescribeTermsByClient
APIs.")
        print()

        try:
            self.setup()
            self.get_client_secret()
            self.demonstrate_get_client_token()
            self.demonstrate_describe_terms()
        except Exception:
            logger.exception("An error occurred during the scenario.")
        finally:
            self.cleanup()

        print(SEPARATOR)
        print("Amazon Cognito Client Token and Terms scenario completed.")

    # ------------------------------------------------------------------
    # Phase 1: Setup - deploy CloudFormation stack
    # ------------------------------------------------------------------
    def setup(self) -> None:
        """Deploys the CloudFormation prerequisite stack and retrieves
outputs."""
        stack_name = q.ask(
            f"\nEnter a name for the CloudFormation stack [{DEFAULT_STACK_NAME}]:
"
        )
        if not stack_name:
            stack_name = DEFAULT_STACK_NAME
```

```
        # Check whether the stack name already exists.
        while self._stack_exists(stack_name):
            print(f"A stack named '{stack_name}' already exists.")
            stack_name = q.ask("Enter a different stack name: ")

        self.stack_name = stack_name

        template_body = self._read_template()
        print(f"Deploying CloudFormation stack: {self.stack_name}")
        self.cfn_client.create_stack(
            StackName=self.stack_name,
            TemplateBody=template_body,
            Capabilities=["CAPABILITY_IAM"],
        )
        print(f"CloudFormation stack creation started: {self.stack_name}")
        print("Waiting for CloudFormation stack creation to complete...")

        waiter = self.cfn_client.get_waiter("stack_create_complete")
        try:
            waiter.wait(
                StackName=self.stack_name,
                WaiterConfig={"Delay": 10, "MaxAttempts": 60},
            )
        except WaiterError:
            logger.error("Stack creation failed or timed out.")
            self._print_stack_events()
            raise

        print("CloudFormation stack creation complete.")
        self._retrieve_stack_outputs()
        print(SEPARATOR)

    def _stack_exists(self, stack_name: str) -> bool:
        """Returns True if a stack with the given name exists and is not
deleted."""
        if not stack_name:
            return False
        try:
            response = self.cfn_client.describe_stacks(StackName=stack_name)
            stacks = response.get("Stacks", list())
            for stack in stacks:
                status = stack.get("StackStatus", "")
                if "DELETE_COMPLETE" not in status:
                    return True
```

```
            return False
        except ClientError as err:
            if "does not exist" in str(err):
                return False
            raise

    def _read_template(self) -> str:
        """Reads the CloudFormation template from the local file system."""
        template_path = os.path.join(
            os.path.dirname(os.path.abspath(__file__)), CFN_TEMPLATE_PATH
        )
        if not os.path.exists(template_path):
            raise FileNotFoundError(
                f"CloudFormation template not found at {template_path}. "
                "Ensure the resources/cfn_template.yaml file exists."
            )
        with open(template_path, "r") as f:
            return f.read()

    def _retrieve_stack_outputs(self) -> None:
        """Retrieves and displays outputs from the CloudFormation stack."""
        response = self.cfn_client.describe_stacks(StackName=self.stack_name)
        stacks = response.get("Stacks", list())
        if not stacks:
            raise RuntimeError(f"Stack '{self.stack_name}' not found.")

        outputs = stacks[0].get("Outputs", list())
        for output in outputs:
            key = output.get("OutputKey", "")
            value = output.get("OutputValue", "")
            print(f"Stack output {key}: {value}")
            if key == "UserPoolId":
                self.user_pool_id = value
            elif key == "ClientId":
                self.client_id = value

        if not self.user_pool_id or not self.client_id:
            raise RuntimeError(
                "Failed to retrieve UserPoolId or ClientId from stack outputs."
            )

    def _print_stack_events(self) -> None:
        """Prints recent stack events to help diagnose failures."""
        try:
```

```
            response =
self.cfn_client.describe_stack_events(StackName=self.stack_name)
            events = response.get("StackEvents", list())
            for event in events[:10]:
                status = event.get("ResourceStatus", "")
                reason = event.get("ResourceStatusReason", "")
                resource = event.get("LogicalResourceId", "")
                if "FAILED" in status:
                    print(f"  FAILED: {resource} - {reason}")
        except ClientError:
            logger.warning("Could not retrieve stack events.")

    # ------------------------------------------------------------------
    # Phase 2: Retrieve the app client secret
    # ------------------------------------------------------------------
    def get_client_secret(self) -> None:
        """Calls DescribeUserPoolClient to retrieve the client secret."""
        print(SEPARATOR)
        print("Retrieving the app client secret...")
        client_info = self.cognito_wrapper.describe_user_pool_client(
            self.user_pool_id, self.client_id
        )
        self.client_secret = client_info.get("ClientSecret", "")
        if not self.client_secret:
            raise RuntimeError(
                "Client secret not found. Ensure the app client was created "
                "with GenerateSecret: true."
            )
        print("Client secret retrieved successfully.")
        print(SEPARATOR)

    # ------------------------------------------------------------------
    # Phase 3: Demonstrate GetClientToken
    # ------------------------------------------------------------------
    def demonstrate_get_client_token(self) -> None:
        """Demonstrates obtaining M2M access tokens with GetClientToken."""
        print(SEPARATOR)
        print("Obtaining an M2M access token with GetClientToken...")
        print()

        # 3a: Get token with all scopes (no scopes param = all configured scopes)
        print("Step 1: Request a token with all configured scopes.")
        auth_result = self.cognito_wrapper.get_client_token(
            client_id=self.client_id,
```

```
            client_secret=self.client_secret,
        )
        self._display_token_result(auth_result, "all scopes")

        q.ask("\nPress Enter to continue...")

        # 3b: Get token with specific scopes
        print()
        print("Step 2: Request a token with a specific scope (my-m2m-api/read).")
        scoped_result = self.cognito_wrapper.get_client_token(
            client_id=self.client_id,
            client_secret=self.client_secret,
            scopes=["my-m2m-api/read"],
        )
        self._display_token_result(scoped_result, "my-m2m-api/read")

        print()
        print("Token was successfully scoped to the requested permissions.")
        print(SEPARATOR)

    @staticmethod
    def _display_token_result(auth_result: dict, scope_description: str) -> None:
        """Displays token metadata."""
        token_type = auth_result.get("TokenType", "Unknown")
        expires_in = auth_result.get("ExpiresIn", 0)
        access_token = auth_result.get("AccessToken", "")
        preview = access_token[:20] + "..." if len(access_token) > 20 else
access_token
        print(f"  Scope:      {scope_description}")
        print(f"  Token type: {token_type}")
        print(f"  Expires in: {expires_in} seconds")
        print(f"  Token preview: {preview}")

    # ------------------------------------------------------------------
    # Phase 4: Demonstrate DescribeTermsByClient
    # ------------------------------------------------------------------
    def demonstrate_describe_terms(self) -> None:
        """Demonstrates discovering Terms documents with
DescribeTermsByClient."""
        print(SEPARATOR)
        print("Discovering Terms documents with DescribeTermsByClient...")
        print()

        # 4a: Look up terms-of-use
```

```
        print("Looking up terms-of-use for the app client...")
        terms_of_use = self.cognito_wrapper.describe_terms_by_client(
            user_pool_id=self.user_pool_id,
            client_id=self.client_id,
            terms_name="terms-of-use",
        )
        self._display_terms(terms_of_use)

        q.ask("\nPress Enter to continue...")
        print()

        # 4b: Look up privacy-policy
        print("Looking up privacy-policy for the app client...")
        privacy_policy = self.cognito_wrapper.describe_terms_by_client(
            user_pool_id=self.user_pool_id,
            client_id=self.client_id,
            terms_name="privacy-policy",
        )
        self._display_terms(privacy_policy)

        print()
        print(
            "DescribeTermsByClient lets you discover which Terms documents are "
            "associated with a client without knowing the Terms resource ID up
front."
        )
        print(SEPARATOR)

    @staticmethod
    def _display_terms(terms: dict) -> None:
        """Displays a Terms document's details."""
        print(f"  Terms ID:          {terms.get('TermsId', 'N/A')}")
        print(f"  Terms name:        {terms.get('TermsName', 'N/A')}")
        print(f"  Client ID:         {terms.get('ClientId', 'N/A')}")
        print(f"  User pool ID:      {terms.get('UserPoolId', 'N/A')}")
        print(f"  Enforcement:       {terms.get('Enforcement', 'N/A')}")
        print(f"  Terms source:      {terms.get('TermsSource', 'N/A')}")
        links = terms.get("Links", dict())
        if links:
            print("  Links:")
            for lang_tag, url in links.items():
                print(f"    {lang_tag}: {url}")
        # Cognito returns CreationDate / LastModifiedDate as datetime objects
        # (boto3 deserializes the timestamps), so display them directly.
```

```
        creation_date = terms.get("CreationDate", None)
        if creation_date is not None:
            print(f"  Created:           {creation_date}")
        modified_date = terms.get("LastModifiedDate", None)
        if modified_date is not None:
            print(f"  Last modified:     {modified_date}")

    # ------------------------------------------------------------------
    # Phase 5: Cleanup - delete CloudFormation stack
    # ------------------------------------------------------------------
    def cleanup(self) -> None:
        """Prompts the user and deletes the CloudFormation stack."""
        if self.stack_name is None:
            return

        print(SEPARATOR)
        answer = q.ask(
            "Do you want to delete the CloudFormation stack and all resources?
(y/n) "
        )
        if answer.lower() != "y":
            print(
                f"Resources will remain. You can delete the stack later "
                f"through the AWS Console or CLI."
            )
            print(f"Stack name: {self.stack_name}")
            return

        print(
            f"CloudFormation stack '{self.stack_name}' is being deleted. "
            f"This may take a few minutes."
        )
        try:
            self.cfn_client.delete_stack(StackName=self.stack_name)
            waiter = self.cfn_client.get_waiter("stack_delete_complete")
            print("Waiting for CloudFormation stack deletion to complete...")
            waiter.wait(
                StackName=self.stack_name,
                WaiterConfig={"Delay": 10, "MaxAttempts": 60},
            )
            print(f"CloudFormation stack '{self.stack_name}' has been deleted.")
            print("All Cognito resources have been removed.")
        except (ClientError, WaiterError):
            logger.warning(
```

```
                "Stack deletion encountered an error. Attempting force delete."
            )
            try:
                self.cfn_client.delete_stack(
                    StackName=self.stack_name,
                    DeletionMode="FORCE_DELETE_STACK",
                )
                print("Force delete initiated. Please verify in the console.")
            except ClientError:
                logger.error(
                    "Force delete also failed. Clean up manually via the
 console."
                )
```

Crea una classe che racchiuda le operazioni di Amazon Cognito Identity Provider.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito Identity Provider actions."""

    def __init__(self, cognito_idp_client: BaseClient):
        """
        Initializes the CognitoIdentityProviderWrapper with an Amazon Cognito
        Identity Provider client.

        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        """
        self.cognito_idp_client = cognito_idp_client
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - DescribeTermsByClient
  - DescribeUserPoolClient
  - GetClientToken

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.

Nozioni di base sui bacini d'utenza

L'esempio di codice seguente mostra come:

- Creazione di un pool di utenti
- Creazione di un client dell'App
- Configura un dominio per il tuo pool di utenti
- Crea un utente come amministratore
- Abilita l'autoregistrazione
- Elenca gli utenti nel pool di utenti
- 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 Cognito User Pools Getting Started Script
# This script creates and configures an Amazon Cognito user pool with an app
 client

set -euo pipefail

# Security: Set restrictive umask
umask 0077

# Set up logging with secure permissions
```

```
LOG_FILE="cognito-user-pool-setup.log"
touch "$LOG_FILE"
chmod 600 "$LOG_FILE"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon Cognito User Pool setup 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 -qi "error\|failed"; then
    echo "ERROR: Command failed: $cmd" >&2
    echo "Output: $output" >&2
    cleanup_on_error
    exit 1
  fi
}

# Function to check AWS CLI return code
check_aws_error() {
  local exit_code=$?
  local cmd=$1

  if [ $exit_code -ne 0 ]; then
    echo "ERROR: AWS CLI command failed with exit code $exit_code: $cmd" >&2
    cleanup_on_error
    exit "$exit_code"
  fi
}

# Function to clean up resources on error
cleanup_on_error() {
  echo "Error encountered. Attempting to clean up resources..." >&2

  if [ -n "${DOMAIN_NAME:-}" ] && [ -n "${USER_POOL_ID:-}" ]; then
    echo "Deleting user pool domain: $DOMAIN_NAME" >&2
    aws cognito-idp delete-user-pool-domain \
      --user-pool-id "$USER_POOL_ID" \
      --domain "$DOMAIN_NAME" 2>/dev/null || true
  fi
```

```
  if [ -n "${USER_POOL_ID:-}" ]; then
    echo "Deleting user pool: $USER_POOL_ID" >&2
    aws cognito-idp delete-user-pool \
      --user-pool-id "$USER_POOL_ID" 2>/dev/null || true
  fi
}

# Set trap for cleanup on exit
trap cleanup_on_error EXIT ERR

# Validate AWS CLI is installed and configured
if ! command -v aws &> /dev/null; then
  echo "ERROR: AWS CLI is not installed" >&2
  exit 1
fi

# Validate jq is installed
if ! command -v jq &> /dev/null; then
  echo "ERROR: jq is not installed" >&2
  exit 1
fi

# Validate openssl is installed
if ! command -v openssl &> /dev/null; then
  echo "ERROR: openssl is not installed" >&2
  exit 1
fi

if ! aws sts get-caller-identity &> /dev/null; then
  echo "ERROR: AWS CLI is not configured or credentials are invalid" >&2
  exit 1
fi

# Get the current AWS region
AWS_REGION=$(aws configure get region)
if [ -z "$AWS_REGION" ]; then
  AWS_REGION="us-east-1" # Default region if not configured
fi
echo "Using AWS Region: $AWS_REGION"

# Validate region format
if ! [[ "$AWS_REGION" =~ ^[a-z]{2}-[a-z]+-[0-9]{1}$ ]]; then
  echo "ERROR: Invalid AWS region format: $AWS_REGION" >&2
  exit 1
```

```
fi

# Generate random identifier for resource names using secure method
RANDOM_ID=$(openssl rand -hex 6)
if [ -z "$RANDOM_ID" ]; then
  echo "ERROR: Failed to generate random identifier" >&2
  exit 1
fi

USER_POOL_NAME="MyUserPool-${RANDOM_ID}"
APP_CLIENT_NAME="MyAppClient-${RANDOM_ID}"
DOMAIN_NAME="my-auth-domain-${RANDOM_ID}"

# Validate resource names don't exceed limits
if [ ${#USER_POOL_NAME} -gt 128 ]; then
  echo "ERROR: User pool name exceeds maximum length of 128 characters" >&2
  exit 1
fi

if [ ${#APP_CLIENT_NAME} -gt 128 ]; then
  echo "ERROR: App client name exceeds maximum length of 128 characters" >&2
  exit 1
fi

echo "Using random identifier: $RANDOM_ID"
echo "User pool name: $USER_POOL_NAME"
echo "App client name: $APP_CLIENT_NAME"
echo "Domain name: $DOMAIN_NAME"

# Step 1: Create a User Pool
echo "Creating user pool..."
USER_POOL_OUTPUT=$(aws cognito-idp create-user-pool \
  --pool-name "$USER_POOL_NAME" \
  --auto-verified-attributes email \
  --username-attributes email \
  --policies '{"PasswordPolicy":
{"MinimumLength":12,"RequireUppercase":true,"RequireLowercase":true,"RequireNumbers":true
 \
  --schema '[{"Name":"email","Required":true,"Mutable":true}]' \
  --mfa-configuration OFF \
  --user-attribute-update-settings '{"AttributesRequireVerificationBeforeUpdate":
["email"]}' \
  --account-recovery-setting
 'RecoveryMechanisms=[{Name=verified_email,Priority=1}]' \
```

```
  --deletion-protection INACTIVE \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "create-user-pool"

# Extract the User Pool ID using jq for safety
USER_POOL_ID=$(echo "$USER_POOL_OUTPUT" | jq -r '.UserPool.Id // empty')
if [ -z "$USER_POOL_ID" ]; then
  echo "ERROR: Failed to extract User Pool ID" >&2
  exit 1
fi

# Validate User Pool ID format
if ! [[ "$USER_POOL_ID" =~ ^[a-z]{2}-[a-z]+-[0-9]+_[a-zA-Z0-9]+$ ]]; then
  echo "ERROR: Invalid User Pool ID format: $USER_POOL_ID" >&2
  exit 1
fi

echo "User Pool created with ID: $USER_POOL_ID"

USER_POOL_ARN=$(echo "$USER_POOL_OUTPUT" | jq -r '.UserPool.Arn // empty')
aws cognito-idp tag-resource \
  --resource-arn "$USER_POOL_ARN" \
  --tags project=doc-smith,tutorial=amazon-cognito-gs

# Wait for user pool to be ready
echo "Waiting for user pool to be ready..."
sleep 5

# Step 2: Create an App Client with enhanced security
echo "Creating app client..."
APP_CLIENT_OUTPUT=$(aws cognito-idp create-user-pool-client \
  --user-pool-id "$USER_POOL_ID" \
  --client-name "$APP_CLIENT_NAME" \
  --no-generate-secret \
  --explicit-auth-flows ALLOW_REFRESH_TOKEN_AUTH ALLOW_USER_PASSWORD_AUTH \
  --callback-urls '["https://localhost:3000/callback"]' \
  --allowed-o-auth-flows 'code' \
  --allowed-o-auth-scopes 'openid' 'email' 'profile' \
  --allowed-o-auth-flows-user-pool-client \
  --prevent-user-existence-errors ENABLED \
  --enable-token-revocation \
  --access-token-validity 1 \
  --id-token-validity 1 \
```

```
  --refresh-token-validity 30 \
  --token-validity-units 'AccessToken=hours,IdToken=hours,RefreshToken=days' \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "create-user-pool-client"

# Extract the Client ID using jq for safety
CLIENT_ID=$(echo "$APP_CLIENT_OUTPUT" | jq -r '.UserPoolClient.ClientId //
 empty')
if [ -z "$CLIENT_ID" ]; then
  echo "ERROR: Failed to extract Client ID" >&2
  cleanup_on_error
  exit 1
fi

# Validate Client ID format
if ! [[ "$CLIENT_ID" =~ ^[a-z0-9]{26}$ ]]; then
  echo "ERROR: Invalid Client ID format: $CLIENT_ID" >&2
  cleanup_on_error
  exit 1
fi

echo "App Client created with ID: $CLIENT_ID"

# Step 3: Set Up a Domain for Your User Pool
echo "Setting up user pool domain..."
DOMAIN_OUTPUT=$(aws cognito-idp create-user-pool-domain \
  --user-pool-id "$USER_POOL_ID" \
  --domain "$DOMAIN_NAME" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "create-user-pool-domain"
echo "Domain created: $DOMAIN_NAME.auth.$AWS_REGION.amazoncognito.com"

# Step 4: View User Pool Details
echo "Retrieving user pool details..."
USER_POOL_DETAILS=$(aws cognito-idp describe-user-pool \
  --user-pool-id "$USER_POOL_ID" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "describe-user-pool"
echo "User Pool details retrieved successfully"

# Step 5: View App Client Details
```

```
echo "Retrieving app client details..."
APP_CLIENT_DETAILS=$(aws cognito-idp describe-user-pool-client \
  --user-pool-id "$USER_POOL_ID" \
  --client-id "$CLIENT_ID" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "describe-user-pool-client"
echo "App Client details retrieved successfully"

# Step 6: Create a User (Admin)
echo "Creating admin user..."
ADMIN_USER_EMAIL="admin@example.com"
TEMP_PASSWORD="$(openssl rand -base64 12 | tr -d '\n')!@#"
if [ -z "$TEMP_PASSWORD" ]; then
  echo "ERROR: Failed to generate temporary password" >&2
  cleanup_on_error
  exit 1
fi

ADMIN_USER_OUTPUT=$(aws cognito-idp admin-create-user \
  --user-pool-id "$USER_POOL_ID" \
  --username "$ADMIN_USER_EMAIL" \
  --user-attributes Name=email,Value="$ADMIN_USER_EMAIL"
 Name=email_verified,Value=true \
  --temporary-password "$TEMP_PASSWORD" \
  --message-action SUPPRESS \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "admin-create-user"
echo "Admin user created: $ADMIN_USER_EMAIL"

# Securely clear temporary password from memory
unset TEMP_PASSWORD

# Step 7: Self-Registration
echo "Demonstrating self-registration..."
USER_EMAIL="user@example.com"
USER_PASSWORD="SecurePassword123!"
SIGNUP_OUTPUT=$(aws cognito-idp sign-up \
  --client-id "$CLIENT_ID" \
  --username "$USER_EMAIL" \
  --password "$USER_PASSWORD" \
  --user-attributes Name=email,Value="$USER_EMAIL" \
  --region "$AWS_REGION" \
```

```
  2>&1)
check_aws_error "sign-up"
echo "User signed up: $USER_EMAIL"
echo "A confirmation code would be sent to the user's email in a real scenario"

# Securely clear password from memory
unset USER_PASSWORD

echo ""
echo "================================================="
echo "IMPORTANT: In a real scenario, the user would receive"
echo "a confirmation code via email. For this demo, we'll"
echo "use admin-confirm-sign-up instead."
echo "================================================="
echo ""

# Step 8: Confirm User Registration (using admin privileges for demo)
echo "Confirming user registration (admin method)..."
CONFIRM_OUTPUT=$(aws cognito-idp admin-confirm-sign-up \
  --user-pool-id "$USER_POOL_ID" \
  --username "$USER_EMAIL" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "admin-confirm-sign-up"
echo "User confirmed: $USER_EMAIL"

# Step 9: Set permanent password for user
echo "Setting permanent password for user..."
SET_PASSWORD="SecureUserPassword123!"
SET_PASS_OUTPUT=$(aws cognito-idp admin-set-user-password \
  --user-pool-id "$USER_POOL_ID" \
  --username "$USER_EMAIL" \
  --password "$SET_PASSWORD" \
  --permanent \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "admin-set-user-password"
echo "Permanent password set for user"

unset SET_PASSWORD

# Step 10: Authenticate a User
echo "Authenticating user..."
AUTH_PASSWORD="SecureUserPassword123!"
```

```
AUTH_OUTPUT=$(aws cognito-idp initiate-auth \
  --client-id "$CLIENT_ID" \
  --auth-flow USER_PASSWORD_AUTH \
  --auth-parameters "USERNAME=$USER_EMAIL,PASSWORD=$AUTH_PASSWORD" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "initiate-auth"
echo "User authenticated successfully"

unset AUTH_PASSWORD

# Extract auth tokens securely
ID_TOKEN=$(echo "$AUTH_OUTPUT" | jq -r '.AuthenticationResult.IdToken // empty')
ACCESS_TOKEN=$(echo "$AUTH_OUTPUT" | jq -r '.AuthenticationResult.AccessToken //
 empty')

# Validate tokens exist
if [ -z "$ID_TOKEN" ] || [ -z "$ACCESS_TOKEN" ]; then
  echo "WARNING: Failed to extract authentication tokens" >&2
else
  echo "Authentication tokens obtained successfully"
fi

# Securely clear tokens from memory
unset ID_TOKEN
unset ACCESS_TOKEN

# Step 11: List Users in the User Pool
echo "Listing users in the user pool..."
USERS_OUTPUT=$(aws cognito-idp list-users \
  --user-pool-id "$USER_POOL_ID" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "list-users"
echo "Users listed successfully"

# Display summary of created resources
echo ""
echo "================================================"
echo "RESOURCE SUMMARY"
echo "================================================"
echo "User Pool ID: $USER_POOL_ID"
echo "User Pool Name: $USER_POOL_NAME"
echo "App Client ID: $CLIENT_ID"
```

```
echo "App Client Name: $APP_CLIENT_NAME"
echo "Domain: $DOMAIN_NAME.auth.$AWS_REGION.amazoncognito.com"
echo "Admin User: $ADMIN_USER_EMAIL"
echo "Regular User: $USER_EMAIL"
echo "================================================="
echo ""

# Auto-confirm cleanup
echo ""
echo "=========================================="
echo "CLEANUP"
echo "=========================================="
echo "Starting cleanup process..."

# Step 12: Clean Up Resources
echo "Deleting user pool domain..."
DELETE_DOMAIN_OUTPUT=$(aws cognito-idp delete-user-pool-domain \
  --user-pool-id "$USER_POOL_ID" \
  --domain "$DOMAIN_NAME" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "delete-user-pool-domain"
echo "Domain deleted successfully"

# Wait for domain deletion to complete
echo "Waiting for domain deletion to complete..."
sleep 5

echo "Deleting user pool (this will also delete the app client)..."
DELETE_POOL_OUTPUT=$(aws cognito-idp delete-user-pool \
  --user-pool-id "$USER_POOL_ID" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "delete-user-pool"
echo "User pool deleted successfully"

echo "All resources have been cleaned up"

echo "Script completed at $(date)"

# Remove trap to prevent cleanup on successful exit
trap - EXIT ERR
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AdminConfirmSignUp
  - AdminCreateUser
  - CreateUserPool
  - CreateUserPoolClient
  - CreateUserPoolDomain
  - DeleteUserPool
  - DeleteUserPoolDomain
  - DescribeUserPool
  - DescribeUserPoolClient
  - InitiateAuth
  - ListUsers
  - SignUp

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.

Registra un utente con un pool di utenti Amazon Cognito che richiede l'autenticazione a più fattori utilizzando un AWS SDK

Gli esempi di codice seguenti mostrano come:

- Registra e conferma un utente con nome utente, password e indirizzo e-mail.
- Configura l'autenticazione a più fattori associando un'applicazione MFA all'utente.
- Accedi utilizzando una password e un codice MFA.

.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 CognitoBasics;

public class CognitoBasics
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for Amazon Cognito.
        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<IAmazonCognitoIdentityProvider>()
            .AddTransient<CognitoWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<CognitoBasics>();

        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();
```

```
        var cognitoWrapper = host.Services.GetRequiredService<CognitoWrapper>();

        Console.WriteLine(new string('-', 80));
        UiMethods.DisplayOverview();
        Console.WriteLine(new string('-', 80));

        // clientId - The app client Id value that you get from the AWS CDK
script.
        var clientId = configuration["ClientId"]; // "*** REPLACE WITH CLIENT ID
VALUE FROM CDK SCRIPT";

        // poolId - The pool Id that you get from the AWS CDK script.
        var poolId = configuration["PoolId"]!; // "*** REPLACE WITH POOL ID VALUE
FROM CDK SCRIPT";
        var userName = configuration["UserName"];
        var password = configuration["Password"];
        var email = configuration["Email"];

        // If the username wasn't set in the configuration file,
        // get it from the user now.
        if (userName is null)
        {
            do
            {
                Console.Write("Username: ");
                userName = Console.ReadLine();
            }
            while (string.IsNullOrEmpty(userName));
        }
        Console.WriteLine($"\nUsername: {userName}");

        // If the password wasn't set in the configuration file,
        // get it from the user now.
        if (password is null)
        {
            do
            {
                Console.Write("Password: ");
                password = Console.ReadLine();
            }
            while (string.IsNullOrEmpty(password));
        }

        // If the email address wasn't set in the configuration file,
```

```
        // get it from the user now.
        if (email is null)
        {
            do
            {
                Console.Write("Email: ");
                email = Console.ReadLine();
            } while (string.IsNullOrEmpty(email));
        }

        // Now sign up the user.
        Console.WriteLine($"\nSigning up {userName} with email address:
{email}");
        await cognitoWrapper.SignUpAsync(clientId, userName, password, email);

        // Add the user to the user pool.
        Console.WriteLine($"Adding {userName} to the user pool");
        await cognitoWrapper.GetAdminUserAsync(userName, poolId);

        UiMethods.DisplayTitle("Get confirmation code");
        Console.WriteLine($"Conformation code sent to {userName}.");
        Console.Write("Would you like to send a new code? (Y/N) ");
        var answer = Console.ReadLine();

        if (answer!.ToLower() == "y")
        {
            await cognitoWrapper.ResendConfirmationCodeAsync(clientId, userName);
            Console.WriteLine("Sending a new confirmation code");
        }

        Console.Write("Enter confirmation code (from Email): ");
        var code = Console.ReadLine();

        await cognitoWrapper.ConfirmSignupAsync(clientId, code, userName);

        UiMethods.DisplayTitle("Checking status");
        Console.WriteLine($"Rechecking the status of {userName} in the user
pool");
        await cognitoWrapper.GetAdminUserAsync(userName, poolId);

        Console.WriteLine($"Setting up authenticator for {userName} in the user
pool");
        var setupResponse = await cognitoWrapper.InitiateAuthAsync(clientId,
userName, password);
```

```
        var setupSession = await
 cognitoWrapper.AssociateSoftwareTokenAsync(setupResponse.Session);
        Console.Write("Enter the 6-digit code displayed in Google Authenticator:
 ");
        var setupCode = Console.ReadLine();

        var setupResult = await
 cognitoWrapper.VerifySoftwareTokenAsync(setupSession, setupCode);
        Console.WriteLine($"Setup status: {setupResult}");

        Console.WriteLine($"Now logging in {userName} in the user pool");
        var authSession = await cognitoWrapper.AdminInitiateAuthAsync(clientId,
 poolId, userName, password);

        Console.Write("Enter a new 6-digit code displayed in Google
 Authenticator: ");
        var authCode = Console.ReadLine();

        var authResult = await
 cognitoWrapper.AdminRespondToAuthChallengeAsync(userName, clientId, authCode,
 authSession, poolId);
        Console.WriteLine($"Authenticated and received access token:
 {authResult.AccessToken}");

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Cognito scenario is complete.");
        Console.WriteLine(new string('-', 80));
    }
}


using System.Net;

namespace CognitoActions;

/// <summary>
/// Methods to perform Amazon Cognito Identity Provider actions.
/// </summary>
public class CognitoWrapper
{
    private readonly IAmazonCognitoIdentityProvider _cognitoService;

    /// <summary>
```

```
    /// Constructor for the wrapper class containing Amazon Cognito actions.
    /// </summary>
    /// <param name="cognitoService">The Amazon Cognito client object.</param>
    public CognitoWrapper(IAmazonCognitoIdentityProvider cognitoService)
    {
        _cognitoService = cognitoService;
    }

    /// <summary>
    /// List the Amazon Cognito user pools for an account.
    /// </summary>
    /// <returns>A list of UserPoolDescriptionType objects.</returns>
    public async Task<List<UserPoolDescriptionType>> ListUserPoolsAsync()
    {
        var userPools = new List<UserPoolDescriptionType>();

        var userPoolsPaginator = _cognitoService.Paginators.ListUserPools(new
ListUserPoolsRequest());

        await foreach (var response in userPoolsPaginator.Responses)
        {
            userPools.AddRange(response.UserPools);
        }

        return userPools;
    }


    /// <summary>
    /// Get a list of users for the Amazon Cognito user pool.
    /// </summary>
    /// <param name="userPoolId">The user pool ID.</param>
    /// <returns>A list of users.</returns>
    public async Task<List<UserType>> ListUsersAsync(string userPoolId)
    {
        var request = new ListUsersRequest
        {
            UserPoolId = userPoolId
        };

        var users = new List<UserType>();

        var usersPaginator = _cognitoService.Paginators.ListUsers(request);
        await foreach (var response in usersPaginator.Responses)
```

```
        {
            users.AddRange(response.Users);
        }

        return users;
    }


    /// <summary>
    /// Respond to an admin authentication challenge.
    /// </summary>
    /// <param name="userName">The name of the user.</param>
    /// <param name="clientId">The client ID.</param>
    /// <param name="mfaCode">The multi-factor authentication code.</param>
    /// <param name="session">The current application session.</param>
    /// <param name="clientId">The user pool ID.</param>
    /// <returns>The result of the authentication response.</returns>
    public async Task<AuthenticationResultType> AdminRespondToAuthChallengeAsync(
        string userName,
        string clientId,
        string mfaCode,
        string session,
        string userPoolId)
    {
        Console.WriteLine("SOFTWARE_TOKEN_MFA challenge is generated");

        var challengeResponses = new Dictionary<string, string>();
        challengeResponses.Add("USERNAME", userName);
        challengeResponses.Add("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

        var respondToAuthChallengeRequest = new
AdminRespondToAuthChallengeRequest
        {
            ChallengeName = ChallengeNameType.SOFTWARE_TOKEN_MFA,
            ClientId = clientId,
            ChallengeResponses = challengeResponses,
            Session = session,
            UserPoolId = userPoolId,
        };

        var response = await
_cognitoService.AdminRespondToAuthChallengeAsync(respondToAuthChallengeRequest);
        Console.WriteLine($"Response to Authentication
{response.AuthenticationResult.TokenType}");
```

```
        return response.AuthenticationResult;
    }


    /// <summary>
    /// Verify the TOTP and register for MFA.
    /// </summary>
    /// <param name="session">The name of the session.</param>
    /// <param name="code">The MFA code.</param>
    /// <returns>The status of the software token.</returns>
    public async Task<VerifySoftwareTokenResponseType>
VerifySoftwareTokenAsync(string session, string code)
    {
        var tokenRequest = new VerifySoftwareTokenRequest
        {
            UserCode = code,
            Session = session,
        };

        var verifyResponse = await
_cognitoService.VerifySoftwareTokenAsync(tokenRequest);

        return verifyResponse.Status;
    }


    /// <summary>
    /// Get an MFA token to authenticate the user with the authenticator.
    /// </summary>
    /// <param name="session">The session name.</param>
    /// <returns>The session name.</returns>
    public async Task<string> AssociateSoftwareTokenAsync(string session)
    {
        var softwareTokenRequest = new AssociateSoftwareTokenRequest
        {
            Session = session,
        };

        var tokenResponse = await
_cognitoService.AssociateSoftwareTokenAsync(softwareTokenRequest);
        var secretCode = tokenResponse.SecretCode;

        Console.WriteLine($"Use the following secret code to set up the
authenticator: {secretCode}");
```

```
        return tokenResponse.Session;
    }


    /// <summary>
    /// Initiate an admin auth request.
    /// </summary>
    /// <param name="clientId">The client ID to use.</param>
    /// <param name="userPoolId">The ID of the user pool.</param>
    /// <param name="userName">The username to authenticate.</param>
    /// <param name="password">The user's password.</param>
    /// <returns>The session to use in challenge-response.</returns>
    public async Task<string> AdminInitiateAuthAsync(string clientId, string
 userPoolId, string userName, string password)
    {
        var authParameters = new Dictionary<string, string>();
        authParameters.Add("USERNAME", userName);
        authParameters.Add("PASSWORD", password);

        var request = new AdminInitiateAuthRequest
        {
            ClientId = clientId,
            UserPoolId = userPoolId,
            AuthParameters = authParameters,
            AuthFlow = AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
        };

        var response = await _cognitoService.AdminInitiateAuthAsync(request);
        return response.Session;
    }

    /// <summary>
    /// Initiate authorization.
    /// </summary>
    /// <param name="clientId">The client Id of the application.</param>
    /// <param name="userName">The name of the user who is authenticating.</
param>
    /// <param name="password">The password for the user who is authenticating.</
param>
    /// <returns>The response from the initiate auth request.</returns>
    public async Task<InitiateAuthResponse> InitiateAuthAsync(string clientId,
 string userName, string password)
    {
```

```
        var authParameters = new Dictionary<string, string>();
        authParameters.Add("USERNAME", userName);
        authParameters.Add("PASSWORD", password);

        var authRequest = new InitiateAuthRequest

        {
            ClientId = clientId,
            AuthParameters = authParameters,
            AuthFlow = AuthFlowType.USER_PASSWORD_AUTH,
        };

        var response = await _cognitoService.InitiateAuthAsync(authRequest);
        Console.WriteLine($"Result Challenge is : {response.ChallengeName}");

        return response;
    }

    /// <summary>
    /// Confirm that the user has signed up.
    /// </summary>
    /// <param name="clientId">The Id of this application.</param>
    /// <param name="code">The confirmation code sent to the user.</param>
    /// <param name="userName">The username.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ConfirmSignupAsync(string clientId, string code,
string userName)
    {
        var signUpRequest = new ConfirmSignUpRequest
        {
            ClientId = clientId,
            ConfirmationCode = code,
            Username = userName,
        };

        var response = await _cognitoService.ConfirmSignUpAsync(signUpRequest);
        if (response.HttpStatusCode == HttpStatusCode.OK)
        {
            Console.WriteLine($"{userName} was confirmed");
            return true;
        }
        return false;
    }
```

```
    /// <summary>
    /// Initiates and confirms tracking of the device.
    /// </summary>
    /// <param name="accessToken">The user's access token.</param>
    /// <param name="deviceKey">The key of the device from Amazon Cognito.</
param>
    /// <param name="deviceName">The device name.</param>
    /// <returns></returns>
    public async Task<bool> ConfirmDeviceAsync(string accessToken, string
 deviceKey, string deviceName)
    {
        var request = new ConfirmDeviceRequest
        {
            AccessToken = accessToken,
            DeviceKey = deviceKey,
            DeviceName = deviceName
        };

        var response = await _cognitoService.ConfirmDeviceAsync(request);
        return response.UserConfirmationNecessary;
    }


    /// <summary>
    /// Send a new confirmation code to a user.
    /// </summary>
    /// <param name="clientId">The Id of the client application.</param>
    /// <param name="userName">The username of user who will receive the code.</
param>
    /// <returns>The delivery details.</returns>
    public async Task<CodeDeliveryDetailsType> ResendConfirmationCodeAsync(string
 clientId, string userName)
    {
        var codeRequest = new ResendConfirmationCodeRequest
        {
            ClientId = clientId,
            Username = userName,
        };

        var response = await
 _cognitoService.ResendConfirmationCodeAsync(codeRequest);
```

```
        Console.WriteLine($"Method of delivery is
 {response.CodeDeliveryDetails.DeliveryMedium}");

        return response.CodeDeliveryDetails;
    }


    /// <summary>
    /// Get the specified user from an Amazon Cognito user pool with
 administrator access.
    /// </summary>
    /// <param name="userName">The name of the user.</param>
    /// <param name="poolId">The Id of the Amazon Cognito user pool.</param>
    /// <returns>Async task.</returns>
    public async Task<UserStatusType> GetAdminUserAsync(string userName, string
 poolId)
    {
        AdminGetUserRequest userRequest = new AdminGetUserRequest
        {
            Username = userName,
            UserPoolId = poolId,
        };

        var response = await _cognitoService.AdminGetUserAsync(userRequest);

        Console.WriteLine($"User status {response.UserStatus}");
        return response.UserStatus;
    }


    /// <summary>
    /// Sign up a new user.
    /// </summary>
    /// <param name="clientId">The client Id of the application.</param>
    /// <param name="userName">The username to use.</param>
    /// <param name="password">The user's password.</param>
    /// <param name="email">The email address of the user.</param>
    /// <returns>A Boolean value indicating whether the user was confirmed.</
returns>
    public async Task<bool> SignUpAsync(string clientId, string userName, string
 password, string email)
    {
        var userAttrs = new AttributeType
        {
```

```
            Name = "email",
            Value = email,
        };

        var userAttrsList = new List<AttributeType>();

        userAttrsList.Add(userAttrs);

        var signUpRequest = new SignUpRequest
        {
            UserAttributes = userAttrsList,
            Username = userName,
            ClientId = clientId,
            Password = password
        };

        var response = await _cognitoService.SignUpAsync(signUpRequest);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }

}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per .NET .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

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.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

//! Scenario that adds a user to an Amazon Cognito user pool.
/*!
  \sa gettingStartedWithUserPools()
  \param clientID: Client ID associated with an Amazon Cognito user pool.
  \param userPoolID: An Amazon Cognito user pool ID.
  \param clientConfig: Aws client configuration.
  \return bool: Successful completion.
 */
bool AwsDoc::Cognito::gettingStartedWithUserPools(const Aws::String &clientID,
                                                  const Aws::String &userPoolID,
                                                  const
 Aws::Client::ClientConfiguration &clientConfig) {
    printAsterisksLine();
    std::cout
            << "Welcome to the Amazon Cognito example scenario."
            << std::endl;
    printAsterisksLine();

    std::cout
            << "This scenario will add a user to an Amazon Cognito user pool."
            << std::endl;
    const Aws::String userName = askQuestion("Enter a new username: ");
    const Aws::String password = askQuestion("Enter a new password: ");
    const Aws::String email = askQuestion("Enter a valid email for the user: ");

    std::cout << "Signing up " << userName << std::endl;
```

```
   Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);
   bool userExists = false;
   do {
       // 1. Add a user with a username, password, and email address.
       Aws::CognitoIdentityProvider::Model::SignUpRequest request;
       request.AddUserAttributes(
               Aws::CognitoIdentityProvider::Model::AttributeType().WithName(
                       "email").WithValue(email));
       request.SetUsername(userName);
       request.SetPassword(password);
       request.SetClientId(clientID);
       Aws::CognitoIdentityProvider::Model::SignUpOutcome outcome =
               client.SignUp(request);

       if (outcome.IsSuccess()) {
           std::cout << "The signup request for " << userName << " was
successful."
                     << std::endl;
       }
       else if (outcome.GetError().GetErrorType() ==

Aws::CognitoIdentityProvider::CognitoIdentityProviderErrors::USERNAME_EXISTS) {
           std::cout
                   << "The username already exists. Please enter a different
username."
                   << std::endl;
           userExists = true;
       }
       else {
           std::cerr << "Error with CognitoIdentityProvider::SignUpRequest. "
                     << outcome.GetError().GetMessage()
                     << std::endl;
           return false;
       }
   } while (userExists);

   printAsterisksLine();
   std::cout << "Retrieving status of " << userName << " in the user pool."
             << std::endl;
   // 2. Confirm that the user was added to the user pool.
   if (!checkAdminUserStatus(userName, userPoolID, client)) {
       return false;
   }
```

```
   std::cout << "A confirmation code was sent to " << email << "." << std::endl;

   bool resend = askYesNoQuestion("Would you like to send a new code? (y/n) ");
   if (resend) {
       // Request a resend of the confirmation code to the email address.
(ResendConfirmationCode)
       Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeRequest
request;
       request.SetUsername(userName);
       request.SetClientId(clientID);

       Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeOutcome
outcome =
               client.ResendConfirmationCode(request);

       if (outcome.IsSuccess()) {
           std::cout
                   << "CognitoIdentityProvider::ResendConfirmationCode was
successful."
                   << std::endl;
       }
       else {
           std::cerr << "Error with
CognitoIdentityProvider::ResendConfirmationCode. "
                     << outcome.GetError().GetMessage()
                     << std::endl;
           return false;
       }
   }

   printAsterisksLine();

   {
       // 4. Send the confirmation code that's received in the email.
(ConfirmSignUp)
       const Aws::String confirmationCode = askQuestion(
               "Enter the confirmation code that was emailed: ");
       Aws::CognitoIdentityProvider::Model::ConfirmSignUpRequest request;
       request.SetClientId(clientID);
       request.SetConfirmationCode(confirmationCode);
       request.SetUsername(userName);

       Aws::CognitoIdentityProvider::Model::ConfirmSignUpOutcome outcome =
```

```
                client.ConfirmSignUp(request);

        if (outcome.IsSuccess()) {
            std::cout << "ConfirmSignup was Successful."
                      << std::endl;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::ConfirmSignUp. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    std::cout << "Rechecking the status of " << userName << " in the user pool."
              << std::endl;
    if (!checkAdminUserStatus(userName, userPoolID, client)) {
        return false;
    }

    printAsterisksLine();

    std::cout << "Initiating authorization using the username and password."
              << std::endl;

    Aws::String session;
    // 5. Initiate authorization with username and password. (AdminInitiateAuth)
    if (!adminInitiateAuthorization(clientID, userPoolID,  userName, password,
 session, client)) {
        return false;
    }

    printAsterisksLine();

    std::cout
            << "Starting setup of time-based one-time password (TOTP) multi-
factor authentication (MFA)."
            << std::endl;

    {
        // 6. Request a setup key for one-time password (TOTP)
        //    multi-factor authentication (MFA). (AssociateSoftwareToken)
        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenRequest
 request;
```

```
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenOutcome
 outcome =
                client.AssociateSoftwareToken(request);

        if (outcome.IsSuccess()) {
            std::cout
                    << "Enter this setup key into an authenticator app, for
 example Google Authenticator."
                    << std::endl;
            std::cout << "Setup key: " << outcome.GetResult().GetSecretCode()
                      << std::endl;
#ifdef USING_QR
            printAsterisksLine();
            std::cout << "\nOr scan the QR code in the file '" << QR_CODE_PATH <<
 "."
                      << std::endl;

            saveQRCode(std::string("otpauth://totp/") + userName + "?secret=" +
                       outcome.GetResult().GetSecretCode());
#endif // USING_QR
            session = outcome.GetResult().GetSession();
        }
        else {
            std::cerr << "Error with
 CognitoIdentityProvider::AssociateSoftwareToken. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }
    askQuestion("Type enter to continue...", alwaysTrueTest);

    printAsterisksLine();

    {
        Aws::String userCode = askQuestion(
                "Enter the 6 digit code displayed in the authenticator app: ");

        //  7. Send the MFA code copied from an authenticator app.
 (VerifySoftwareToken)
        Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenRequest request;
        request.SetUserCode(userCode);
```

```
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenOutcome outcome =
                client.VerifySoftwareToken(request);

        if (outcome.IsSuccess()) {
            std::cout << "Verification of the code was successful."
                      << std::endl;
            session = outcome.GetResult().GetSession();
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::VerifySoftwareToken. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    printAsterisksLine();
    std::cout << "You have completed the MFA authentication setup." << std::endl;
    std::cout << "Now, sign in." << std::endl;

    // 8. Initiate authorization again with username and password.
(AdminInitiateAuth)
    if (!adminInitiateAuthorization(clientID, userPoolID, userName, password,
session, client)) {
        return false;
    }

    Aws::String accessToken;
    {
        Aws::String mfaCode = askQuestion(
                "Re-enter the 6 digit code displayed in the authenticator app:
");

        // 9. Send a new MFA code copied from an authenticator app.
(AdminRespondToAuthChallenge)
        Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeRequest
request;
        request.AddChallengeResponses("USERNAME", userName);
        request.AddChallengeResponses("SOFTWARE_TOKEN_MFA_CODE", mfaCode);
        request.SetChallengeName(
```

```
Aws::CognitoIdentityProvider::Model::ChallengeNameType::SOFTWARE_TOKEN_MFA);
       request.SetClientId(clientID);
       request.SetUserPoolId(userPoolID);
       request.SetSession(session);

       Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeOutcome
outcome =
               client.AdminRespondToAuthChallenge(request);

       if (outcome.IsSuccess()) {
           std::cout << "Here is the response to the challenge.\n" <<

outcome.GetResult().GetAuthenticationResult().Jsonize().View().WriteReadable()
                     << std::endl;

           accessToken =
outcome.GetResult().GetAuthenticationResult().GetAccessToken();
       }
       else {
           std::cerr << "Error with
CognitoIdentityProvider::AdminRespondToAuthChallenge. "
                     << outcome.GetError().GetMessage()
                     << std::endl;
           return false;
       }

       std::cout << "You have successfully added a user to Amazon Cognito."
                 << std::endl;
   }

   if (askYesNoQuestion("Would you like to delete the user that you just added?
(y/n) ")) {
       // 10. Delete the user that you just added. (DeleteUser)
       Aws::CognitoIdentityProvider::Model::DeleteUserRequest request;
       request.SetAccessToken(accessToken);

       Aws::CognitoIdentityProvider::Model::DeleteUserOutcome outcome =
               client.DeleteUser(request);

       if (outcome.IsSuccess()) {
           std::cout << "The user " << userName << " was deleted."
                     << std::endl;
       }
```

```
        else {
            std::cerr << "Error with CognitoIdentityProvider::DeleteUser. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
    }

    return true;
}

//! Routine which checks the user status in an Amazon Cognito user pool.
/*!
 \sa checkAdminUserStatus()
 \param userName: A username.
 \param userPoolID: An Amazon Cognito user pool ID.
 \return bool: Successful completion.
 */
bool AwsDoc::Cognito::checkAdminUserStatus(const Aws::String &userName,
                                           const Aws::String &userPoolID,
                                           const
 Aws::CognitoIdentityProvider::CognitoIdentityProviderClient &client) {
    Aws::CognitoIdentityProvider::Model::AdminGetUserRequest request;
    request.SetUsername(userName);
    request.SetUserPoolId(userPoolID);

    Aws::CognitoIdentityProvider::Model::AdminGetUserOutcome outcome =
            client.AdminGetUser(request);

    if (outcome.IsSuccess()) {
        std::cout << "The status for " << userName << " is " <<

 Aws::CognitoIdentityProvider::Model::UserStatusTypeMapper::GetNameForUserStatusType(
                          outcome.GetResult().GetUserStatus()) << std::endl;
        std::cout << "Enabled is " << outcome.GetResult().GetEnabled() <<
 std::endl;
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::AdminGetUser. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

```
//! Routine which starts authorization of an Amazon Cognito user.
//! This routine requires administrator credentials.
/*!
 \sa adminInitiateAuthorization()
 \param clientID: Client ID of tracked device.
 \param userPoolID: An Amazon Cognito user pool ID.
 \param userName: A username.
 \param password: A password.
 \param sessionResult: String to receive a session token.
 \return bool: Successful completion.
 */
bool AwsDoc::Cognito::adminInitiateAuthorization(const Aws::String &clientID,
                                                 const Aws::String &userPoolID,
                                                 const Aws::String &userName,
                                                 const Aws::String &password,
                                                 Aws::String &sessionResult,
                                                 const
 Aws::CognitoIdentityProvider::CognitoIdentityProviderClient &client) {
    Aws::CognitoIdentityProvider::Model::AdminInitiateAuthRequest request;
    request.SetClientId(clientID);
    request.SetUserPoolId(userPoolID);
    request.AddAuthParameters("USERNAME", userName);
    request.AddAuthParameters("PASSWORD", password);
    request.SetAuthFlow(

 Aws::CognitoIdentityProvider::Model::AuthFlowType::ADMIN_USER_PASSWORD_AUTH);


    Aws::CognitoIdentityProvider::Model::AdminInitiateAuthOutcome outcome =
            client.AdminInitiateAuth(request);

    if (outcome.IsSuccess()) {
        std::cout << "Call to AdminInitiateAuth was successful." << std::endl;
        sessionResult = outcome.GetResult().GetSession();
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::AdminInitiateAuth. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per C++ .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

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.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminGetUserRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminGetUserResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminInitiateAuthRequest;
```

```
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminInitiateAuthResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminRespondToAuthChalleng
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminRespondToAuthChalleng
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AssociateSoftwareTokenRequ
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AssociateSoftwareTokenResp
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AttributeType;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AuthFlowType;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ChallengeNameType;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ConfirmSignUpRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ResendConfirmationCodeRequ
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ResendConfirmationCodeResp
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.SignUpRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.VerifySoftwareTokenRequest
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.VerifySoftwareTokenRespons
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * TIP: To set up the required user pool, run the AWS Cloud Development Kit (AWS
 * CDK) script provided in this GitHub repo at
 * resources/cdk/cognito_scenario_user_pool_with_mfa.
 *
 * This code example performs the following operations:
 *
 * 1. Invokes the signUp method to sign up a user.
 * 2. Invokes the adminGetUser method to get the user's confirmation status.
 * 3. Invokes the ResendConfirmationCode method if the user requested another
 * code.
 * 4. Invokes the confirmSignUp method.
 * 5. Invokes the AdminInitiateAuth to sign in. This results in being prompted
 * to set up TOTP (time-based one-time password). (The response is
 * “ChallengeName”: “MFA_SETUP”).
 * 6. Invokes the AssociateSoftwareToken method to generate a TOTP MFA private
 * key. This can be used with Google Authenticator.
 * 7. Invokes the VerifySoftwareToken method to verify the TOTP and register for
 * MFA.
 * 8. Invokes the AdminInitiateAuth to sign in again. This results in being
 * prompted to submit a TOTP (Response: “ChallengeName”: “SOFTWARE_TOKEN_MFA”).
 * 9. Invokes the AdminRespondToAuthChallenge to get back a token.
 */

public class CognitoMVP {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws NoSuchAlgorithmException,
 InvalidKeyException {
        final String usage = """

                Usage:
                    <clientId> <poolId>

                Where:
                    clientId - The app client Id value that you can get from the
 AWS CDK script.
                    poolId - The pool Id that you can get from the AWS CDK
 script.\s
                """;
```

```
        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String clientId = args[0];
        String poolId = args[1];
        CognitoIdentityProviderClient identityProviderClient =
CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon Cognito example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Enter your user name");
        Scanner in = new Scanner(System.in);
        String userName = in.nextLine();

        System.out.println("*** Enter your password");
        String password = in.nextLine();

        System.out.println("*** Enter your email");
        String email = in.nextLine();

        System.out.println("1. Signing up " + userName);
        signUp(identityProviderClient, clientId, userName, password, email);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Getting " + userName + " in the user pool");
        getAdminUser(identityProviderClient, userName, poolId);

        System.out
                .println("*** Conformation code sent to " + userName + ". Would
you like to send a new code? (Yes/No)");
        System.out.println(DASHES);

        System.out.println(DASHES);
        String ans = in.nextLine();

        if (ans.compareTo("Yes") == 0) {
```

```
            resendConfirmationCode(identityProviderClient, clientId, userName);
            System.out.println("3. Sending a new confirmation code");
        }
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Enter confirmation code that was emailed");
        String code = in.nextLine();
        confirmSignUp(identityProviderClient, clientId, code, userName);
        System.out.println("Rechecking the status of " + userName + " in the user
pool");
        getAdminUser(identityProviderClient, userName, poolId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Invokes the initiateAuth to sign in");
        AdminInitiateAuthResponse authResponse =
initiateAuth(identityProviderClient, clientId, userName, password,
                poolId);
        String mySession = authResponse.session();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Invokes the AssociateSoftwareToken method to
generate a TOTP key");
        String newSession = getSecretForAppMFA(identityProviderClient,
mySession);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Enter the 6-digit code displayed in Google
Authenticator");
        String myCode = in.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Verify the TOTP and register for MFA");
        verifyTOTP(identityProviderClient, newSession, myCode);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Re-enter a 6-digit code displayed in Google
Authenticator");
        String mfaCode = in.nextLine();
```

```
        AdminInitiateAuthResponse authResponse1 =
initiateAuth(identityProviderClient, clientId, userName, password,
                poolId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("9.  Invokes the AdminRespondToAuthChallenge");
        String session2 = authResponse1.session();
        adminRespondToAuthChallenge(identityProviderClient, userName, clientId,
mfaCode, session2);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("All Amazon Cognito operations were successfully
performed");
        System.out.println(DASHES);
    }

    // Respond to an authentication challenge.
    public static void adminRespondToAuthChallenge(CognitoIdentityProviderClient
identityProviderClient,
            String userName, String clientId, String mfaCode, String session) {
        System.out.println("SOFTWARE_TOKEN_MFA challenge is generated");
        Map<String, String> challengeResponses = new HashMap<>();

        challengeResponses.put("USERNAME", userName);
        challengeResponses.put("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

        AdminRespondToAuthChallengeRequest respondToAuthChallengeRequest =
AdminRespondToAuthChallengeRequest.builder()
                .challengeName(ChallengeNameType.SOFTWARE_TOKEN_MFA)
                .clientId(clientId)
                .challengeResponses(challengeResponses)
                .session(session)
                .build();

        AdminRespondToAuthChallengeResponse respondToAuthChallengeResult =
identityProviderClient
                .adminRespondToAuthChallenge(respondToAuthChallengeRequest);

System.out.println("respondToAuthChallengeResult.getAuthenticationResult()"
                + respondToAuthChallengeResult.authenticationResult());
    }
```

```
   // Verify the TOTP and register for MFA.
   public static void verifyTOTP(CognitoIdentityProviderClient
identityProviderClient, String session, String code) {
       try {
           VerifySoftwareTokenRequest tokenRequest =
VerifySoftwareTokenRequest.builder()
                   .userCode(code)
                   .session(session)
                   .build();

           VerifySoftwareTokenResponse verifyResponse =
identityProviderClient.verifySoftwareToken(tokenRequest);
           System.out.println("The status of the token is " +
verifyResponse.statusAsString());

       } catch (CognitoIdentityProviderException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }

   public static AdminInitiateAuthResponse
initiateAuth(CognitoIdentityProviderClient identityProviderClient,
           String clientId, String userName, String password, String userPoolId)
{
       try {
           Map<String, String> authParameters = new HashMap<>();
           authParameters.put("USERNAME", userName);
           authParameters.put("PASSWORD", password);

           AdminInitiateAuthRequest authRequest =
AdminInitiateAuthRequest.builder()
                   .clientId(clientId)
                   .userPoolId(userPoolId)
                   .authParameters(authParameters)
                   .authFlow(AuthFlowType.ADMIN_USER_PASSWORD_AUTH)
                   .build();

           AdminInitiateAuthResponse response =
identityProviderClient.adminInitiateAuth(authRequest);
           System.out.println("Result Challenge is : " +
response.challengeName());
           return response;
```

```
        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return null;
    }

    public static String getSecretForAppMFA(CognitoIdentityProviderClient
identityProviderClient, String session) {
        AssociateSoftwareTokenRequest softwareTokenRequest =
AssociateSoftwareTokenRequest.builder()
                .session(session)
                .build();

        AssociateSoftwareTokenResponse tokenResponse = identityProviderClient
                .associateSoftwareToken(softwareTokenRequest);
        String secretCode = tokenResponse.secretCode();
        System.out.println("Enter this token into Google Authenticator");
        System.out.println(secretCode);
        return tokenResponse.session();
    }

    public static void confirmSignUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String code,
            String userName) {
        try {
            ConfirmSignUpRequest signUpRequest = ConfirmSignUpRequest.builder()
                    .clientId(clientId)
                    .confirmationCode(code)
                    .username(userName)
                    .build();

            identityProviderClient.confirmSignUp(signUpRequest);
            System.out.println(userName + " was confirmed");

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void resendConfirmationCode(CognitoIdentityProviderClient
identityProviderClient, String clientId,
```

```
            String userName) {
        try {
            ResendConfirmationCodeRequest codeRequest =
ResendConfirmationCodeRequest.builder()
                    .clientId(clientId)
                    .username(userName)
                    .build();

            ResendConfirmationCodeResponse response =
identityProviderClient.resendConfirmationCode(codeRequest);
            System.out.println("Method of delivery is " +
response.codeDeliveryDetails().deliveryMediumAsString());

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void signUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String userName,
            String password, String email) {
        AttributeType userAttrs = AttributeType.builder()
                .name("email")
                .value(email)
                .build();

        List<AttributeType> userAttrsList = new ArrayList<>();
        userAttrsList.add(userAttrs);
        try {
            SignUpRequest signUpRequest = SignUpRequest.builder()
                    .userAttributes(userAttrsList)
                    .username(userName)
                    .clientId(clientId)
                    .password(password)
                    .build();

            identityProviderClient.signUp(signUpRequest);
            System.out.println("User has been signed up ");

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
```

```
    }

    public static void getAdminUser(CognitoIdentityProviderClient
 identityProviderClient, String userName,
            String poolId) {
        try {
            AdminGetUserRequest userRequest = AdminGetUserRequest.builder()
                    .username(userName)
                    .userPoolId(poolId)
                    .build();

            AdminGetUserResponse response =
 identityProviderClient.adminGetUser(userRequest);
            System.out.println("User status " + response.userStatusAsString());

        } catch (CognitoIdentityProviderException 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 .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

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.

Per un'esperienza ottimale, clona il GitHub repository ed esegui questo esempio. Il codice seguente rappresenta un esempio dell'applicazione completa.

```
import { logger } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { signUp } from "../../../actions/sign-up.js";
import { FILE_USER_POOLS } from "./constants.js";
import { getSecondValuesFromEntries } from "@aws-doc-sdk-examples/lib/utils/util-
csv.js";

const validateClient = (clientId) => {
  if (!clientId) {
    throw new Error(
      `App client id is missing. Did you run 'create-user-pool'?`,
    );
  }
};

const validateUser = (username, password, email) => {
  if (!(username && password && email)) {
    throw new Error(
      `Username, password, and email must be provided as arguments to the 'sign-
up' command.`,
    );
  }
};

const signUpHandler = async (commands) => {
  const [_, username, password, email] = commands;

  try {
    validateUser(username, password, email);
    /**
```

```
     * @type {string[]}
     */
    const values = getSecondValuesFromEntries(FILE_USER_POOLS);
    const clientId = values[0];
    validateClient(clientId);
    logger.log("Signing up.");
    await signUp({ clientId, username, password, email });
    logger.log(`Signed up. A confirmation email has been sent to: ${email}.`);
    logger.log(
      `Run 'confirm-sign-up ${username} <code>' to confirm your account.`,
    );
  } catch (err) {
    logger.error(err);
  }
};

export { signUpHandler };

const signUp = ({ clientId, username, password, email }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new SignUpCommand({
    ClientId: clientId,
    Username: username,
    Password: password,
    UserAttributes: [{ Name: "email", Value: email }],
  });

  return client.send(command);
};

import { logger } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { confirmSignUp } from "../../../actions/confirm-sign-up.js";
import { FILE_USER_POOLS } from "./constants.js";
import { getSecondValuesFromEntries } from "@aws-doc-sdk-examples/lib/utils/util-
csv.js";

const validateClient = (clientId) => {
  if (!clientId) {
    throw new Error(
      `App client id is missing. Did you run 'create-user-pool'?`,
    );
  }
};
```

```
const validateUser = (username) => {
  if (!username) {
    throw new Error(
      `Username name is missing. It must be provided as an argument to the
 'confirm-sign-up' command.`,
    );
  }
};

const validateCode = (code) => {
  if (!code) {
    throw new Error(
      `Verification code is missing. It must be provided as an argument to the
 'confirm-sign-up' command.`,
    );
  }
};

const confirmSignUpHandler = async (commands) => {
  const [_, username, code] = commands;

  try {
    validateUser(username);
    validateCode(code);
    /**
     * @type {string[]}
     */
    const values = getSecondValuesFromEntries(FILE_USER_POOLS);
    const clientId = values[0];
    validateClient(clientId);
    logger.log("Confirming user.");
    await confirmSignUp({ clientId, username, code });
    logger.log(
      `User confirmed. Run 'admin-initiate-auth ${username} <password>' to sign
 in.`,
    );
  } catch (err) {
    logger.error(err);
  }
};

export { confirmSignUpHandler };
```

```
const confirmSignUp = ({ clientId, username, code }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmSignUpCommand({
    ClientId: clientId,
    Username: username,
    ConfirmationCode: code,
  });

  return client.send(command);
};

import qrcode from "qrcode-terminal";
import { logger } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { adminInitiateAuth } from "../../../actions/admin-initiate-auth.js";
import { associateSoftwareToken } from "../../../actions/associate-software-
token.js";
import { FILE_USER_POOLS } from "./constants.js";
import { getFirstEntry } from "@aws-doc-sdk-examples/lib/utils/util-csv.js";

const handleMfaSetup = async (session, username) => {
  const { SecretCode, Session } = await associateSoftwareToken(session);

  // Store the Session for use with 'VerifySoftwareToken'.
  process.env.SESSION = Session;

  console.log(
    "Scan this code in your preferred authenticator app, then run 'verify-
software-token' to finish the setup.",
  );
  qrcode.generate(
    `otpauth://totp/${username}?secret=${SecretCode}`,
    { small: true },
    console.log,
  );
};

const handleSoftwareTokenMfa = (session) => {
  // Store the Session for use with 'AdminRespondToAuthChallenge'.
  process.env.SESSION = session;
};

const validateClient = (id) => {
  if (!id) {
```

```
    throw new Error(
      `User pool client id is missing. Did you run 'create-user-pool'?`,
    );
  }
};

const validateId = (id) => {
  if (!id) {
    throw new Error(`User pool id is missing. Did you run 'create-user-pool'?`);
  }
};

const validateUser = (username, password) => {
  if (!(username && password)) {
    throw new Error(
      `Username and password must be provided as arguments to the 'admin-
initiate-auth' command.`,
    );
  }
};

const adminInitiateAuthHandler = async (commands) => {
  const [_, username, password] = commands;

  try {
    validateUser(username, password);

    const [userPoolId, clientId] = getFirstEntry(FILE_USER_POOLS);
    validateId(userPoolId);
    validateClient(clientId);

    logger.log("Signing in.");
    const { ChallengeName, Session } = await adminInitiateAuth({
      clientId,
      userPoolId,
      username,
      password,
    });

    if (ChallengeName === "MFA_SETUP") {
      logger.log("MFA setup is required.");
      return handleMfaSetup(Session, username);
    }
```

```
    if (ChallengeName === "SOFTWARE_TOKEN_MFA") {
      handleSoftwareTokenMfa(Session);
      logger.log(`Run 'admin-respond-to-auth-challenge ${username} <totp>'`);
    }
  } catch (err) {
    logger.error(err);
  }
};

export { adminInitiateAuthHandler };

const adminInitiateAuth = ({ clientId, userPoolId, username, password }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new AdminInitiateAuthCommand({
    ClientId: clientId,
    UserPoolId: userPoolId,
    AuthFlow: AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
    AuthParameters: { USERNAME: username, PASSWORD: password },
  });

  return client.send(command);
};

import { logger } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { adminRespondToAuthChallenge } from "../../../actions/admin-respond-to-
auth-challenge.js";
import { getFirstEntry } from "@aws-doc-sdk-examples/lib/utils/util-csv.js";
import { FILE_USER_POOLS } from "./constants.js";

const verifyUsername = (username) => {
  if (!username) {
    throw new Error(
      `Username is missing. It must be provided as an argument to the 'admin-
respond-to-auth-challenge' command.`,
    );
  }
};

const verifyTotp = (totp) => {
  if (!totp) {
    throw new Error(
      `Time-based one-time password (TOTP) is missing. It must be provided as an
 argument to the 'admin-respond-to-auth-challenge' command.`,
```

```
    );
  }
};

const storeAccessToken = (token) => {
  process.env.AccessToken = token;
};

const adminRespondToAuthChallengeHandler = async (commands) => {
  const [_, username, totp] = commands;

  try {
    verifyUsername(username);
    verifyTotp(totp);

    const [userPoolId, clientId] = getFirstEntry(FILE_USER_POOLS);
    const session = process.env.SESSION;

    const { AuthenticationResult } = await adminRespondToAuthChallenge({
      clientId,
      userPoolId,
      username,
      totp,
      session,
    });

    storeAccessToken(AuthenticationResult.AccessToken);

    logger.log("Successfully authenticated.");
  } catch (err) {
    logger.error(err);
  }
};

export { adminRespondToAuthChallengeHandler };

const respondToAuthChallenge = ({
  clientId,
  username,
  session,
  userPoolId,
  code,
}) => {
  const client = new CognitoIdentityProviderClient({});
```

```
  const command = new RespondToAuthChallengeCommand({
    ChallengeName: ChallengeNameType.SOFTWARE_TOKEN_MFA,
    ChallengeResponses: {
      SOFTWARE_TOKEN_MFA_CODE: code,
      USERNAME: username,
    },
    ClientId: clientId,
    UserPoolId: userPoolId,
    Session: session,
  });

  return client.send(command);
};

import { logger } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { verifySoftwareToken } from "../../../actions/verify-software-token.js";

const validateTotp = (totp) => {
  if (!totp) {
    throw new Error(
      `Time-based one-time password (TOTP) must be provided to the 'validate-
software-token' command.`,
    );
  }
};
const verifySoftwareTokenHandler = async (commands) => {
  const [_, totp] = commands;

  try {
    validateTotp(totp);

    logger.log("Verifying TOTP.");
    await verifySoftwareToken(totp);
    logger.log("TOTP Verified. Run 'admin-initiate-auth' again to sign-in.");
  } catch (err) {
    logger.error(err);
  }
};

export { verifySoftwareTokenHandler };

const verifySoftwareToken = (totp) => {
  const client = new CognitoIdentityProviderClient({});
```

```
  // The 'Session' is provided in the response to 'AssociateSoftwareToken'.
  const session = process.env.SESSION;

  if (!session) {
    throw new Error(
      "Missing a valid Session. Did you run 'admin-initiate-auth'?",
    );
  }

  const command = new VerifySoftwareTokenCommand({
    Session: session,
    UserCode: totp,
  });

  return client.send(command);
};
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per JavaScript .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

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.

```
/**
 Before running this Kotlin code example, set up your development environment,
 including your credentials.

 For more information, see the following documentation:
 https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html

 TIP: To set up the required user pool, run the AWS Cloud Development
 Kit (AWS CDK) script provided in this GitHub repo at resources/cdk/
cognito_scenario_user_pool_with_mfa.

 This code example performs the following operations:

 1. Invokes the signUp method to sign up a user.
 2. Invokes the adminGetUser method to get the user's confirmation status.
 3. Invokes the ResendConfirmationCode method if the user requested another code.
 4. Invokes the confirmSignUp method.
 5. Invokes the initiateAuth to sign in. This results in being prompted to
 set up TOTP (time-based one-time password). (The response is “ChallengeName”:
 “MFA_SETUP”).
 6. Invokes the AssociateSoftwareToken method to generate a TOTP MFA private key.
 This can be used with Google Authenticator.
 7. Invokes the VerifySoftwareToken method to verify the TOTP and register for
 MFA.
 8. Invokes the AdminInitiateAuth to sign in again. This results in being
 prompted to submit a TOTP (Response: “ChallengeName”: “SOFTWARE_TOKEN_MFA”).
 9. Invokes the AdminRespondToAuthChallenge to get back a token.
 */

suspend fun main(args: Array<String>) {
    val usage = """
```

```
        Usage:
            <clientId> <poolId>
        Where:
            clientId - The app client Id value that you can get from the AWS CDK
script.
            poolId - The pool Id that you can get from the AWS CDK script.
    """

    if (args.size != 2) {
        println(usage)
        exitProcess(1)
    }

    val clientId = args[0]
    val poolId = args[1]

    // Use the console to get data from the user.
    println("*** Enter your use name")
    val inOb = Scanner(System.`in`)
    val userName = inOb.nextLine()
    println(userName)

    println("*** Enter your password")
    val password: String = inOb.nextLine()

    println("*** Enter your email")
    val email = inOb.nextLine()

    println("*** Signing up $userName")
    signUp(clientId, userName, password, email)

    println("*** Getting $userName in the user pool")
    getAdminUser(userName, poolId)

    println("*** Conformation code sent to $userName. Would you like to send a
new code? (Yes/No)")
    val ans = inOb.nextLine()

    if (ans.compareTo("Yes") == 0) {
        println("*** Sending a new confirmation code")
        resendConfirmationCode(clientId, userName)
    }
    println("*** Enter the confirmation code that was emailed")
    val code = inOb.nextLine()
```

```
    confirmSignUp(clientId, code, userName)

    println("*** Rechecking the status of $userName in the user pool")
    getAdminUser(userName, poolId)

    val authResponse = checkAuthMethod(clientId, userName, password, poolId)
    val mySession = authResponse.session
    val newSession = getSecretForAppMFA(mySession)
    println("*** Enter the 6-digit code displayed in Google Authenticator")
    val myCode = inOb.nextLine()

    // Verify the TOTP and register for MFA.
    verifyTOTP(newSession, myCode)
    println("*** Re-enter a 6-digit code displayed in Google Authenticator")
    val mfaCode: String = inOb.nextLine()
    val authResponse1 = checkAuthMethod(clientId, userName, password, poolId)
    val session2 = authResponse1.session
    adminRespondToAuthChallenge(userName, clientId, mfaCode, session2)
}

suspend fun checkAuthMethod(
    clientIdVal: String,
    userNameVal: String,
    passwordVal: String,
    userPoolIdVal: String,
): AdminInitiateAuthResponse {
    val authParas = mutableMapOf<String, String>()
    authParas["USERNAME"] = userNameVal
    authParas["PASSWORD"] = passwordVal

    val authRequest =
        AdminInitiateAuthRequest {
            clientId = clientIdVal
            userPoolId = userPoolIdVal
            authParameters = authParas
            authFlow = AuthFlowType.AdminUserPasswordAuth
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.adminInitiateAuth(authRequest)
        println("Result Challenge is ${response.challengeName}")
        return response
    }
```

```
}

suspend fun resendConfirmationCode(
    clientIdVal: String?,
    userNameVal: String?,
) {
    val codeRequest =
        ResendConfirmationCodeRequest {
            clientId = clientIdVal
            username = userNameVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.resendConfirmationCode(codeRequest)
        println("Method of delivery is " +
 (response.codeDeliveryDetails?.deliveryMedium))
    }
}

// Respond to an authentication challenge.
suspend fun adminRespondToAuthChallenge(
    userName: String,
    clientIdVal: String?,
    mfaCode: String,
    sessionVal: String?,
) {
    println("SOFTWARE_TOKEN_MFA challenge is generated")
    val challengeResponsesOb = mutableMapOf<String, String>()
    challengeResponsesOb["USERNAME"] = userName
    challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

    val adminRespondToAuthChallengeRequest =
        AdminRespondToAuthChallengeRequest {
            challengeName = ChallengeNameType.SoftwareTokenMfa
            clientId = clientIdVal
            challengeResponses = challengeResponsesOb
            session = sessionVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val respondToAuthChallengeResult =
 identityProviderClient.adminRespondToAuthChallenge(adminRespondToAuthChallengeRequest)
```

```
        println("respondToAuthChallengeResult.getAuthenticationResult()
 ${respondToAuthChallengeResult.authenticationResult}")
    }
}

// Verify the TOTP and register for MFA.
suspend fun verifyTOTP(
    sessionVal: String?,
    codeVal: String?,
) {
    val tokenRequest =
        VerifySoftwareTokenRequest {
            userCode = codeVal
            session = sessionVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val verifyResponse =
 identityProviderClient.verifySoftwareToken(tokenRequest)
        println("The status of the token is ${verifyResponse.status}")
    }
}

suspend fun getSecretForAppMFA(sessionVal: String?): String? {
    val softwareTokenRequest =
        AssociateSoftwareTokenRequest {
            session = sessionVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val tokenResponse =
 identityProviderClient.associateSoftwareToken(softwareTokenRequest)
        val secretCode = tokenResponse.secretCode
        println("Enter this token into Google Authenticator")
        println(secretCode)
        return tokenResponse.session
    }
}

suspend fun confirmSignUp(
    clientIdVal: String?,
    codeVal: String?,
```

```
    userNameVal: String?,
) {
    val signUpRequest =
        ConfirmSignUpRequest {
            clientId = clientIdVal
            confirmationCode = codeVal
            username = userNameVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        identityProviderClient.confirmSignUp(signUpRequest)
        println("$userNameVal  was confirmed")
    }
}

suspend fun getAdminUser(
    userNameVal: String?,
    poolIdVal: String?,
) {
    val userRequest =
        AdminGetUserRequest {
            username = userNameVal
            userPoolId = poolIdVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.adminGetUser(userRequest)
        println("User status ${response.userStatus}")
    }
}

suspend fun signUp(
    clientIdVal: String?,
    userNameVal: String?,
    passwordVal: String?,
    emailVal: String?,
) {
    val userAttrs =
        AttributeType {
            name = "email"
            value = emailVal
        }
```

```
    val userAttrsList = mutableListOf<AttributeType>()
    userAttrsList.add(userAttrs)
    val signUpRequest =
        SignUpRequest {
            userAttributes = userAttrsList
            username = userNameVal
            clientId = clientIdVal
            password = passwordVal
        }

    CognitoIdentityProviderClient.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        identityProviderClient.signUp(signUpRequest)
        println("User has been signed up")
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Kotlin.
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

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.

Creazione di una classe che racchiude le funzioni di Amazon Cognito utilizzate nello scenario.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def _secret_hash(self, user_name):
        """
        Calculates a secret hash from a user name and a client secret.

        :param user_name: The user name to use when calculating the hash.
        :return: The secret hash.
        """
        key = self.client_secret.encode()
        msg = bytes(user_name + self.client_id, "utf-8")
        secret_hash = base64.b64encode(
            hmac.new(key, msg, digestmod=hashlib.sha256).digest()
        ).decode()
```

```
        logger.info("Made secret hash for %s: %s.", user_name, secret_hash)
        return secret_hash

    def sign_up_user(self, user_name, password, user_email):
        """
        Signs up a new user with Amazon Cognito. This action prompts Amazon
Cognito
        to send an email to the specified email address. The email contains a
code that
        can be used to confirm the user.

        When the user already exists, the user status is checked to determine
whether
        the user has been confirmed.

        :param user_name: The user name that identifies the new user.
        :param password: The password for the new user.
        :param user_email: The email address for the new user.
        :return: True when the user is already confirmed with Amazon Cognito.
                 Otherwise, false.
        """
        try:
            kwargs = {
                "ClientId": self.client_id,
                "Username": user_name,
                "Password": password,
                "UserAttributes": [{"Name": "email", "Value": user_email}],
            }
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            response = self.cognito_idp_client.sign_up(**kwargs)
            confirmed = response["UserConfirmed"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "UsernameExistsException":
                response = self.cognito_idp_client.admin_get_user(
                    UserPoolId=self.user_pool_id, Username=user_name
                )
                logger.warning(
                    "User %s exists and is %s.", user_name,
response["UserStatus"]
                )
                confirmed = response["UserStatus"] == "CONFIRMED"
            else:
                logger.error(
```

```
                    "Couldn't sign up %s. Here's why: %s: %s",
                    user_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        return confirmed


    def resend_confirmation(self, user_name):
        """
        Prompts Amazon Cognito to resend an email with a new confirmation code.

        :param user_name: The name of the user who will receive the email.
        :return: Delivery information about where the email is sent.
        """
        try:
            kwargs = {"ClientId": self.client_id, "Username": user_name}
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            response = self.cognito_idp_client.resend_confirmation_code(**kwargs)
            delivery = response["CodeDeliveryDetails"]
        except ClientError as err:
            logger.error(
                "Couldn't resend confirmation to %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return delivery


    def confirm_user_sign_up(self, user_name, confirmation_code):
        """
        Confirms a previously created user. A user must be confirmed before they
        can sign in to Amazon Cognito.

        :param user_name: The name of the user to confirm.
        :param confirmation_code: The confirmation code sent to the user's
registered
                                  email address.
        :return: True when the confirmation succeeds.
```

```
        """
        try:
            kwargs = {
                "ClientId": self.client_id,
                "Username": user_name,
                "ConfirmationCode": confirmation_code,
            }
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            self.cognito_idp_client.confirm_sign_up(**kwargs)
        except ClientError as err:
            logger.error(
                "Couldn't confirm sign up for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return True


    def list_users(self):
        """
        Returns a list of the users in the current user pool.

        :return: The list of users.
        """
        try:
            response =
self.cognito_idp_client.list_users(UserPoolId=self.user_pool_id)
            users = response["Users"]
        except ClientError as err:
            logger.error(
                "Couldn't list users for %s. Here's why: %s: %s",
                self.user_pool_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return users
```

```
    def start_sign_in(self, user_name, password):
        """
        Starts the sign-in process for a user by using administrator credentials.
        This method of signing in is appropriate for code running on a secure
 server.

        If the user pool is configured to require MFA and this is the first sign-
in
        for the user, Amazon Cognito returns a challenge response to set up an
        MFA application. When this occurs, this function gets an MFA secret from
        Amazon Cognito and returns it to the caller.

        :param user_name: The name of the user to sign in.
        :param password: The user's password.
        :return: The result of the sign-in attempt. When sign-in is successful,
 this
                 returns an access token that can be used to get AWS credentials.
 Otherwise,
                 Amazon Cognito returns a challenge to set up an MFA application,
                 or a challenge to enter an MFA code from a registered MFA
 application.
        """
        try:
            kwargs = {
                "UserPoolId": self.user_pool_id,
                "ClientId": self.client_id,
                "AuthFlow": "ADMIN_USER_PASSWORD_AUTH",
                "AuthParameters": {"USERNAME": user_name, "PASSWORD": password},
            }
            if self.client_secret is not None:
                kwargs["AuthParameters"]["SECRET_HASH"] =
 self._secret_hash(user_name)
            response = self.cognito_idp_client.admin_initiate_auth(**kwargs)
            challenge_name = response.get("ChallengeName", None)
            if challenge_name == "MFA_SETUP":
                if (
                    "SOFTWARE_TOKEN_MFA"
                    in response["ChallengeParameters"]["MFAS_CAN_SETUP"]
                ):
                    response.update(self.get_mfa_secret(response["Session"]))
                else:
                    raise RuntimeError(
                        "The user pool requires MFA setup, but the user pool is
 not "
```

```
                        "configured for TOTP MFA. This example requires TOTP
MFA."
                    )
        except ClientError as err:
            logger.error(
                "Couldn't start sign in for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response


    def get_mfa_secret(self, session):
        """
        Gets a token that can be used to associate an MFA application with the
user.

        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :return: An MFA token that can be used to set up an MFA application.
        """
        try:
            response =
self.cognito_idp_client.associate_software_token(Session=session)
        except ClientError as err:
            logger.error(
                "Couldn't get MFA secret. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response


    def verify_mfa(self, session, user_code):
        """
        Verify a new MFA application that is associated with a user.
```

```
        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :param user_code: A code generated by the associated MFA application.
        :return: Status that indicates whether the MFA application is verified.
        """
        try:
            response = self.cognito_idp_client.verify_software_token(
                Session=session, UserCode=user_code
            )
        except ClientError as err:
            logger.error(
                "Couldn't verify MFA. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response


    def respond_to_mfa_challenge(self, user_name, session, mfa_code):
        """
        Responds to a challenge for an MFA code. This completes the second step
of
        a two-factor sign-in. When sign-in is successful, it returns an access
token
        that can be used to get AWS credentials from Amazon Cognito.

        :param user_name: The name of the user who is signing in.
        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :param mfa_code: A code generated by the associated MFA application.
        :return: The result of the authentication. When successful, this contains
an
                 access token for the user.
        """
        try:
            kwargs = {
                "UserPoolId": self.user_pool_id,
                "ClientId": self.client_id,
```

```
                "ChallengeName": "SOFTWARE_TOKEN_MFA",
                "Session": session,
                "ChallengeResponses": {
                    "USERNAME": user_name,
                    "SOFTWARE_TOKEN_MFA_CODE": mfa_code,
                },
            }
            if self.client_secret is not None:
                kwargs["ChallengeResponses"]["SECRET_HASH"] = self._secret_hash(
                    user_name
                )
            response =
self.cognito_idp_client.admin_respond_to_auth_challenge(**kwargs)
            auth_result = response["AuthenticationResult"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "ExpiredCodeException":
                logger.warning(
                    "Your MFA code has expired or has been used already. You
might have "
                    "to wait a few seconds until your app shows you a new code."
                )
            else:
                logger.error(
                    "Couldn't respond to mfa challenge for %s. Here's why: %s:
%s",
                    user_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return auth_result


    def confirm_mfa_device(
        self,
        user_name,
        device_key,
        device_group_key,
        device_password,
        access_token,
        aws_srp,
    ):
        """
```

```
        Confirms an MFA device to be tracked by Amazon Cognito. When a device is
        tracked, its key and password can be used to sign in without requiring a
new
        MFA code from the MFA application.

        :param user_name: The user that is associated with the device.
        :param device_key: The key of the device, returned by Amazon Cognito.
        :param device_group_key: The group key of the device, returned by Amazon
Cognito.
        :param device_password: The password that is associated with the device.
        :param access_token: The user's access token.
        :param aws_srp: A class that helps with Secure Remote Password (SRP)
                        calculations. The scenario associated with this example
uses
                        the warrant package.
        :return: True when the user must confirm the device. Otherwise, False.
When
                 False, the device is automatically confirmed and tracked.
        """
        srp_helper = aws_srp.AWSSRP(
            username=user_name,
            password=device_password,
            pool_id="_",
            client_id=self.client_id,
            client_secret=None,
            client=self.cognito_idp_client,
        )
        device_and_pw = f"{device_group_key}{device_key}:{device_password}"
        device_and_pw_hash = aws_srp.hash_sha256(device_and_pw.encode("utf-8"))
        salt = aws_srp.pad_hex(aws_srp.get_random(16))
        x_value = aws_srp.hex_to_long(aws_srp.hex_hash(salt +
device_and_pw_hash))
        verifier = aws_srp.pad_hex(pow(srp_helper.val_g, x_value,
srp_helper.big_n))
        device_secret_verifier_config = {
            "PasswordVerifier": base64.standard_b64encode(
                bytearray.fromhex(verifier)
            ).decode("utf-8"),
            "Salt":
base64.standard_b64encode(bytearray.fromhex(salt)).decode("utf-8"),
        }
        try:
            response = self.cognito_idp_client.confirm_device(
                AccessToken=access_token,
```

```
                DeviceKey=device_key,
                DeviceSecretVerifierConfig=device_secret_verifier_config,
            )
            user_confirm = response["UserConfirmationNecessary"]
        except ClientError as err:
            logger.error(
                "Couldn't confirm mfa device %s. Here's why: %s: %s",
                device_key,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return user_confirm


    def sign_in_with_tracked_device(
        self,
        user_name,
        password,
        device_key,
        device_group_key,
        device_password,
        aws_srp,
    ):
        """
        Signs in to Amazon Cognito as a user who has a tracked device. Signing in
        with a tracked device lets a user sign in without entering a new MFA
 code.

        Signing in with a tracked device requires that the client respond to the
 SRP
        protocol. The scenario associated with this example uses the warrant
 package
        to help with SRP calculations.

        For more information on SRP, see https://en.wikipedia.org/wiki/
Secure_Remote_Password_protocol.

        :param user_name: The user that is associated with the device.
        :param password: The user's password.
        :param device_key: The key of a tracked device.
        :param device_group_key: The group key of a tracked device.
        :param device_password: The password that is associated with the device.
```

```
        :param aws_srp: A class that helps with SRP calculations. The scenario
                        associated with this example uses the warrant package.
        :return: The result of the authentication. When successful, this contains
an
                 access token for the user.
        """
        try:
            srp_helper = aws_srp.AWSSRP(
                username=user_name,
                password=device_password,
                pool_id="_",
                client_id=self.client_id,
                client_secret=None,
                client=self.cognito_idp_client,
            )

            response_init = self.cognito_idp_client.initiate_auth(
                ClientId=self.client_id,
                AuthFlow="USER_PASSWORD_AUTH",
                AuthParameters={
                    "USERNAME": user_name,
                    "PASSWORD": password,
                    "DEVICE_KEY": device_key,
                },
            )
            if response_init["ChallengeName"] != "DEVICE_SRP_AUTH":
                raise RuntimeError(
                    f"Expected DEVICE_SRP_AUTH challenge but got
{response_init['ChallengeName']}."
                )

            auth_params = srp_helper.get_auth_params()
            auth_params["DEVICE_KEY"] = device_key
            response_auth = self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_SRP_AUTH",
                ChallengeResponses=auth_params,
            )
            if response_auth["ChallengeName"] != "DEVICE_PASSWORD_VERIFIER":
                raise RuntimeError(
                    f"Expected DEVICE_PASSWORD_VERIFIER challenge but got "
                    f"{response_init['ChallengeName']}."
                )
```

```
            challenge_params = response_auth["ChallengeParameters"]
            challenge_params["USER_ID_FOR_SRP"] = device_group_key + device_key
            cr = srp_helper.process_challenge(challenge_params, {"USERNAME":
 user_name})
            cr["USERNAME"] = user_name
            cr["DEVICE_KEY"] = device_key
            response_verifier =
 self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_PASSWORD_VERIFIER",
                ChallengeResponses=cr,
            )
            auth_tokens = response_verifier["AuthenticationResult"]
        except ClientError as err:
            logger.error(
                "Couldn't start client sign in for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return auth_tokens
```

Creazione di una classe che esegue lo scenario. Questo esempio registra un dispositivo MFA che deve essere monitorato da Amazon Cognito e mostra come accedere utilizzando la password e le informazioni del dispositivo monitorato. evitando così di dover inserire un nuovo codice MFA.

```
def run_scenario(cognito_idp_client, user_pool_id, client_id):
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Welcome to the Amazon Cognito user signup with MFA demo.")
    print("-" * 88)

    cog_wrapper = CognitoIdentityProviderWrapper(
        cognito_idp_client, user_pool_id, client_id
    )
```

```
    user_name = q.ask("Let's sign up a new user. Enter a user name: ",
q.non_empty)
    password = q.ask("Enter a password for the user: ", q.non_empty)
    email = q.ask("Enter a valid email address that you own: ", q.non_empty)
    confirmed = cog_wrapper.sign_up_user(user_name, password, email)
    while not confirmed:
        print(
            f"User {user_name} requires confirmation. Check {email} for "
            f"a verification code."
        )
        confirmation_code = q.ask("Enter the confirmation code from the email: ")
        if not confirmation_code:
            if q.ask("Do you need another confirmation code (y/n)? ",
q.is_yesno):
                delivery = cog_wrapper.resend_confirmation(user_name)
                print(
                    f"Confirmation code sent by {delivery['DeliveryMedium']} "
                    f"to {delivery['Destination']}."
                )
        else:
            confirmed = cog_wrapper.confirm_user_sign_up(user_name,
confirmation_code)
    print(f"User {user_name} is confirmed and ready to use.")
    print("-" * 88)

    print("Let's get a list of users in the user pool.")
    q.ask("Press Enter when you're ready.")
    users = cog_wrapper.list_users()
    if users:
        print(f"Found {len(users)} users:")
        pp(users)
    else:
        print("No users found.")
    print("-" * 88)

    print("Let's sign in and get an access token.")
    auth_tokens = None
    challenge = "ADMIN_USER_PASSWORD_AUTH"
    response = {}
    while challenge is not None:
        if challenge == "ADMIN_USER_PASSWORD_AUTH":
            response = cog_wrapper.start_sign_in(user_name, password)
            # start_sign_in may return an AuthenticationResult with no challenge
```

```
            # (for example, when the user pool does not require MFA). Read the
            # challenge defensively and stop if sign-in already succeeded.
            challenge = response.get("ChallengeName")
            if challenge is None:
                auth_tokens = response.get("AuthenticationResult")
                if auth_tokens is not None:
                    print(f"You're signed in as {user_name}.")
                    print("Here's your access token:")
                    pp(auth_tokens["AccessToken"])
                break
        elif response["ChallengeName"] == "MFA_SETUP":
            print("First, we need to set up an MFA application.")
            qr_img = qrcode.make(
                f"otpauth://totp/{user_name}?secret={response['SecretCode']}"
            )
            qr_img.save("qr.png")
            q.ask(
                "Press Enter to see a QR code on your screen. Scan it into an MFA
"
                "application, such as Google Authenticator."
            )
            webbrowser.open("qr.png")
            mfa_code = q.ask(
                "Enter the verification code from your MFA application: ",
q.non_empty
            )
            response = cog_wrapper.verify_mfa(response["Session"], mfa_code)
            print(f"MFA device setup {response['Status']}")
            print("Now that an MFA application is set up, let's sign in again.")
            print(
                "You might have to wait a few seconds for a new MFA code to
appear in "
                "your MFA application."
            )
            challenge = "ADMIN_USER_PASSWORD_AUTH"
        elif response["ChallengeName"] == "SOFTWARE_TOKEN_MFA":
            auth_tokens = None
            while auth_tokens is None:
                mfa_code = q.ask(
                    "Enter a verification code from your MFA application: ",
q.non_empty
                )
                auth_tokens = cog_wrapper.respond_to_mfa_challenge(
                    user_name, response["Session"], mfa_code
```

```
                )
            print(f"You're signed in as {user_name}.")
            print("Here's your access token:")
            pp(auth_tokens["AccessToken"])
            if "NewDeviceMetadata" in auth_tokens:
                print("And your device information:")
                pp(auth_tokens["NewDeviceMetadata"])
            challenge = None
        else:
            raise Exception(f"Got unexpected challenge
 {response['ChallengeName']}")
    print("-" * 88)

    # If sign-in returned neither a challenge nor an AuthenticationResult, there
    # are no tokens to work with; nothing more to do.
    if auth_tokens is None:
        return user_name

    # respond_to_mfa_challenge returns NewDeviceMetadata only when the user pool
    # has device tracking configured. When it's absent, skip the device-
confirmation
    # and tracked-device sign-in steps instead of failing on a missing key.
    device_metadata = auth_tokens.get("NewDeviceMetadata")
    if device_metadata is None:
        print(
            "This user pool doesn't track devices, so there's no device to
 confirm. "
            "To try the tracked-device sign-in flow, enable device tracking on
 the "
            "user pool and run this example again."
        )
        print("-" * 88)
    else:
        device_group_key = device_metadata["DeviceGroupKey"]
        device_key = device_metadata["DeviceKey"]
        device_password =
 base64.standard_b64encode(os.urandom(40)).decode("utf-8")

        print(
            "Let's confirm your MFA device so you don't have to re-enter MFA
 tokens for it."
        )
        q.ask("Press Enter when you're ready.")
        cog_wrapper.confirm_mfa_device(
```

```
            user_name,
            device_key,
            device_group_key,
            device_password,
            auth_tokens["AccessToken"],
            aws_srp,
        )
        print(f"Your device {device_key} is confirmed.")
        print("-" * 88)

        print(
            f"Now let's sign in as {user_name} from your confirmed device
 {device_key}.\n"
            f"Because this device is tracked by Amazon Cognito, you won't have to
 re-enter an MFA code."
        )
        q.ask("Press Enter when ready.")
        auth_tokens = cog_wrapper.sign_in_with_tracked_device(
            user_name, password, device_key, device_group_key, device_password,
 aws_srp
        )
        print("You're signed in. Your access token is:")
        pp(auth_tokens["AccessToken"])
        print("-" * 88)

    print("Don't forget to delete your user pool when you're done with this
 example.")
    print("\nThanks for watching!")
    print("-" * 88)
    return user_name


def main():
    parser = argparse.ArgumentParser(
        description="Shows how to sign up a new user with Amazon Cognito and
 associate "
        "the user with an MFA application for multi-factor authentication."
    )
    parser.add_argument(
        "user_pool_id", help="The ID of the user pool to use for the example."
    )
    parser.add_argument(
        "client_id", help="The ID of the client application to use for the
 example."
```

```
    )
    args = parser.parse_args()
    client = boto3.client("cognito-idp")
    created_user = None
    try:
        created_user = run_scenario(client, args.user_pool_id, args.client_id)
    except Exception:
        logging.exception("Something went wrong with the demo.")
    finally:
        # Clean up the user this scenario created, on success or failure, so a
        # run never leaves an orphaned user in the pool. Cleanup errors are
        # logged but do not mask the original outcome.
        if created_user is not None:
            try:
                client.admin_delete_user(
                    UserPoolId=args.user_pool_id, Username=created_user
                )
                print(f"Deleted user {created_user}.")
            except Exception:
                logging.exception(
                    "Couldn't delete user %s; delete it manually.", created_user
                )


if __name__ == "__main__":
    main()
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode

- RespondToAuthChallenge
- SignUp
- VerifySoftwareToken

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.

Il file `Package.swift`.

```
// swift-tools-version: 5.9
//
// The swift-tools-version declares the minimum version of Swift required to
// build this package.

import PackageDescription

let package = Package(
    name: "cognito-scenario",
    // Let Xcode know the minimum Apple platforms supported.
    platforms: [
        .macOS(.v13),
        .iOS(.v15)
    ],
    dependencies: [
        // Dependencies declare other packages that this package depends on.
        .package(
            url: "https://github.com/awslabs/aws-sdk-swift",
            from: "1.0.0"),
        .package(
            url: "https://github.com/apple/swift-argument-parser.git",
            branch: "main"
        )
    ],
    targets: [
```

```
        // Targets are the basic building blocks of a package, defining a module
 or a test suite.
        // Targets can depend on other targets in this package and products
        // from dependencies.
        .executableTarget(
            name: "cognito-scenario",
            dependencies: [
                .product(name: "AWSCognitoIdentityProvider", package: "aws-sdk-
swift"),
                .product(name: "ArgumentParser", package: "swift-argument-
parser")
            ],
            path: "Sources")

    ]
)
```

File del codice Swift.

```
// An example demonstrating various features of Amazon Cognito. Before running
// this Swift code example, set up your development environment, including
// your credentials.
//
// For more information, see the following documentation:
// https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
//
// TIP: To set up the required user pool, run the AWS Cloud Development Kit
// (AWS CDK) script provided in this GitHub repo at
// resources/cdk/cognito_scenario_user_pool_with_mfa.
//
// This example performs the following functions:
//
// 1. Invokes the signUp method to sign up a user.
// 2. Invokes the adminGetUser method to get the user's confirmation status.
// 3. Invokes the ResendConfirmationCode method if the user requested another
//    code.
// 4. Invokes the confirmSignUp method.
// 5. Invokes the initiateAuth to sign in. This results in being prompted to
//    set up TOTP (time-based one-time password). (The response is
//    “ChallengeName”: “MFA_SETUP”).
// 6. Invokes the AssociateSoftwareToken method to generate a TOTP MFA private
//    key. This can be used with Google Authenticator.
```

```
// 7. Invokes the VerifySoftwareToken method to verify the TOTP and register
//    for MFA.
// 8. Invokes the AdminInitiateAuth to sign in again. This results in being
//    prompted to submit a TOTP (Response: “ChallengeName”:
//    “SOFTWARE_TOKEN_MFA”).
// 9. Invokes the AdminRespondToAuthChallenge to get back a token.

import ArgumentParser
import Foundation

import AWSClientRuntime
import AWSCognitoIdentityProvider

struct ExampleCommand: ParsableCommand {
    @Argument(help: "The application clientId.")
    var clientId: String
    @Argument(help: "The user pool ID to use.")
    var poolId: String
    @Option(help: "Name of the Amazon Region to use")
    var region = "us-east-1"

    static var configuration = CommandConfiguration(
        commandName: "cognito-scenario",
        abstract: """
        Demonstrates various features of Amazon Cognito.
        """,
        discussion: """
        """
    )

    /// Prompt for an input string of at least a minimum length.
    ///
    /// - Parameters:
    ///   - prompt: The prompt string to display.
    ///   - minLength: The minimum number of characters to allow in the
    ///     response. Default value is 0.
    ///
    /// - Returns: The entered string.
    func stringRequest(_ prompt: String, minLength: Int = 1) -> String {
        while true {
            print(prompt, terminator: "")
            let str = readLine()

            guard let str else {
```

```
                continue
            }
            if str.count >= minLength {
                return str
            } else {
                print("*** Response must be at least \(minLength) character(s)
long.")
            }
        }
    }

    /// Ask a yes/no question.
    ///
    /// - Parameter prompt: A prompt string to print.
    ///
    /// - Returns: `true` if the user answered "Y", otherwise `false`.
    func yesNoRequest(_ prompt: String) -> Bool {
        while true {
            let answer = stringRequest(prompt).lowercased()
            if answer == "y" || answer == "n" {
                return answer == "y"
            }
        }
    }

    /// Get information about a specific user in a user pool.
    ///
    /// - Parameters:
    ///   - cipClient: The Amazon Cognito Identity Provider client to use.
    ///   - userName: The user to retrieve information about.
    ///   - userPoolId: The user pool to search for the specified user.
    ///
    /// - Returns: `true` if the user's information was successfully
    ///   retrieved. Otherwise returns `false`.
    func adminGetUser(cipClient: CognitoIdentityProviderClient, userName: String,
                      userPoolId: String) async -> Bool {
        do {
            let output = try await cipClient.adminGetUser(
                input: AdminGetUserInput(
                    userPoolId: userPoolId,
                    username: userName
                )
            )
```

```
            guard let userStatus = output.userStatus else {
                print("*** Unable to get the user's status.")
                return false
            }

            print("User status: \(userStatus)")
            return true
        } catch {
            return false
        }
    }

    /// Create a new user in a user pool.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The ID of the app client to create a user for.
    ///   - userName: The username for the new user.
    ///   - password: The new user's password.
    ///   - email: The new user's email address.
    ///
    /// - Returns: `true` if successful; otherwise `false`.
    func signUp(cipClient: CognitoIdentityProviderClient, clientId: String,
userName: String, password: String, email: String) async -> Bool {
        let emailAttr = CognitoIdentityProviderClientTypes.AttributeType(
            name: "email",
            value: email
        )

        let userAttrsList = [emailAttr]

        do {
            _ = try await cipClient.signUp(
                input: SignUpInput(
                    clientId: clientId,
                    password: password,
                    userAttributes: userAttrsList,
                    username: userName
                )

            )

            print("=====> User \(userName) signed up.")
        } catch _ as AWSCognitoIdentityProvider.UsernameExistsException {
```

```
            print("*** The username \(userName) already exists. Please use a
different one.")
            return false
        } catch let error as AWSCognitoIdentityProvider.InvalidPasswordException
{
            print("*** Error: The specified password is invalid. Reason:
\(error.properties.message ?? "<none available>").")
            return false
        } catch _ as AWSCognitoIdentityProvider.ResourceNotFoundException {
            print("*** Error: The specified client ID (\(clientId)) doesn't
exist.")
            return false
        } catch {
            print("*** Unexpected error: \(error)")
            return false
        }

        return true
    }

    /// Requests a new confirmation code be sent to the given user's contact
    /// method.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The application client ID.
    ///   - userName: The user to resend a code for.
    ///
    /// - Returns: `true` if a new code was sent successfully, otherwise
    ///   `false`.
    func resendConfirmationCode(cipClient: CognitoIdentityProviderClient,
clientId: String,
                                userName: String) async -> Bool {
        do {
            let output = try await cipClient.resendConfirmationCode(
                input: ResendConfirmationCodeInput(
                    clientId: clientId,
                    username: userName
                )
            )

            guard let deliveryMedium = output.codeDeliveryDetails?.deliveryMedium
else {
```

```
                print("*** Unable to get the delivery method for the resent
code.")
                return false
            }

            print("=====> A new code has been sent by \(deliveryMedium)")
            return true
        } catch {
            print("*** Unable to resend the confirmation code to user
\(userName).")
            return false
        }
    }

    /// Submit a confirmation code for the specified user. This is the code as
    /// entered by the user after they've received it by email or text
    /// message.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The app client ID the user is signing up for.
    ///   - userName: The username of the user whose code is being sent.
    ///   - code: The user's confirmation code.
    ///
    /// - Returns: `true` if the code was successfully confirmed; otherwise
`false`.
    func confirmSignUp(cipClient: CognitoIdentityProviderClient, clientId:
String,
                       userName: String, code: String) async -> Bool {
        do {
            _ = try await cipClient.confirmSignUp(
                input: ConfirmSignUpInput(
                    clientId: clientId,
                    confirmationCode: code,
                    username: userName
                )
            )

            print("=====> \(userName) has been confirmed.")
            return true
        } catch {
            print("=====> \(userName)'s code was entered incorrectly.")
            return false
        }
```

```
    }

    /// Begin an authentication session.
    ///
    /// - Parameters:
    ///   - cipClient: The `CongitoIdentityProviderClient` to use.
    ///   - clientId: The app client ID to use.
    ///   - userName: The username to check.
    ///   - password: The user's password.
    ///   - userPoolId: The user pool to use.
    ///
    /// - Returns: The session token associated with this authentication
    ///   session.
    func initiateAuth(cipClient: CognitoIdentityProviderClient, clientId: String,
                      userName: String, password: String,
                      userPoolId: String) async -> String? {
        var authParams: [String: String] = [:]

        authParams["USERNAME"] = userName
        authParams["PASSWORD"] = password

        do {
            let output = try await cipClient.adminInitiateAuth(
                input: AdminInitiateAuthInput(
                    authFlow:
CognitoIdentityProviderClientTypes.AuthFlowType.adminUserPasswordAuth,
                    authParameters: authParams,
                    clientId: clientId,
                    userPoolId: userPoolId
                )
            )

            guard let challengeName = output.challengeName else {
                print("*** Invalid response from the auth service.")
                return nil
            }

            print("=====> Response challenge is \(challengeName)")

            return output.session
        } catch _ as UserNotFoundException {
            print("*** The specified username, \(userName), doesn't exist.")
            return nil
        } catch _ as UserNotConfirmedException {
```

```
            print("*** The user \(userName) has not been confirmed.")
            return nil
        } catch {
            print("*** An unexpected error occurred.")
            return nil
        }
    }

    /// Request and display an MFA secret token that the user should enter
    /// into their authenticator to set it up for the user account.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - authSession: The authentication session to request an MFA secret
    ///     for.
    ///
    /// - Returns: A string containing the MFA secret token that should be
    ///   entered into the authenticator software.
    func getSecretForAppMFA(cipClient: CognitoIdentityProviderClient,
authSession: String?) async -> String? {
        do {
            let output = try await cipClient.associateSoftwareToken(
                input: AssociateSoftwareTokenInput(
                    session: authSession
                )
            )

            guard let secretCode = output.secretCode else {
                print("*** Unable to get the secret code")
                return nil
            }

            print("=====> Enter this token into Google Authenticator:
\(secretCode)")
            return output.session
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return nil
        } catch {
            print("*** An unexpected error occurred getting the secret for the
app's MFA.")
            return nil
        }
    }
```

```
    /// Confirm that the user's TOTP authenticator is configured correctly by
    /// sending a code to it to check that it matches successfully.
    ///
    /// - Parameters:
    ///   - cipClient: The `CongnitoIdentityProviderClient` to use.
    ///   - session: An authentication session previously returned by an
    ///     `associateSoftwareToken()` call.
    ///   - mfaCode: The 6-digit code currently displayed by the user's
    ///     authenticator, as provided by the user.
    func verifyTOTP(cipClient: CognitoIdentityProviderClient, session: String?,
mfaCode: String?) async {
        do {
            let output = try await cipClient.verifySoftwareToken(
                input: VerifySoftwareTokenInput(
                    session: session,
                    userCode: mfaCode
                )
            )

            guard let tokenStatus = output.status else {
                print("*** Unable to get the token's status.")
                return
            }
            print("=====> The token's status is: \(tokenStatus)")
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return
        } catch _ as CodeMismatchException {
            print("*** The specified MFA code doesn't match the expected value.")
            return
        } catch _ as UserNotFoundException {
            print("*** The specified username doesn't exist.")
            return
        } catch _ as UserNotConfirmedException {
            print("*** The user has not been confirmed.")
            return
        } catch {
            print("*** Error verifying the MFA token!")
            return
        }
    }

    /// Respond to the authentication challenge received from Cognito after
```

```
    /// initiating an authentication session. This involves sending a current
    /// MFA code to the service.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - userName: The user's username.
    ///   - clientId: The app client ID.
    ///   - userPoolId: The user pool to sign into.
    ///   - mfaCode: The 6-digit MFA code currently displayed by the user's
    ///     authenticator.
    ///   - session: The authentication session to continue processing.
    func adminRespondToAuthChallenge(cipClient: CognitoIdentityProviderClient,
userName: String,
                                     clientId: String, userPoolId: String,
mfaCode: String,
                                     session: String) async {
        print("=====> SOFTWARE_TOKEN_MFA challenge is generated...")

        var challengeResponsesOb: [String: String] = [:]
        challengeResponsesOb["USERNAME"] = userName
        challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

        do {
            let output = try await cipClient.adminRespondToAuthChallenge(
                input: AdminRespondToAuthChallengeInput(
                    challengeName:
CognitoIdentityProviderClientTypes.ChallengeNameType.softwareTokenMfa,
                    challengeResponses: challengeResponsesOb,
                    clientId: clientId,
                    session: session,
                    userPoolId: userPoolId
                )
            )

            guard let authenticationResult = output.authenticationResult else {
                print("*** Unable to get authentication result.")
                return
            }

            print("=====> Authentication result (JWTs are redacted):")
            print(authenticationResult)
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return
```

```
        } catch _ as CodeMismatchException {
            print("*** The specified MFA code doesn't match the expected value.")
            return
        } catch _ as UserNotFoundException {
            print("*** The specified username, \(userName), doesn't exist.")
            return
        } catch _ as UserNotConfirmedException {
            print("*** The user \(userName) has not been confirmed.")
            return
        } catch let error as NotAuthorizedException {
            print("*** Unauthorized access. Reason: \(error.properties.message ??
"<unknown>")")
        } catch {
            print("*** Error responding to the MFA challenge.")
            return
        }
    }

    /// Called by ``main()`` to run the bulk of the example.
    func runAsync() async throws {
        let config = try await
CognitoIdentityProviderClient.CognitoIdentityProviderClientConfig(region:
region)
        let cipClient = CognitoIdentityProviderClient(config: config)

        print("""
              This example collects information about a user, then creates that
user in the
              specified user pool. Then, it enables Multi-Factor Authentication
(MFA) for that
              user by associating an authenticator application (such as Google
Authenticator
              or a password manager that supports TOTP). Then, the user uses a
code from their
              authenticator application to sign in.

              """)

        let userName = stringRequest("Please enter a new username: ")
        let password = stringRequest("Enter a password: ")
        let email = stringRequest("Enter your email address: ", minLength: 5)

        // Submit the sign-up request to AWS.
```

```
        print("==> Signing up user \(userName)...")
        if await signUp(cipClient: cipClient, clientId: clientId,
                        userName: userName, password: password,
                        email: email) == false {
            return
        }

        // Check the user's status. This time, it should come back "unconfirmed".

        print("==> Getting the status of user \(userName) from the user pool
(should be 'unconfirmed')...")
        if await adminGetUser(cipClient: cipClient, userName: userName,
userPoolId: poolId) == false {
            return
        }

        // Ask the user if they want a replacement code sent, such as if the
        // code hasn't arrived yet. If the user responds with a "yes," send a
        // new code.

        if yesNoRequest("==> A confirmation code was sent to \(userName). Would
you like to send a new code (Y/N)? ") {
            print("==> Sending a new confirmation code...")
            if await resendConfirmationCode(cipClient: cipClient, clientId:
clientId, userName: userName) == false {
                return
            }
        }

        // Ask the user to enter the confirmation code, then send it to Amazon
        // Cognito to verify it.

        let code = stringRequest("==> Enter the confirmation code sent to
\(userName): ")
        if await confirmSignUp(cipClient: cipClient, clientId: clientId,
userName: userName, code: code) == false {
            // The code didn't match. Your application may wish to offer to
            // re-send the confirmation code here and try again.
            return
        }

        // Check the user's status again. This time it should come back
        // "confirmed".
```

```
        print("==> Rechecking status of user \(userName) in the user pool (should
be 'confirmed')...")
        if await adminGetUser(cipClient: cipClient, userName: userName,
userPoolId: poolId) == false {
            return
        }
        // Check the challenge mode. Here, it should be "mfaSetup", indicating
        // that the user needs to add MFA before using it. This returns a
        // session that can be used to register MFA, or nil if an error occurs.

        let authSession = await initiateAuth(cipClient: cipClient, clientId:
clientId,
                                             userName: userName, password:
password,
                                             userPoolId: poolId)
        if authSession == nil {
            return
        }

        // Ask Cognito for an MFA secret token that the user should enter into
        // their authenticator software (such as Google Authenticator) or
        // password manager to configure it for this user account. This
        // returns a new session that should be used for the new stage of the
        // authentication process.

        let newSession = await getSecretForAppMFA(cipClient: cipClient,
authSession: authSession)
        if newSession == nil {
            return
        }

        // Ask the user to enter the current 6-digit code displayed by their
        // authenticator. Then verify that it matches the value expected for
        // the session.

        let mfaCode1 = stringRequest("==> Enter the 6-digit code displayed in
your authenticator: ",
                                     minLength: 6)
        await verifyTOTP(cipClient: cipClient, session: newSession, mfaCode:
mfaCode1)

        // Ask the user to authenticate now that the authenticator has been
        // configured. This creates a new session using the user's username
        // and password as already entered.
```

```
        print("\nNow starting the sign-in process for user \(userName)...\n")

        let session2 = await initiateAuth(cipClient: cipClient, clientId:
 clientId,
                                          userName: userName, password: password,
 userPoolId: poolId)
        guard let session2 else {
            return
        }

        // Now that we have a new auth session, `session2`, ask the user for a
        // new 6-digit code from their authenticator, and send it to the auth
        // session.

        let mfaCode2 = stringRequest("==> Wait for your authenticator to show a
 new 6-digit code, then enter it: ",
                                     minLength: 6)
        await adminRespondToAuthChallenge(cipClient: cipClient, userName:
 userName,
                                          clientId: clientId, userPoolId: poolId,
                                          mfaCode: mfaCode2, session: session2)
    }
}

/// The program's asynchronous entry point.
@main
struct Main {
    static func main() async {
        let args = Array(CommandLine.arguments.dropFirst())

        do {
            let command = try ExampleCommand.parse(args)
            try await command.runAsync()
        } catch {
            ExampleCommand.exit(withError: error)
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API SDK AWS per Swift.

- AdminGetUser
- AdminInitiateAuth
- AdminRespondToAuthChallenge
- AssociateSoftwareToken
- ConfirmDevice
- ConfirmSignUp
- InitiateAuth
- ListUsers
- ResendConfirmationCode
- RespondToAuthChallenge
- SignUp
- VerifySoftwareToken

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.

Usa i pool di identità e i flussi di autenticazione di Amazon Cognito

Il seguente esempio di codice mostra come creare un'applicazione demo basata sul Web che dimostri i flussi di autenticazione dei pool di identità.

Python

SDK per Python (Boto3)

Mostra un'applicazione demo basata sul Web che illustra i flussi di autenticazione dei pool di identità di Amazon Cognito, che consente agli utenti di esplorare in modo interattivo i flussi di autenticazione avanzati e di base con vari provider di identità.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Gestore dell'identità per Amazon Cognito

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.

Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito utilizzando un AWS SDK

L'esempio di codice seguente mostra come scrivere dati di attività personalizzate con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito.

- Usa le funzioni di amministratore per aggiungere un utente a un pool di utenti.
- Configura un pool di utenti per chiamare una funzione Lambda per il trigger `PostAuthentication`.
- Accedere con il nuovo utente ad Amazon Cognito.
- La funzione Lambda scrive informazioni personalizzate nei CloudWatch log e in una tabella DynamoDB.
- Acquisisci e visualizza dati personalizzati dalla tabella DynamoDB, quindi elimina le risorse.

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.

Esegui uno scenario interattivo al prompt dei comandi.

```
import (
 "context"
 "errors"
 "log"
 "strings"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
```

```
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// ActivityLog separates the steps of this scenario into individual functions so
 that
// they are simpler to read and understand.
type ActivityLog struct {
 helper       IScenarioHelper
 questioner   demotools.IQuestioner
 resources    Resources
 cognitoActor *actions.CognitoActions
}

// NewActivityLog constructs a new activity log runner.
func NewActivityLog(sdkConfig aws.Config, questioner demotools.IQuestioner,
 helper IScenarioHelper) ActivityLog {
 scenario := ActivityLog{
  helper:       helper,
  questioner:   questioner,
  resources:    Resources{},
  cognitoActor: &actions.CognitoActions{CognitoClient:
 cognitoidentityprovider.NewFromConfig(sdkConfig)},
 }
 scenario.resources.init(scenario.cognitoActor, questioner)
 return scenario
}

// AddUserToPool selects a user from the known users table and uses administrator
 credentials to add the user to the user pool.
func (runner *ActivityLog) AddUserToPool(ctx context.Context, userPoolId string,
 tableName string) (string, string) {
 log.Println("To facilitate this example, let's add a user to the user pool using
 administrator privileges.")
 users, err := runner.helper.GetKnownUsers(ctx, tableName)
 if err != nil {
  panic(err)
 }
 user := users.Users[0]
 log.Printf("Adding known user %v to the user pool.\n", user.UserName)
 err = runner.cognitoActor.AdminCreateUser(ctx, userPoolId, user.UserName,
 user.UserEmail)
 if err != nil {
```

```
  panic(err)
 }
 pwSet := false
 password := runner.questioner.AskPassword("\nEnter a password that has at least
 eight characters, uppercase, lowercase, numbers and symbols.\n"+
  "(the password will not display as you type):", 8)
 for !pwSet {
  log.Printf("\nSetting password for user '%v'.\n", user.UserName)
  err = runner.cognitoActor.AdminSetUserPassword(ctx, userPoolId, user.UserName,
 password)
  if err != nil {
   var invalidPassword *types.InvalidPasswordException
   if errors.As(err, &invalidPassword) {
    password = runner.questioner.AskPassword("\nEnter another password:", 8)
   } else {
    panic(err)
   }
  } else {
   pwSet = true
  }
 }

 log.Println(strings.Repeat("-", 88))

 return user.UserName, password
}

// AddActivityLogTrigger adds a Lambda handler as an invocation target for the
 PostAuthentication trigger.
func (runner *ActivityLog) AddActivityLogTrigger(ctx context.Context, userPoolId
 string, activityLogArn string) {
 log.Println("Let's add a Lambda function to handle the PostAuthentication
 trigger from Cognito.\n" +
  "This trigger happens after a user is authenticated, and lets your function
 take action, such as logging\n" +
  "the outcome.")
 err := runner.cognitoActor.UpdateTriggers(
  ctx, userPoolId,
  actions.TriggerInfo{Trigger: actions.PostAuthentication, HandlerArn:
 aws.String(activityLogArn)})
 if err != nil {
  panic(err)
 }
```

```
 runner.resources.triggers = append(runner.resources.triggers,
 actions.PostAuthentication)
 log.Printf("Lambda function %v added to user pool %v to handle
 PostAuthentication Cognito trigger.\n",
  activityLogArn, userPoolId)

 log.Println(strings.Repeat("-", 88))
}

// SignInUser signs in as the specified user.
func (runner *ActivityLog) SignInUser(ctx context.Context, clientId string,
 userName string, password string) {
 log.Printf("Now we'll sign in user %v and check the results in the logs and the
 DynamoDB table.", userName)
 runner.questioner.Ask("Press Enter when you're ready.")
 authResult, err := runner.cognitoActor.SignIn(ctx, clientId, userName, password)
 if err != nil {
  panic(err)
 }
 log.Println("Sign in successful.",
  "The PostAuthentication Lambda handler writes custom information to CloudWatch
 Logs.")

 runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
 *authResult.AccessToken)
}

// GetKnownUserLastLogin gets the login info for a user from the Amazon DynamoDB
 table and displays it.
func (runner *ActivityLog) GetKnownUserLastLogin(ctx context.Context, tableName
 string, userName string) {
 log.Println("The PostAuthentication handler also writes login data to the
 DynamoDB table.")
 runner.questioner.Ask("Press Enter when you're ready to continue.")
 users, err := runner.helper.GetKnownUsers(ctx, tableName)
 if err != nil {
  panic(err)
 }
 for _, user := range users.Users {
  if user.UserName == userName {
   log.Println("The last login info for the user in the known users table is:")
   log.Printf("\t%+v", *user.LastLogin)
  }
 }
```

```
 log.Println(strings.Repeat("-", 88))
}

// Run runs the scenario.
func (runner *ActivityLog) Run(ctx context.Context, stackName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   runner.resources.Cleanup(ctx)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Printf("Welcome\n")

 log.Println(strings.Repeat("-", 88))

 stackOutputs, err := runner.helper.GetStackOutputs(ctx, stackName)
 if err != nil {
  panic(err)
 }
 runner.resources.userPoolId = stackOutputs["UserPoolId"]
 runner.helper.PopulateUserTable(ctx, stackOutputs["TableName"])
 userName, password := runner.AddUserToPool(ctx, stackOutputs["UserPoolId"],
 stackOutputs["TableName"])

 runner.AddActivityLogTrigger(ctx, stackOutputs["UserPoolId"],
 stackOutputs["ActivityLogFunctionArn"])
 runner.SignInUser(ctx, stackOutputs["UserPoolClientId"], userName, password)
 runner.helper.ListRecentLogEvents(ctx, stackOutputs["ActivityLogFunction"])
 runner.GetKnownUserLastLogin(ctx, stackOutputs["TableName"], userName)

 runner.resources.Cleanup(ctx)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Gestisci il trigger `PostAuthentication` con una funzione Lambda.

```
import (
 "context"
 "fmt"
 "log"
 "os"
 "time"

 "github.com/aws/aws-lambda-go/events"
 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 dynamodbtypes "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

const TABLE_NAME = "TABLE_NAME"

// LoginInfo defines structured login data that can be marshalled to a DynamoDB
 format.
type LoginInfo struct {
 UserPoolId string `dynamodbav:"UserPoolId"`
 ClientId   string `dynamodbav:"ClientId"`
 Time       string `dynamodbav:"Time"`
}

// UserInfo defines structured user data that can be marshalled to a DynamoDB
 format.
type UserInfo struct {
 UserName  string    `dynamodbav:"UserName"`
 UserEmail string    `dynamodbav:"UserEmail"`
 LastLogin LoginInfo `dynamodbav:"LastLogin"`
}

// GetKey marshals the user email value to a DynamoDB key format.
func (user UserInfo) GetKey() map[string]dynamodbtypes.AttributeValue {
 userEmail, err := attributevalue.Marshal(user.UserEmail)
 if err != nil {
  panic(err)
 }
 return map[string]dynamodbtypes.AttributeValue{"UserEmail": userEmail}
}
```

```
type handler struct {
 dynamoClient *dynamodb.Client
}

// HandleRequest handles the PostAuthentication event by writing custom data to
 the logs and
// to an Amazon DynamoDB table.
func (h *handler) HandleRequest(ctx context.Context,
 event events.CognitoEventUserPoolsPostAuthentication)
 (events.CognitoEventUserPoolsPostAuthentication, error) {
 log.Printf("Received post authentication trigger from %v for user '%v'",
 event.TriggerSource, event.UserName)
 tableName := os.Getenv(TABLE_NAME)
 user := UserInfo{
  UserName:  event.UserName,
  UserEmail: event.Request.UserAttributes["email"],
  LastLogin: LoginInfo{
   UserPoolId: event.UserPoolID,
   ClientId:   event.CallerContext.ClientID,
   Time:       time.Now().Format(time.UnixDate),
  },
 }
 // Write to CloudWatch Logs.
 fmt.Printf("%#v", user)

 // Also write to an external system. This examples uses DynamoDB to demonstrate.
 userMap, err := attributevalue.MarshalMap(user)
 if err != nil {
  log.Printf("Couldn't marshal to DynamoDB map. Here's why: %v\n", err)
 } else if len(userMap) == 0 {
  log.Printf("User info marshaled to an empty map.")
 } else {
  _, err := h.dynamoClient.PutItem(ctx, &dynamodb.PutItemInput{
   Item:      userMap,
   TableName: aws.String(tableName),
  })
  if err != nil {
   log.Printf("Couldn't write to DynamoDB. Here's why: %v\n", err)
  } else {
   log.Printf("Wrote user info to DynamoDB table %v.\n", tableName)
  }
 }
```

```
 return event, nil
}

func main() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 if err != nil {
  log.Panicln(err)
 }
 h := handler{
  dynamoClient: dynamodb.NewFromConfig(sdkConfig),
 }
 lambda.Start(h.HandleRequest)
}
```

Crea una struttura che esegue operazioni comuni.

```
import (
 "context"
 "log"
 "strings"
 "time"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// IScenarioHelper defines common functions used by the workflows in this
 example.
type IScenarioHelper interface {
 Pause(secs int)
 GetStackOutputs(ctx context.Context, stackName string) (actions.StackOutputs,
 error)
 PopulateUserTable(ctx context.Context, tableName string)
 GetKnownUsers(ctx context.Context, tableName string) (actions.UserList, error)
 AddKnownUser(ctx context.Context, tableName string, user actions.User)
```

```
 ListRecentLogEvents(ctx context.Context, functionName string)
}

// ScenarioHelper contains AWS wrapper structs used by the workflows in this
 example.
type ScenarioHelper struct {
 questioner  demotools.IQuestioner
 dynamoActor *actions.DynamoActions
 cfnActor    *actions.CloudFormationActions
 cwlActor    *actions.CloudWatchLogsActions
 isTestRun   bool
}

// NewScenarioHelper constructs a new scenario helper.
func NewScenarioHelper(sdkConfig aws.Config, questioner demotools.IQuestioner)
 ScenarioHelper {
 scenario := ScenarioHelper{
  questioner:  questioner,
  dynamoActor: &actions.DynamoActions{DynamoClient:
 dynamodb.NewFromConfig(sdkConfig)},
  cfnActor:    &actions.CloudFormationActions{CfnClient:
 cloudformation.NewFromConfig(sdkConfig)},
  cwlActor:    &actions.CloudWatchLogsActions{CwlClient:
 cloudwatchlogs.NewFromConfig(sdkConfig)},
 }
 return scenario
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
 if !helper.isTestRun {
  time.Sleep(time.Duration(secs) * time.Second)
 }
}

// GetStackOutputs gets the outputs from the specified CloudFormation stack in a
 structured format.
func (helper ScenarioHelper) GetStackOutputs(ctx context.Context, stackName
 string) (actions.StackOutputs, error) {
 return helper.cfnActor.GetOutputs(ctx, stackName), nil
}

// PopulateUserTable fills the known user table with example data.
```

```
func (helper ScenarioHelper) PopulateUserTable(ctx context.Context, tableName
 string) {
 log.Printf("First, let's add some users to the DynamoDB %v table we'll use for
 this example.\n", tableName)
 err := helper.dynamoActor.PopulateTable(ctx, tableName)
 if err != nil {
  panic(err)
 }
}

// GetKnownUsers gets the users from the known users table in a structured
 format.
func (helper ScenarioHelper) GetKnownUsers(ctx context.Context, tableName string)
 (actions.UserList, error) {
 knownUsers, err := helper.dynamoActor.Scan(ctx, tableName)
 if err != nil {
  log.Printf("Couldn't get known users from table %v. Here's why: %v\n",
 tableName, err)
 }
 return knownUsers, err
}

// AddKnownUser adds a user to the known users table.
func (helper ScenarioHelper) AddKnownUser(ctx context.Context, tableName string,
 user actions.User) {
 log.Printf("Adding user '%v' with email '%v' to the DynamoDB known users
 table...\n",
  user.UserName, user.UserEmail)
 err := helper.dynamoActor.AddUser(ctx, tableName, user)
 if err != nil {
  panic(err)
 }
}

// ListRecentLogEvents gets the most recent log stream and events for the
 specified Lambda function and displays them.
func (helper ScenarioHelper) ListRecentLogEvents(ctx context.Context,
 functionName string) {
 log.Println("Waiting a few seconds to let Lambda write to CloudWatch Logs...")
 helper.Pause(10)
 log.Println("Okay, let's check the logs to find what's happened recently with
 your Lambda function.")
 logStream, err := helper.cwlActor.GetLatestLogStream(ctx, functionName)
 if err != nil {
```

```
  panic(err)
 }
 log.Printf("Getting some recent events from log stream %v\n",
 *logStream.LogStreamName)
 events, err := helper.cwlActor.GetLogEvents(ctx, functionName,
 *logStream.LogStreamName, 10)
 if err != nil {
  panic(err)
 }
 for _, event := range events {
  log.Printf("\t%v", *event.Message)
 }
 log.Println(strings.Repeat("-", 88))
}
```

Crea una struttura che racchiude le azioni di Amazon Cognito.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
 trigger.
type Trigger int

const (
 PreSignUp Trigger = iota
 UserMigration
```

```
 PostAuthentication
)

type TriggerInfo struct {
 Trigger    Trigger
 HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
 is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(ctx context.Context, userPoolId
 string, triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(ctx,
 &cognitoidentityprovider.DescribeUserPoolInput{
  UserPoolId: aws.String(userPoolId),
 })
 if err != nil {
  log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
 userPoolId, err)
  return err
 }
 lambdaConfig := output.UserPool.LambdaConfig
 for _, trigger := range triggers {
  switch trigger.Trigger {
  case PreSignUp:
   lambdaConfig.PreSignUp = trigger.HandlerArn
  case UserMigration:
   lambdaConfig.UserMigration = trigger.HandlerArn
  case PostAuthentication:
   lambdaConfig.PostAuthentication = trigger.HandlerArn
  }
 }
 _, err = actor.CognitoClient.UpdateUserPool(ctx,
 &cognitoidentityprovider.UpdateUserPoolInput{
  UserPoolId:   aws.String(userPoolId),
  LambdaConfig: lambdaConfig,
 })
 if err != nil {
  log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
 }
 return err
}
```

```
// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(ctx context.Context, clientId string, userName
 string, password string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(ctx,
 &cognitoidentityprovider.SignUpInput{
  ClientId: aws.String(clientId),
  Password: aws.String(password),
  Username: aws.String(userName),
  UserAttributes: []types.AttributeType{
   {Name: aws.String("email"), Value: aws.String(userEmail)},
  },
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
  }
 } else {
  confirmed = output.UserConfirmed
 }
 return confirmed, err
}



// SignIn signs in a user to Amazon Cognito using a username and password
 authentication flow.
func (actor CognitoActions) SignIn(ctx context.Context, clientId string, userName
 string, password string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(ctx,
 &cognitoidentityprovider.InitiateAuthInput{
  AuthFlow:       "USER_PASSWORD_AUTH",
  ClientId:       aws.String(clientId),
  AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
 })
 if err != nil {
  var resetRequired *types.PasswordResetRequiredException
  if errors.As(err, &resetRequired) {
```

```
   log.Println(*resetRequired.Message)
  } else {
   log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
  }
 } else {
  authResult = output.AuthenticationResult
 }
 return authResult, err
}



// ForgotPassword starts a password recovery flow for a user. This flow typically
 sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(ctx context.Context, clientId string,
 userName string) (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(ctx,
 &cognitoidentityprovider.ForgotPasswordInput{
  ClientId: aws.String(clientId),
  Username: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't start password reset for user '%v'. Here;s why: %v\n",
 userName, err)
 }
 return output.CodeDeliveryDetails, err
}



// ConfirmForgotPassword confirms a user with a confirmation code and a new
 password.
func (actor CognitoActions) ConfirmForgotPassword(ctx context.Context, clientId
 string, code string, userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(ctx,
 &cognitoidentityprovider.ConfirmForgotPasswordInput{
  ClientId:         aws.String(clientId),
  ConfirmationCode: aws.String(code),
  Password:         aws.String(password),
  Username:         aws.String(userName),
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
```

```
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
  }
 }
 return err
}



// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(ctx context.Context, userAccessToken
 string) error {
 _, err := actor.CognitoClient.DeleteUser(ctx,
 &cognitoidentityprovider.DeleteUserInput{
  AccessToken: aws.String(userAccessToken),
 })
 if err != nil {
  log.Printf("Couldn't delete user. Here's why: %v\n", err)
 }
 return err
}



// AdminCreateUser uses administrator credentials to add a user to a user pool.
 This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(ctx context.Context, userPoolId
 string, userName string, userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(ctx,
 &cognitoidentityprovider.AdminCreateUserInput{
  UserPoolId:     aws.String(userPoolId),
  Username:       aws.String(userName),
  MessageAction:  types.MessageActionTypeSuppress,
  UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
 aws.String(userEmail)}},
 })
 if err != nil {
  var userExists *types.UsernameExistsException
  if errors.As(err, &userExists) {
   log.Printf("User %v already exists in the user pool.", userName)
   err = nil
```

```
  } else {
   log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
  }
 }
 return err
}



// AdminSetUserPassword uses administrator credentials to set a password for a
 user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(ctx context.Context, userPoolId
 string, userName string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(ctx,
 &cognitoidentityprovider.AdminSetUserPasswordInput{
  Password:   aws.String(password),
  UserPoolId: aws.String(userPoolId),
  Username:   aws.String(userName),
  Permanent:  true,
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
 err)
  }
 }
 return err
}
```

Crea una struttura che racchiude le azioni di DynamoDB.

```
import (
 "context"
 "fmt"
 "log"
```

```
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

// DynamoActions encapsulates the Amazon Simple Notification Service (Amazon SNS)
 actions
// used in the examples.
type DynamoActions struct {
 DynamoClient *dynamodb.Client
}

// User defines structured user data.
type User struct {
 UserName  string
 UserEmail string
 LastLogin *LoginInfo `dynamodbav:",omitempty"`
}

// LoginInfo defines structured custom login data.
type LoginInfo struct {
 UserPoolId string
 ClientId   string
 Time       string
}

// UserList defines a list of users.
type UserList struct {
 Users []User
}

// UserNameList returns the usernames contained in a UserList as a list of
 strings.
func (users *UserList) UserNameList() []string {
 names := make([]string, len(users.Users))
 for i := 0; i < len(users.Users); i++ {
  names[i] = users.Users[i].UserName
 }
 return names
}

// PopulateTable adds a set of test users to the table.
```

```
func (actor DynamoActions) PopulateTable(ctx context.Context, tableName string)
 error {
 var err error
 var item map[string]types.AttributeValue
 var writeReqs []types.WriteRequest
 for i := 1; i < 4; i++ {
  item, err = attributevalue.MarshalMap(User{UserName: fmt.Sprintf("test_user_
%v", i), UserEmail: fmt.Sprintf("test_email_%v@example.com", i)})
  if err != nil {
   log.Printf("Couldn't marshall user into DynamoDB format. Here's why: %v\n",
 err)
   return err
  }
  writeReqs = append(writeReqs, types.WriteRequest{PutRequest:
 &types.PutRequest{Item: item}})
 }
 _, err = actor.DynamoClient.BatchWriteItem(ctx, &dynamodb.BatchWriteItemInput{
  RequestItems: map[string][]types.WriteRequest{tableName: writeReqs},
 })
 if err != nil {
  log.Printf("Couldn't populate table %v with users. Here's why: %v\n",
 tableName, err)
 }
 return err
}

// Scan scans the table for all items.
func (actor DynamoActions) Scan(ctx context.Context, tableName string) (UserList,
 error) {
 var userList UserList
 output, err := actor.DynamoClient.Scan(ctx, &dynamodb.ScanInput{
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't scan table %v for items. Here's why: %v\n", tableName,
 err)
 } else {
  err = attributevalue.UnmarshalListOfMaps(output.Items, &userList.Users)
  if err != nil {
   log.Printf("Couldn't unmarshal items into users. Here's why: %v\n", err)
  }
 }
 return userList, err
}
```

```
// AddUser adds a user item to a table.
func (actor DynamoActions) AddUser(ctx context.Context, tableName string, user
 User) error {
 userItem, err := attributevalue.MarshalMap(user)
 if err != nil {
  log.Printf("Couldn't marshall user to item. Here's why: %v\n", err)
 }
 _, err = actor.DynamoClient.PutItem(ctx, &dynamodb.PutItemInput{
  Item:      userItem,
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't put item in table %v. Here's why: %v", tableName, err)
 }
 return err
}
```

Crea una struttura che racchiuda le azioni di CloudWatch Logs.

```
import (
 "context"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs/types"
)

type CloudWatchLogsActions struct {
 CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(ctx context.Context,
 functionName string) (types.LogStream, error) {
 var logStream types.LogStream
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
```

```
 output, err := actor.CwlClient.DescribeLogStreams(ctx,
 &cloudwatchlogs.DescribeLogStreamsInput{
  Descending:   aws.Bool(true),
  Limit:        aws.Int32(1),
  LogGroupName: aws.String(logGroupName),
  OrderBy:      types.OrderByLastEventTime,
 })
 if err != nil {
  log.Printf("Couldn't get log streams for log group %v. Here's why: %v\n",
 logGroupName, err)
 } else {
  logStream = output.LogStreams[0]
 }
 return logStream, err
}

// GetLogEvents gets the most recent eventCount events from the specified log
 stream.
func (actor CloudWatchLogsActions) GetLogEvents(ctx context.Context, functionName
 string, logStreamName string, eventCount int32) (
 []types.OutputLogEvent, error) {
 var events []types.OutputLogEvent
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.GetLogEvents(ctx,
 &cloudwatchlogs.GetLogEventsInput{
  LogStreamName: aws.String(logStreamName),
  Limit:         aws.Int32(eventCount),
  LogGroupName:  aws.String(logGroupName),
 })
 if err != nil {
  log.Printf("Couldn't get log event for log stream %v. Here's why: %v\n",
 logStreamName, err)
 } else {
  events = output.Events
 }
 return events, err
}
```

Crea una struttura che racchiuda le azioni. CloudFormation

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
)

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
 CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
 structured format.
func (actor CloudFormationActions) GetOutputs(ctx context.Context, stackName
 string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(ctx,
 &cloudformation.DescribeStacksInput{
  StackName: aws.String(stackName),
 })
 if err != nil || len(output.Stacks) == 0 {
  log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
 stackName, err)
 }
 stackOutputs := StackOutputs{}
 for _, out := range output.Stacks[0].Outputs {
  stackOutputs[*out.OutputKey] = *out.OutputValue
 }
 return stackOutputs
}
```

Eliminare le risorse.

```
import (
 "context"
 "log"
 "user_pools_and_lambda_triggers/actions"
```

```
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// Resources keeps track of AWS resources created during an example and handles
// cleanup when the example finishes.
type Resources struct {
 userPoolId       string
 userAccessTokens []string
 triggers         []actions.Trigger

 cognitoActor *actions.CognitoActions
 questioner   demotools.IQuestioner
}

func (resources *Resources) init(cognitoActor *actions.CognitoActions, questioner
 demotools.IQuestioner) {
 resources.userAccessTokens = []string{}
 resources.triggers = []actions.Trigger{}
 resources.cognitoActor = cognitoActor
 resources.questioner = questioner
}

// Cleanup deletes all AWS resources created during an example.
func (resources *Resources) Cleanup(ctx context.Context) {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong during cleanup.\n%v\n", r)
   log.Println("Use the AWS Management Console to remove any remaining resources
 \n" +
    "that were created for this scenario.")
  }
 }()

 wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
 resources that were created "+
  "during this demo (y/n)?", "y")
 if wantDelete {
  for _, accessToken := range resources.userAccessTokens {
   err := resources.cognitoActor.DeleteUser(ctx, accessToken)
   if err != nil {
    log.Println("Couldn't delete user during cleanup.")
    panic(err)
   }
```

```
   log.Println("Deleted user.")
  }
  triggerList := make([]actions.TriggerInfo, len(resources.triggers))
  for i := 0; i < len(resources.triggers); i++ {
   triggerList[i] = actions.TriggerInfo{Trigger: resources.triggers[i],
 HandlerArn: nil}
  }
  err := resources.cognitoActor.UpdateTriggers(ctx, resources.userPoolId,
 triggerList...)
  if err != nil {
   log.Println("Couldn't update Cognito triggers during cleanup.")
   panic(err)
  }
  log.Println("Removed Cognito triggers from user pool.")
 } else {
  log.Println("Be sure to remove resources when you're done with them to avoid
 unexpected charges!")
 }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Go .
  - AdminCreateUser
  - AdminSetUserPassword
  - DeleteUser
  - InitiateAuth
  - UpdateUserPool

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 per l'utilizzo di Amazon Cognito Sync AWS SDKs

I seguenti esempi di codice mostrano come usare Amazon Cognito Sync 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.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Amazon Cognito Sync

- Esempi di base per l'utilizzo di Amazon Cognito Sync AWS SDKs
  - Azioni per Amazon Cognito Sync utilizzando AWS SDKs
    - Utilizzalo ListIdentityPoolUsage con un AWS SDK

Esempi di base per l'utilizzo di Amazon Cognito Sync AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di Amazon Cognito Sync con. AWS SDKs

Esempi

- Azioni per Amazon Cognito Sync utilizzando AWS SDKs
  - Utilizzalo ListIdentityPoolUsage con un AWS SDK

Azioni per Amazon Cognito Sync utilizzando AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole azioni di Amazon Cognito Sync con. AWS SDKs Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le azioni più comunemente utilizzate. Per l'elenco completo, consulta la documentazione di riferimento dell'API Amazon Cognito Sync.

Esempi

- Utilizzalo ListIdentityPoolUsage con un AWS SDK

Utilizzalo **`ListIdentityPoolUsage`** con un AWS SDK

Il seguente esempio di codice mostra come utilizzare`ListIdentityPoolUsage`.

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 show_pools(client: &Client) -> Result<(), Error> {
    let response = client
        .list_identity_pool_usage()
        .max_results(10)
        .send()
        .await?;

    let pools = response.identity_pool_usages();
    println!("Identity pools:");

    for pool in pools {
        println!(
            "  Identity pool ID:    {}",
            pool.identity_pool_id().unwrap_or_default()
        );
        println!(
            "  Data storage:        {}",
            pool.data_storage().unwrap_or_default()
        );
        println!(
            "  Sync sessions count: {}",
            pool.sync_sessions_count().unwrap_or_default()
        );
        println!(
            "  Last modified:       {}",
            pool.last_modified_date().unwrap().to_chrono_utc()?
        );
        println!();
    }

    println!("Next token: {}", response.next_token().unwrap_or_default());
```

```
    Ok(())
}
```

- Per i dettagli sulle API, consulta la ListIdentityPoolUsageguida 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.

Altre funzionalità

Oltre a far accedere gli utenti alla tua applicazione con un pool di utenti, Amazon Cognito offre funzionalità aggiuntive per esigenze specifiche. I pool di identità di Amazon Cognito scambiano l'identità di un utente con AWS credenziali temporanee con privilegi limitati in modo che l'applicazione possa chiamare direttamente i servizi. AWS

Argomenti

- Pool di identità di Amazon Cognito
- Guida introduttiva ai pool di identità di Amazon Cognito

Pool di identità di Amazon Cognito

Un pool di identità di Amazon Cognito è una directory di identità federate che puoi scambiare per credenziali AWS . I pool di identità generano AWS credenziali temporanee per gli utenti della tua app, indipendentemente dal fatto che abbiano effettuato l'accesso o che tu non li abbia ancora identificati. Con i ruoli e le policy AWS Identity and Access Management (IAM), puoi scegliere il livello di autorizzazione che desideri concedere ai tuoi utenti. Gli utenti possono iniziare come ospiti e recuperare le risorse mantenute nei Servizi AWS. Quindi, possono accedere con un provider di identità di terze parti per sbloccare l'accesso alle risorse che vengono rese disponibili ai membri registrati. Il provider di identità di terze parti può essere un provider OAuth 2.0 utente (social) come Apple o Google, un provider di identità SAML o OIDC personalizzato o uno schema di autenticazione personalizzato, chiamato anche provider degli sviluppatori, di propria progettazione.

Funzionalità del pool di identità di Amazon Cognito

Firma le richieste per Servizi AWS

> Firma le richieste API su Amazon Simple Storage Service (Amazon S3) e Amazon DynamoDB. Servizi AWS Analizza l'attività degli utenti con servizi come Amazon Pinpoint e Amazon. CloudWatch

Filtrare richieste con policy basate su risorse

> Esercita controllo granulare sull'accesso utente alle risorse. Trasforma le richieste dell'utente in tag della sessione IAM e crea policy IAM che concedono l'accesso alle risorse a sottoinsiemi distinti di utenti.

Assegnare l'accesso guest

Per gli utenti che non hanno ancora effettuato l'accesso, configura il pool di identità per generare credenziali AWS con un ambito di accesso ristretto. Autentica gli utenti tramite un provider Single Sign-On per migliorarne l'accesso.

Assegnare ruoli IAM in base alle caratteristiche degli utenti

Assegna un singolo ruolo IAM a tutti gli utenti autenticati o scegli il ruolo in base alle richieste di ciascun utente.

Accettare un'ampia gamma di provider di identità

Scambia un ID o un token di accesso, un token del pool di utenti, un'asserzione SAML o un token OAuth di un social provider con credenziali. AWS

Convalidare le proprie identità

Esegui la convalida degli utenti e utilizza le tue credenziali di sviluppatore AWS per emettere credenziali per i tuoi utenti.

Potrebbe essere già disponibile un pool di utenti Amazon Cognito che fornisce servizi di autenticazione e autorizzazione all'app. Puoi configurare il pool di utenti come gestore dell'identità digitale per il pool di identità. Quando lo fai, i tuoi utenti possono autenticarsi tramite il tuo pool di utenti IdPs, consolidare le loro affermazioni in un token di identità OIDC comune e scambiare quel token con credenziali. AWS L'utente può quindi presentare le proprie credenziali in una richiesta firmata ai Servizi AWS.

Puoi anche presentare richieste autenticate da uno qualsiasi dei provider di identità direttamente al pool di identità. Amazon Cognito personalizza le dichiarazioni degli utenti dei provider SAML, OAuth e OIDC in una richiesta API per credenziali a breve termine. AssumeRoleWithWebIdentity

I pool di utenti di Amazon Cognito sono come provider di identità OIDC per le tue app. SSO-enabled I pool di identità fungono da provider di identità AWS per qualsiasi app con dipendenze di risorse che funzionano meglio con l'autorizzazione IAM.

I pool di identità di Amazon Cognito supportano i seguenti provider di identità:

- Provider pubblici: Configurazione di Login with Amazon come IdP di pool di identità, Configurazione di Facebook come pool di identità IdP, Configurazione di Google come IdP del pool di identità, Configurazione di Accedi con Apple come IdP del pool di identità, Twitter.
- Pool di utenti Amazon Cognito

- Configurazione di un provider OIDC come IdP del pool di identità
- Configurazione di un provider SAML come IdP del pool di identità
- Developer-authenticated identità

Per informazioni sulla disponibilità regionale dei pool di identità di Amazon Cognito, consulta Disponibilità delle regioni del servizio AWS.

Per ulteriori informazioni sui pool di identità di Amazon Cognito, consulta gli argomenti riportati di seguito.

Argomenti

- Panoramica della console dei pool di identità
- Flusso di autenticazione dei pool di identità
- Ruoli IAM
- Best practice di sicurezza per i pool di identità di Amazon Cognito
- Utilizzo di attributi per il controllo degli accessi
- Utilizzo del controllo degli accessi basato su ruoli
- Ottenere le credenziali
- Accesso Servizi AWS con credenziali temporanee
- Pool di identità, provider di identità di terze parti
- Developer-authenticated identità
- Passaggio da utenti non autenticati a utenti autenticati

Panoramica della console dei pool di identità

I pool di identità di Amazon Cognito forniscono AWS credenziali temporanee per gli utenti ospiti (non autenticati) e per gli utenti che sono stati autenticati e hanno ricevuto un token. Un pool di identità è un archivio di identificatori utente collegati ai tuoi provider di identità esterni.

Un modo per comprendere le caratteristiche e le opzioni dei pool di identità consiste nel crearne uno nella console Amazon Cognito. Puoi esplorare l'effetto di diverse impostazioni sui flussi di autenticazione, sul controllo degli accessi basato su ruoli e attributi e sull'accesso degli ospiti. Da qui, puoi passare ai capitoli successivi di questa guida e aggiungere i componenti appropriati all'applicazione in modo da poter implementare l'autenticazione del pool di identità.

Argomenti

- Crea un pool di identità di
- Ruoli IAM dell'utente
- Identità autenticate e non autenticate
- Attivazione o disattivazione dell'accesso guest
- Modifica del ruolo associato a un tipo di identità
- Modifica dei provider di identità
- Eliminazione di un pool di identità
- Eliminazione di un'identità da un pool di identità

Crea un pool di identità di

Per creare un nuovo pool di identità nella console

1. Accedi alla console di Amazon Cognito e seleziona Pool di identità.
2. Scegli Crea pool di identità.
3. In Configurazione dell'attendibilità del pool di identità, scegli di configurare il pool di identità per Accesso autenticato, Accesso guest o entrambi.

    - Se hai scelto Accesso autenticato, seleziona uno o più Tipi di identità che desideri impostare come origine delle identità autenticate nel pool di identità. Se configuri un Provider degli sviluppatori personalizzato, non puoi modificarlo né eliminarlo dopo aver creato il pool di identità.
4. In Configura le autorizzazioni, scegli un ruolo IAM predefinito per gli utenti autenticati o guest nel pool di identità.

    a. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza documento di policy per esaminare le autorizzazioni che Amazon Cognito assegna al tuo nuovo ruolo IAM.

    b. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo Account AWS che desideri utilizzare. Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente

che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.

5. In Connect identity providers, inserisci i dettagli dei provider di identità (IdPs) che hai scelto in Configure identity pool trust. È possibile che ti venga chiesto di fornire informazioni sul client dell'app OAuth, scegliere un pool di utenti Amazon Cognito, scegliere un IdP IAM o inserire un identificatore personalizzato per un provider degli sviluppatori.

   a. Scegli le impostazioni del ruolo per ogni IdP. Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure puoi selezionare l'opzione Scegli ruolo con regole. Con un IdP del pool di utenti Amazon Cognito, puoi anche scegliere un ruolo con preferred_role nei token. Per ulteriori informazioni sulla richiesta `cognito:preferred_role`, consultare Assegnazione dei valori di priorità ai gruppi.

      i. Se hai scelto Scegli ruolo con regole, inserisci la dichiarazione di origine ricavata dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare l'attestazione, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

      ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

   b. Configura Attributi per il controllo degli accessi per ciascun IdP. L'opzione Attributi per il controllo degli accessi associa le richieste dell'utente ai tag principali applicati da Amazon Cognito alla relativa sessione temporanea. Puoi creare policy IAM per filtrare l'accesso utente in base ai tag applicati alla relativa sessione.

      i. Per non applicare alcun tag principale, scegli Inattivo.

      ii. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

      iii. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

6. In Configura proprietà, inserisci un Nome in Nome del pool di identità.

7. In Autenticazione di base (classica), scegli Attiva flusso di base, se desiderato. Con il flusso di base attivo, puoi ignorare le selezioni di ruolo che hai effettuato per te e chiamare direttamente

IdPs . AssumeRoleWithWebIdentity Per ulteriori informazioni, consulta Flusso di autenticazione dei pool di identità.

8. In Tag, scegli Aggiungi tag se desideri applicare tag al pool di identità.
9. In Esamina e crea, conferma le selezioni effettuate per il nuovo pool di identità. Seleziona Modifica per tornare alla procedura guidata e modificare le eventuali impostazioni. Al termine, seleziona Crea un pool di identità.

Ruoli IAM dell'utente

Un ruolo IAM definisce le autorizzazioni per gli utenti di accedere a AWS risorse, come. Amazon Cognito Sync Gli utenti della tua applicazione assumono i ruoli che crei. Puoi specificare diversi ruoli per utenti autenticati e non autenticati. Per ulteriori informazioni sui ruoli IAM, consulta Ruoli IAM.

Identità autenticate e non autenticate

I pool di identità di Amazon Cognito supportano sia le identità autenticate che le identità non autenticate. Le identità autenticate appartengono agli utenti autenticati da qualsiasi provider di identità supportato. Solitamente le identità non autenticate appartengono a utenti guest.

- Per configurare le identità autenticate con un provider di accesso pubblico, consulta Pool di identità, provider di identità di terze parti.
- Per configurare il tuo processo di autenticazione back-end, consulta. Developer-authenticated identità

Attivazione o disattivazione dell'accesso guest

L'accesso degli ospiti ai pool di identità di Amazon Cognito (identità non autenticate) fornisce un identificatore e AWS credenziali univoci per gli utenti che non si autenticano con un provider di identità. Se la tua applicazione consente agli utenti che non effettuano l'accesso, puoi attivare l'accesso per le identità non autenticate. Per ulteriori informazioni, consulta Guida introduttiva ai pool di identità di Amazon Cognito.

Per aggiornare l'accesso ospite in un pool di identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.

3. Individua Accesso guest. In un pool di identità che attualmente non supporta l'accesso guest, Stato è Inattivo.

   a. Se Accesso guest è Attivo e desideri disattivarlo, seleziona Disattiva.

   b. Se Accesso guest è Inattivo e desideri attivarlo, seleziona Modifica.

      - Scegli un ruolo IAM predefinito per gli utenti guest nel pool di identità.

        A. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza documento di policy per esaminare le autorizzazioni che Amazon Cognito assegna al tuo nuovo ruolo IAM.

        B. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo Account AWS che desideri utilizzare. Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.

        C. Seleziona Salva modifiche.

        D. Per attivare l'accesso guest, seleziona Attiva nella scheda Accesso utente.

Modifica del ruolo associato a un tipo di identità

Ogni identità nel pool di identità può essere autenticata o non autenticata. Le identità autenticate appartengono agli utenti autenticati da un provider di accesso pubblico (pool di utenti di Amazon Cognito, Login with Amazon, Accedi con Apple, Facebook, Google, SAML o qualsiasi provider OpenID Connect) o da un provider di sviluppatori (il tuo processo di autenticazione back-end). Solitamente le identità non autenticate appartengono a utenti guest.

Per ogni tipo di identità, esiste un ruolo assegnato. A questo ruolo è associata una politica che stabilisce a quale ruolo può accedere. Servizi AWS Quando Amazon Cognito riceve una richiesta, il servizio determina il tipo di identità e il ruolo assegnato a quel tipo di identità e utilizza la policy associata a tale ruolo per rispondere. Modificando una politica o assegnando un ruolo diverso a un

tipo di identità, puoi controllare a quale tipo Servizi AWS di identità può accedere. Per visualizzare o modificare le policy associate ai ruoli nel pool di identità, vedi la Console AWS IAM.

Per modificare il ruolo predefinito autenticato o non autenticato del pool di identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Individua Accesso guest o Accesso autenticato. In un pool di identità che non è attualmente configurato per tale tipo di accesso, Stato è Inattivo. Seleziona Edit (Modifica).
4. Scegli un ruolo IAM predefinito per gli utenti guest o autenticati nel pool di identità.

    a. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza documento di policy per esaminare le autorizzazioni che Amazon Cognito assegna al tuo nuovo ruolo IAM.

    b. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo Account AWS che desideri utilizzare. Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.
5. Seleziona Salva modifiche.

Modifica dei provider di identità

Se consenti agli utenti di effettuare l'autenticazione utilizzando i provider di identità utente (ad esempio pool di utenti Amazon Cognito, Login with Amazon, Accedi con Apple, Facebook o Google), puoi specificare gli identificatori dell'applicazione nella console dei pool di identità Amazon Cognito (identità federate). Questo associa l'ID dell'applicazione (fornito dal provider di accesso pubblico) al pool di identità.

Puoi anche configurare regole di autenticazione per ogni provider da questa pagina. Ogni provider consente fino a 25 regole. Le regole vengono applicate nell'ordine in cui le hai salvate per ogni provider. Per ulteriori informazioni, consulta Utilizzo del controllo degli accessi basato su ruoli.

> **Warning**
>
> La modifica dell'ID dell'applicazione IdP collegato nel pool di identità impedisce agli utenti esistenti di effettuare l'autenticazione con il pool di identità. Per ulteriori informazioni, consulta Pool di identità, provider di identità di terze parti.

Per aggiornare un gestore dell'identità digitale del pool di identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Individua Provider di identità. Scegli il provider di identità da modificare. Se desideri aggiungere un nuovo IdP, seleziona Aggiungi provider di identità.
   - Se hai scelto Aggiungi provider di identità, scegli uno dei Tipi di identità che desideri aggiungere.
4. Per modificare l'ID dell'applicazione, scegli Modifica in Informazioni sul provider di identità.
5. Per modificare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, scegli Modificain Impostazioni ruolo.
   - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure puoi selezionare l'opzione Scegli ruolo con regole. Con un IdP del pool di utenti Amazon Cognito, puoi anche scegliere un ruolo con preferred_role nei token. Per ulteriori informazioni sulla richiesta `cognito:preferred_role`, consultare Assegnazione dei valori di priorità ai gruppi.
     i. Se hai scelto Scegli ruolo con regole, inserisci il claim di origine tratto dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare il claim, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che vuoi assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.
     ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.
6. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, scegli Modifica in Attributi per il controllo degli accessi.

a. Per non applicare alcun tag principale, scegli Inattivo.

b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

7. Seleziona Salva modifiche.

Eliminazione di un pool di identità

L'eliminazione del pool di identità non può essere annullata. Dopo aver eliminato un pool di identità, tutte le app e gli utenti che dipendono da esso smettono di funzionare.

Eliminazione di un pool di identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona il pulsante di opzione accanto al pool di identità che desideri eliminare.
2. Seleziona Elimina.
3. Inserisci o incolla il nome del pool di identità e seleziona Elimina.

> **Warning**
>
> Selezionando il pulsante di eliminazione, eliminerai il pool di identità e tutti i dati utente che contiene. L'eliminazione di un pool di identità causa l'interruzione del funzionamento delle applicazioni e degli altri servizi che utilizzano il pool di identità.

Eliminazione di un'identità da un pool di identità

Quando si elimina un'identità da un pool di identità, vengono rimosse le informazioni di identificazione archiviate da Amazon Cognito per tale utente federato. Quando l'utente richiede nuovamente le credenziali, riceve un nuovo ID identità se il pool di identità considera ancora attendibile il provider di identità. Questa operazione non può essere annullata.

Per eliminare un'identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Scegli la scheda Browser di identità.
3. Seleziona le caselle di controllo accanto alle identità che desideri eliminare e scegli Elimina. Conferma che desideri eliminare le identità e scegli Elimina.

Flusso di autenticazione dei pool di identità

Amazon Cognito consente di creare identificatori univoci per i tuoi utenti finali che sono mantenuti coerenti su tutti i dispositivi e le piattaforme. Amazon Cognito fornisce anche credenziali temporanee con privilegi limitati alla tua applicazione per accedere alle risorse. AWS In questa pagina sono riportati i concetti di base del funzionamento dell'autenticazione in Amazon Cognito e viene spiegato il ciclo di vita di un'identità all'interno del tuo pool di identità.

Flusso di autenticazione del provider esterno

Un utente che effettua l'autenticazione con Amazon Cognito passa attraverso un processo multifase per eseguire il bootstrap delle proprie credenziali. Amazon Cognito dispone di due flussi differenti per l'autenticazione con provider pubblici: avanzati e di base.

Una volta completato uno di questi flussi, puoi accedere ad altri, Servizi AWS come definito dalle politiche di accesso del tuo ruolo. Di default, la console Amazon Cognito crea ruoli con accesso all'archivio di Amazon Cognito Sync e ad Amazon Mobile Analytics. Per ulteriori informazioni su come concedere un accesso aggiuntivo, consulta Ruoli IAM.

I pool di identità accettano i seguenti elementi forniti dai provider:

Provider	Artefatto di autenticazione
Bacino d'utenza di Amazon Cognito	Token ID
OpenID Connect (OIDC)	Token ID
SAML 2.0	Asserzione SAML
Fornitore di servizi sociali	Token di accesso

Il flusso di autenticazione migliorato (semplificato)

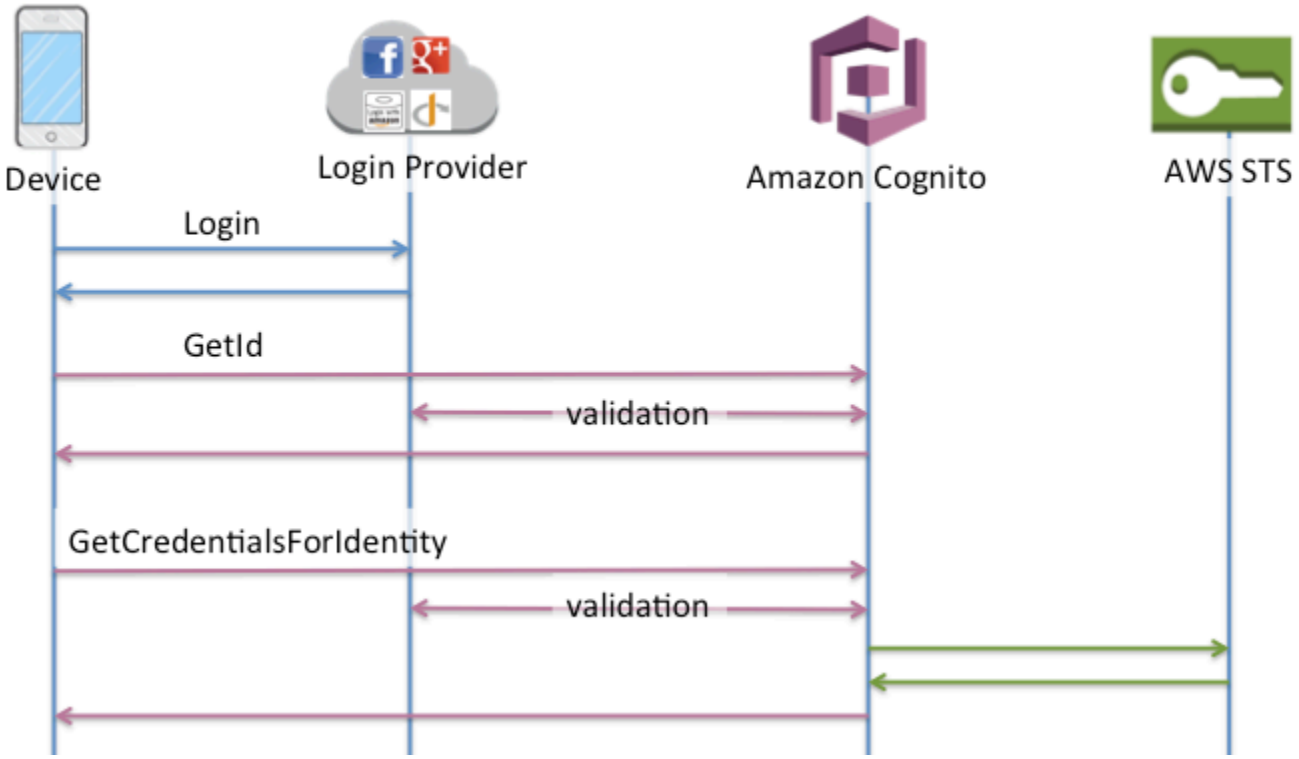
Quando utilizzi il flusso di autenticazione avanzato, l'app presenta innanzitutto una prova di autenticazione da un pool di utenti Amazon Cognito autorizzato o da un provider di identità di terze parti in GetIduna richiesta.

1. L'applicazione presenta una prova di autenticazione, un token web JSON o un'asserzione SAML, da un pool di utenti Amazon Cognito autorizzato o da un provider di identità di terze parti in una richiesta GetID.
2. Il tuo pool di identità restituisce un ID di identità.
3. L'applicazione combina l'ID di identità con la stessa prova di autenticazione in una GetCredentialsForIdentityrichiesta.
4. Il pool di identità restituisce AWS le credenziali.
5. La tua applicazione firma le richieste AWS API con le credenziali temporanee.

L'autenticazione avanzata gestisce la logica della selezione dei ruoli IAM e del recupero delle credenziali nella configurazione del pool di identità. Puoi configurare il tuo pool di identità per selezionare un ruolo predefinito, per applicare i principi del controllo degli accessi basato sugli attributi (ABAC) o del controllo degli accessi basato sui ruoli (RBAC) alla selezione dei ruoli. Le AWS credenziali dell'autenticazione avanzata sono valide per un'ora.

Ordine delle operazioni nell'autenticazione avanzata

1. `GetId`
2. `GetCredentialsForIdentity`



Il flusso di autenticazione di base (classico)

Quando implementi il flusso di autenticazione di base, l'applicazione seleziona il ruolo IAM che desideri che gli utenti assumano.

1. L'applicazione presenta una prova di autenticazione, un token web JSON o un'asserzione SAML, da un pool di utenti Amazon Cognito autorizzato o da un provider di identità di terze parti in una richiesta GetID.
2. Il tuo pool di identità restituisce un ID di identità.
3. L'applicazione combina l'ID di identità con la stessa prova di autenticazione in una GetOpenIdTokenrichiesta.
4. `GetOpenIdToken`restituisce un nuovo token OAuth 2.0 emesso dal tuo pool di identità.
5. L'applicazione presenta il nuovo token in una AssumeRoleWithWebIdentityrichiesta.
6. Servizio di token di sicurezza AWS (AWS STS) restituisce AWS le credenziali.
7. L'applicazione firma le richieste AWS API con le credenziali temporanee.

Il flusso di lavoro di base offre un controllo più granulare sulle credenziali distribuite agli utenti. La richiesta `GetCredentialsForIdentity` del flusso di autenticazione avanzato richiede un ruolo

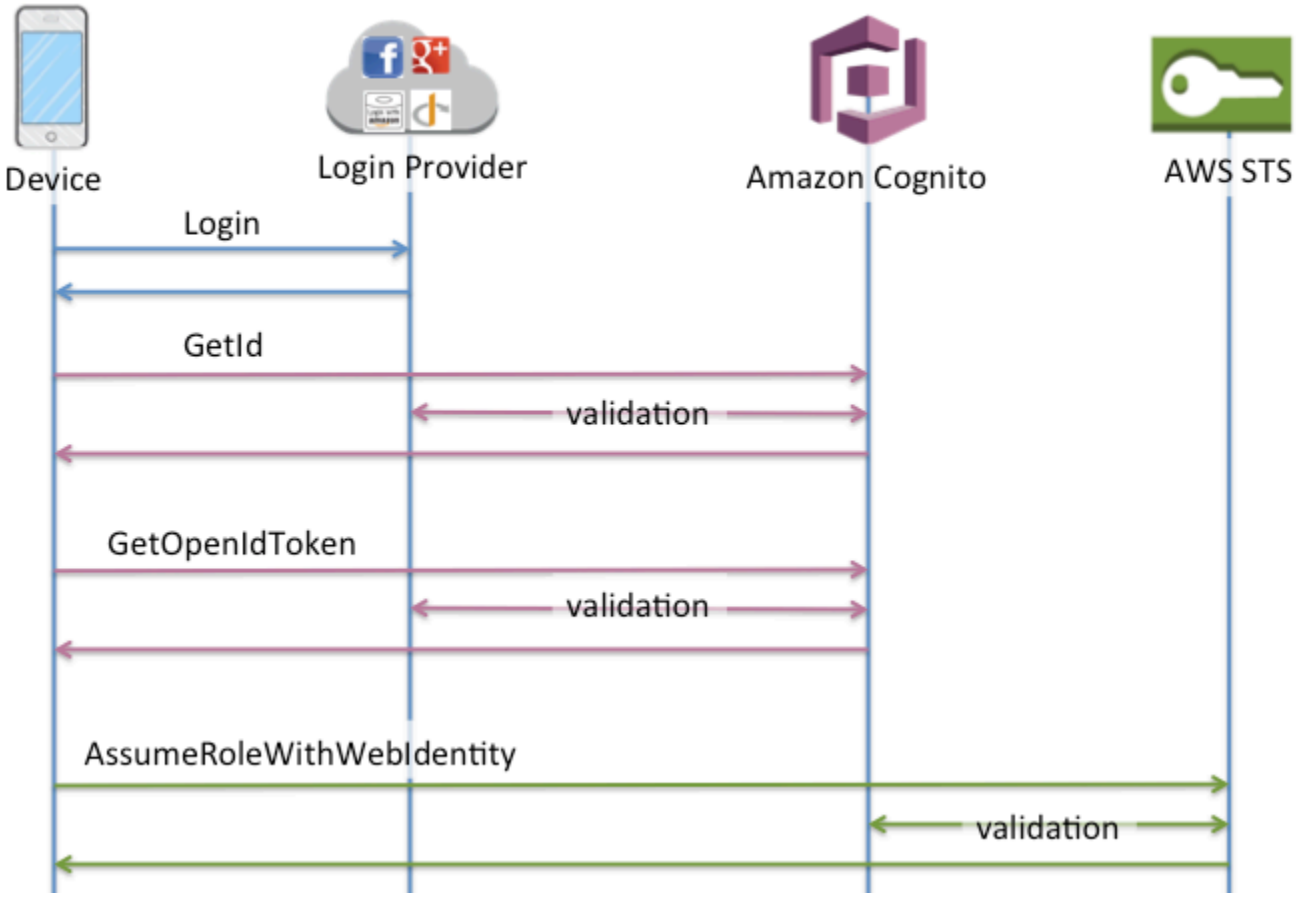
basato sul contenuto di un token di accesso. La `AssumeRoleWithWebIdentity` richiesta nel flusso di lavoro classico garantisce all'app una maggiore capacità di richiedere le credenziali per qualsiasi AWS Identity and Access Management ruolo configurato con una politica di attendibilità sufficiente. Puoi anche richiedere una durata della sessione del ruolo personalizzata.

Puoi accedere con il flusso di autenticazione di base nei pool di utenti che non dispongono di mappature dei ruoli. Questo tipo di pool di identità non ha un ruolo predefinito autenticato o non autenticato e non è configurato il controllo degli accessi basato su ruoli o attributi. Quando si tenta di accedere `GetOpenIdToken` a un pool di identità con mappature dei ruoli, viene visualizzato il seguente errore.

```
Basic (classic) flow is not supported with RoleMappings, please use enhanced flow.
```

Ordine delle operazioni nell'autenticazione di base

1. `GetId`
2. `GetOpenIdToken`
3. `AssumeRoleWithWebIdentity`

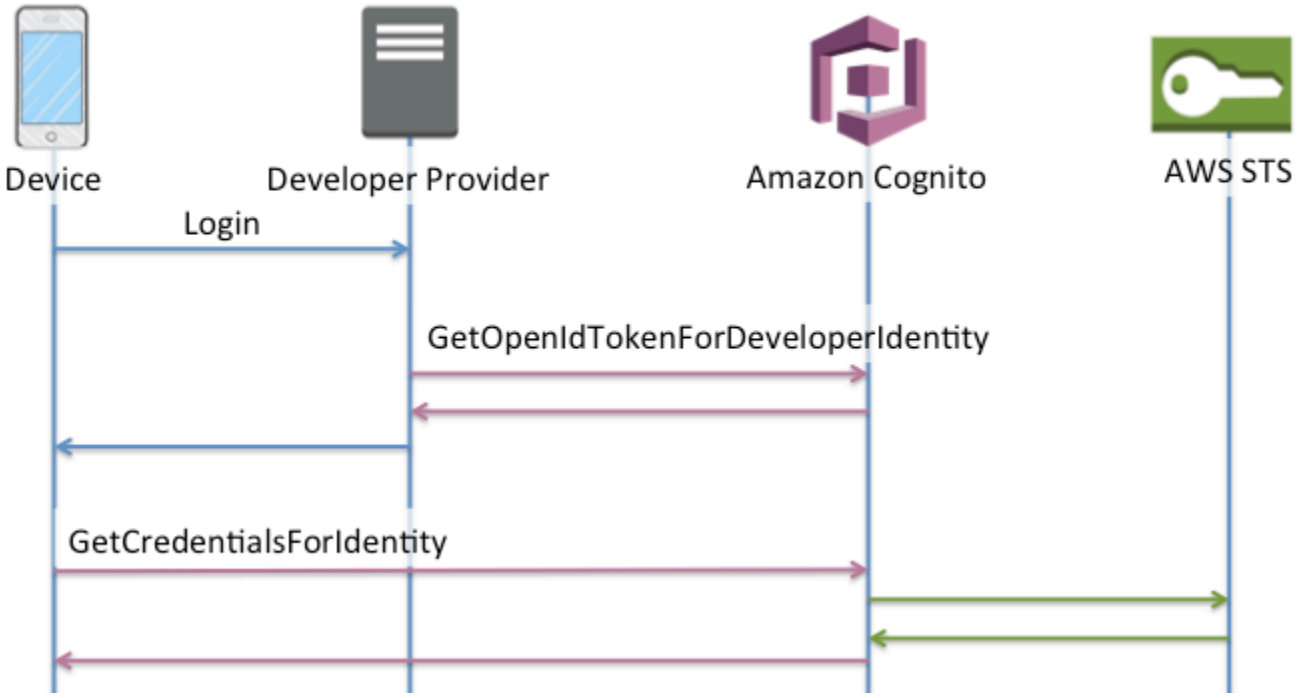


Il flusso di autenticazione autenticato dallo sviluppatore

Quando lo utilizzaDeveloper-authenticated identità, il tuo client utilizza un flusso di autenticazione diverso che include codice esterno ad Amazon Cognito per convalidare l'utente nel tuo sistema di autenticazione. Dal punto di vista del tuo pool di identità, le affermazioni che presenti nella tua richiesta di identità sono identificatori arbitrari e l'autenticazione è autorizzata dalle credenziali IAM che codifichi nella tua applicazione.

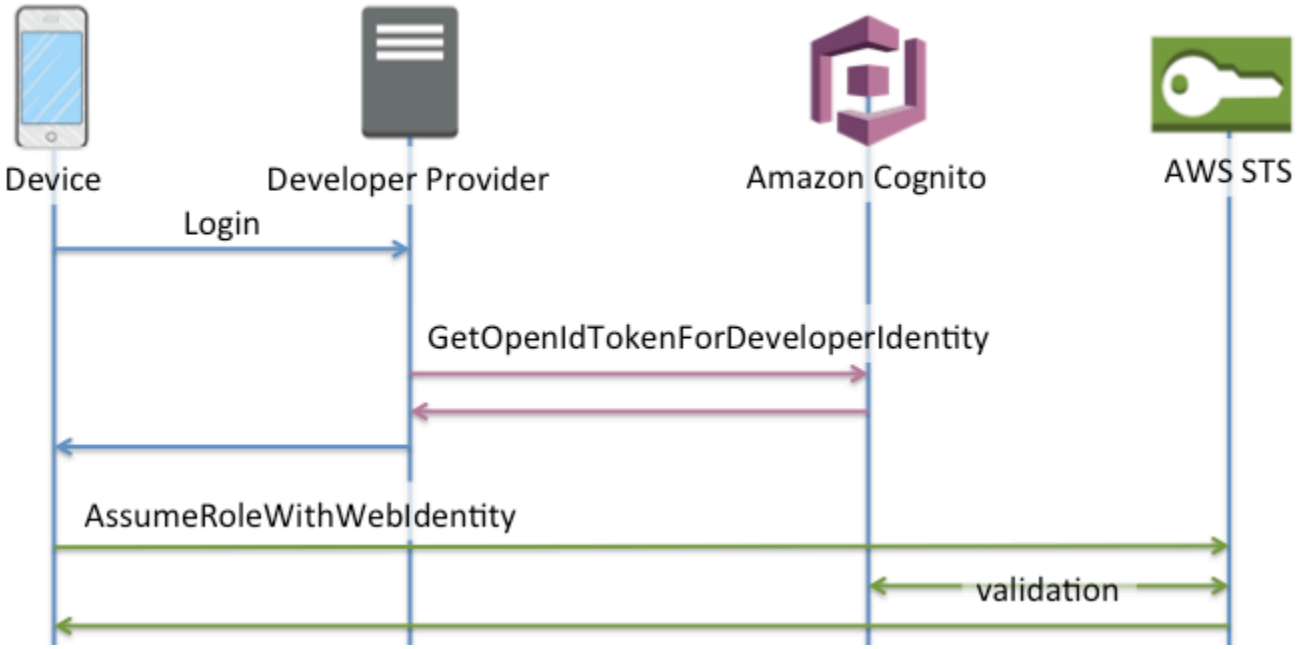
Ordine delle operazioni nell'autenticazione avanzata con un provider di sviluppo

1. Accesso tramite provider per gli sviluppatori (codice esterno Amazon Cognito)
2. Convalida dell'accesso dell'utente (codice esterno ad Amazon Cognito)
3. GetOpenIdTokenForDeveloperIdentity
4. GetCredentialsForIdentity



Ordine delle operazioni nell'autenticazione di base con un provider di sviluppo

1. Implementa la logica all'esterno del pool di identità per accedere e generare un identificatore sviluppatore-provider.
2. Recupera le credenziali memorizzate sul lato server. AWS
3. Invia l'identificatore del provider di sviluppo in una richiesta GetOpenIdTokenForDeveloperIdentityAPI firmata con credenziali autorizzate. AWS
4. Richiedi le credenziali dell'applicazione con. AssumeRoleWithWebIdentity



Quale flusso di autenticazione devo implementare?

Il flusso avanzato è la scelta più sicura con il livello più basso di impegno da parte degli sviluppatori:

- Il flusso migliorato riduce la complessità, le dimensioni e la frequenza delle richieste API.
- Non è necessario che l'applicazione effettui richieste API aggiuntive a AWS STS.
- Il tuo pool di identità valuta gli utenti in base alle credenziali del ruolo IAM che dovrebbero ricevere. Non è necessario incorporare la logica per la selezione dei ruoli nel cliente.

> **Important**
>
> Quando crei un nuovo pool di identità, non attivare l'autenticazione di base (classica) per impostazione predefinita, come best practice. Per implementare l'autenticazione di base, valuta innanzitutto le relazioni di fiducia dei tuoi ruoli IAM per le identità web. Quindi incorpora la logica per la selezione dei ruoli nel tuo client e proteggi il client dalle modifiche da parte degli utenti.

Il flusso di autenticazione di base delega la logica della selezione dei ruoli IAM alla tua applicazione. In questo flusso, Amazon Cognito convalida la sessione autenticata o non autenticata dell'utente ed emette un token con cui puoi scambiare credenziali. AWS STS Gli utenti possono scambiare i token

dell'autenticazione di base con qualsiasi ruolo IAM che si fida del tuo pool di identità e/o del tuo stato. `amr` authenticated/unauthenticated

Allo stesso modo, tenete presente che l'autenticazione degli sviluppatori è una scorciatoia per la convalida dell'autenticazione del provider di identità. Amazon Cognito considera attendibili le AWS credenziali che autorizzano una GetOpenIdTokenForDeveloperIdentityrichiesta senza un'ulteriore convalida del contenuto della richiesta. Proteggi i segreti che autorizzano l'autenticazione degli sviluppatori dall'accesso da parte degli utenti.

Panoramica delle operazioni dell'API del flusso di autenticazione

GetId

La chiamata dell'API GetId è la prima chiamata necessaria per stabilire una nuova identità in Amazon Cognito.

Accesso non autenticato

Amazon Cognito può concedere le autorizzazione di accesso alle tue applicazioni agli utenti guest non autenticati. Se questa caratteristica è abilitata nel pool di identità, gli utenti possono richiedere un nuovo ID identità in qualsiasi momento tramite l'API `GetId`. È previsto che l'applicazione memorizzi nella cache questo ID identità per effettuare chiamate successive a Amazon Cognito. Gli SDK per AWS dispositivi mobili e l' AWS SDK for JavaScript in the Browser dispongono di provider di credenziali che gestiscono questa memorizzazione nella cache per te.

Accesso autenticato

Una volta che hai configurato la tua applicazione con il supporto per un provider di accesso pubblico (Facebook, Google +, Login with Amazon o Accedi con Apple), gli utenti sono anche in grado di fornire i token (OAuth o OpenID Connect) che li identificano in quei provider. Se utilizzato in una chiamata a `GetId`, Amazon Cognito crea una nuova identità autenticata oppure restituisce l'identità già associata con quel particolare accesso. Amazon Cognito fa ciò convalidando il token con il provider e assicurando che:

- Il token sia valido e che provenga dal provider configurato.
- Il token non sia scaduto.
- Il token corrisponda con l'identificatore dell'applicazione creato con quel provider (ad esempio,l'ID dell'app Facebook).
- Il token corrisponda con l'identificatore dell'utente.

GetCredentialsForIdentity

L'API GetCredentialsForIdentity può essere richiamata dopo che hai stabilito un ID identità. Questa operazione è quindi funzionalmente equivalente alla chiamata. GetOpenIdTokenAssumeRoleWithWebIdentity

Affinché Amazon Cognito possa chiamare `AssumeRoleWithWebIdentity` per tuo conto, il tuo bacino d'utenza deve avere i ruoli IAM ad esso associati. Puoi fare ciò tramite la console Amazon Cognito o manualmente tramite l'operazione SetIdentityPoolRoles.

GetOpenIdToken

Effettua una richiesta API GetOpenIdToken dopo che hai stabilito un ID di identità. Memorizza l'ID di identità nella cache dopo la prima richiesta e avvia le successive sessioni di base (classiche) per l'identità con `GetOpenIdToken`.

La risposta a una richiesta API `GetOpenIdToken` è un token generato da Amazon Cognito. Puoi inviare questo token come parametro `WebIdentityToken` in una richiesta AssumeRoleWithWebIdentity.

Prima di inviare il token OpenID, è necessario verificarlo nell'app. Puoi utilizzare le librerie OIDC dell'SDK o una libreria come aws-jwt-verify per verificare che Amazon Cognito abbia emesso il token. L'ID della chiave di firma, o `kid`, del token OpenID è uno di quelli elencati nel documento jwks_uri† dell'identità di Amazon Cognito. Queste chiavi sono soggette a modifiche. La funzione che verifica i token di identità di Amazon Cognito deve aggiornare periodicamente l'elenco delle chiavi dal documento jwks_uri. Amazon Cognito imposta la durata dell'aggiornamento nell'intestazione della risposta cache-control di jwks_uri, attualmente con `max-age` impostato su 30 giorni.

Accesso non autenticato

Per ottenere un token per un'identità non autenticata, necessiti soltanto dell'ID identità. Non è possibile ottenere un token non autenticato per identità autenticate o disattivate.

Accesso autenticato

Se disponi di un'identità autenticata, è necessario trasmettere almeno un token valido per un accesso già associato con quella identità. Tutti i token trasmessi durante la chiamata `GetOpenIdToken` dovranno trasmettere la stessa convalida menzionata in precedenza; se uno dei token fallisce, tutta la chiamata ha esito negativo. La risposta dalla chiamata `GetOpenIdToken` include anche l'ID identità. Questo avviene perché l'ID identità che trasmetti potrebbe non essere quello che viene restituito.

Collegamento degli accessi

Se invii un token per un accesso che non è già associato a un'identità, l'accesso è considerato "collegato" all'identità associata. Puoi collegare un solo accesso per provider pubblico. I tentavi di collegare più di un accesso con un provider pubblico determinano una risposta di errore `ResourceConflictException`. Se un accesso è semplicemente collegato a un'identità esistente, l'ID identità restituito dal `GetOpenIdToken` è uguale a quello che hai trasmesso.

Unione delle identità

Se trasmetti un token per un accesso che non è attualmente collegato a una determinata identità, bensì a un'altra, le due identità vengono unite. Una volta unita, un'identità diventa quella parent/owner di tutti gli accessi associati e l'altra viene disabilitata. In questo caso, viene restituito l'ID identificativo di parent/owner . È necessario aggiornare la cache locale se questo valore è diverso. I provider degli SDK AWS mobili o dell' AWS SDK per JavaScript il browser eseguono questa operazione per conto dell'utente.

GetOpenIdTokenForDeveloperIdentity

L'GetOpenIdTokenForDeveloperIdentityoperazione sostituisce l'uso di GetIde GetOpenIdTokendal dispositivo quando si utilizzano identità autenticate dagli sviluppatori. Poiché l'applicazione firma le richieste a questa operazione API con AWS credenziali, Amazon Cognito ritiene che l'identificatore utente fornito nella richiesta sia valido. L'autenticazione degli sviluppatori sostituisce la convalida dei token eseguita da Amazon Cognito con provider esterni.

Il payload per questa API include una mappa. `logins` Questa mappa deve contenere la chiave del provider di sviluppo e un valore come identificatore per l'utente nel sistema. Se l'identificatore dell'utente non è già collegato a un'identità esistente, Amazon Cognito crea una nuova identità e restituisce il nuovo ID e il token OpenID Connect per l'identità. Se l'identificatore dell'utente è già collegato, Amazon Cognito restituisce l'ID di identità preesistente e il token OpenID Connect. Memorizza gli ID di identità autenticate da sviluppatori nella cache dopo la prima richiesta e avvia le successive sessioni di base (classiche) per l'identità con `GetOpenIdTokenForDeveloperIdentity`.

La risposta a una richiesta API `GetOpenIdTokenForDeveloperIdentity` è un token generato da Amazon Cognito. Puoi inviare questo token come parametro `WebIdentityToken` in una richiesta `AssumeRoleWithWebIdentity`.

Prima di inviare il token OpenID Connect, è necessario verificarlo nell'app. Puoi utilizzare le librerie OIDC dell'SDK o una libreria come aws-jwt-verify per verificare che Amazon Cognito abbia emesso il token. L'ID della chiave di firma, o `kid`, del token OpenID Connect è uno di quelli

elencati nel documento jwks_uri† dell'identità di Amazon Cognito. Queste chiavi sono soggette a modifiche. La funzione che verifica i token di identità di Amazon Cognito deve aggiornare periodicamente l'elenco delle chiavi dal documento jwks_uri. Amazon Cognito imposta la durata dell'aggiornamento nell'intestazione della risposta `cache-control` di jwks_uri, attualmente con `max-age` impostato su 30 giorni.

Collegamento degli accessi

Come per i provider esterni, la fornitura degli accessi aggiuntivi, che non sono già associati a un'identità, implicitamente collega quegli accessi a quella identità. Se colleghi l'accesso di un provider esterno a un'identità, l'utente può utilizzare il flusso di autenticazione del provider esterno con tale provider. Tuttavia, non può usare il nome del provider degli sviluppatori nella mappa degli accessi quando chiama `GetId` o `GetOpenIdToken`.

Unione delle identità

Amazon Cognito utilizza le identità autenticate per gli sviluppatori per supportare sia l'unione implicita che l'unione esplicita tramite l'API MergeDeveloperIdentities. Tramite l'unione esplicita puoi contrassegnare due identità con gli identificatori utente nel tuo sistema come una singola identità. È sufficiente che tu fornisca gli identificatori dell'utente di origine e di destinazione e Amazon Cognito li unisce. La prossima volta che richiedi un token OpenID Connect per ciascuno degli identificatori dell'utente, viene restituito lo stesso ID identità.

AssumeRoleWithWebIdentity

Dopo aver ottenuto un token OpenID Connect, puoi scambiarlo con AWS credenziali temporanee tramite la richiesta AssumeRoleWithWebIdentityAPI a Servizio di token di sicurezza AWS ().AWS STS

Poiché non esiste alcuna restrizione sul numero di identità che puoi creare, è importante comprendere le autorizzazioni che concedi agli utenti. Configura diversi ruoli IAM per la tua applicazione: uno per gli utenti non autenticati e uno per gli utenti autenticati. La console Amazon Cognito può creare ruoli predefiniti quando configuri per la prima volta il tuo pool di identità. A questi ruoli non è effettivamente concessa alcuna autorizzazione. Modificali per soddisfare le tue esigenze.

Ulteriori informazioni su Attendibilità del ruolo e autorizzazioni.

† Il documento jwks_uri predefinito dell'identità di Amazon Cognito contiene le informazioni sulle chiavi che firmano i token per i pool di identità nella maggior parte delle Regioni AWS. Le seguenti regioni hanno documenti jwks_uri diversi.

Amazon Cognito Identity JSON web key URIs in other Regioni AWS

Regione AWS	Percorso del documento jwks_uri
AWS GovCloud (US-West)	`https://cognito-identity.us-gov-west-1.amazonaws.com/.well-known/jwks_uri`
Cina (Pechino)	`https://cognito-identity.cn-north-1.amazonaws.com.cn/.well-known/jwks_uri`
Opt-in Regioni come l'Europa (Milano) e l'Africa (Città del Capo)	`https://cognito-identity.` *`Region`*`.amazonaws.com/.well-known/jwks_uri`

Puoi anche estrapolare il documento jwks_uri dall'emittente, o `iss`, che ricevi nel token OpenID da Amazon Cognito. L'endpoint di OIDC-standard scoperta `<issuer>/.well-known/openid-configuration` elenca un percorso verso jwks_uri per il tuo token.

Ruoli IAM

Durante la creazione di un pool di identità, ti viene chiesto di aggiornare i ruoli IAM che gli utenti assumono. I ruoli IAM funzionino in questo modo: quando un utente effettua l'accesso nella tua applicazione, Amazon Cognito genera credenziali AWS temporanee per l'utente. Queste credenziali temporanee sono associate a un determinato ruolo IAM. Il ruolo IAM ti consente di definire un set di autorizzazioni per accedere alle tue risorse AWS .

Puoi specificare ruoli IAM predefiniti per utenti autenticati e non autenticati. Inoltre, puoi definire le regole per scegliere il ruolo di ogni utente in base alle richieste nel token ID dell'utente. Per ulteriori informazioni, consulta Utilizzo del controllo degli accessi basato su ruoli.

Di default, la console Amazon Cognito crea ruoli IAM con accesso ad Amazon Cognito Sync e ad Amazon Mobile Analytics. In alternativa puoi scegliere di utilizzare ruoli IAM; esistenti.

Modificare i ruoli IAM per consentire o limitare l'accesso ad altri servizi. Per farlo, accedi alla console IAM. Quindi seleziona Roles (Ruoli) e seleziona un ruolo. Le policy associate al ruolo selezionato sono elencate nella scheda Permissions (Autorizzazioni). Puoi personalizzare una policy di accesso

predefinita selezionando il collegamento Manage Policy (Gestisci Policy) corrispondente. Per ulteriori informazioni sull'utilizzo e per definire le policy, consulta Panoramica delle policy IAM.

> **Note**
>
> È consigliabile definire policy in grado di seguire il principio di concessione di privilegi minimi. In altre parole, le policy includono solo le autorizzazioni che gli utenti richiedono per eseguire le attività. Per ulteriori informazioni, consulta Assegnare il privilegio minimo nella Guida per l'utente di IAM.
>
> Ricorda che le identità non autenticate vengono assunte dagli utenti che non accedono alla tua app. Di solito, le autorizzazioni che assegni per le identità non autenticate devono essere più restrittive rispetto a quelle per le identità autenticate.

Argomenti

- Configurazione di una policy di attendibilità
- Policy di accesso
- Attendibilità del ruolo e autorizzazioni

Configurazione di una policy di attendibilità

Amazon Cognito usa i ruoli IAM per generare credenziali temporanee per gli utenti della tua applicazione. L'accesso alle autorizzazioni è controllato da una relazione di attendibilità del ruolo. Ulteriori informazioni su Attendibilità del ruolo e autorizzazioni. Amazon Cognito gestisce le connessioni tra un pool di identità AWS STS e un pool di identità. IdPs

Il token presentato AWS STS viene generato da un pool di identità, che traduce un pool di utenti, un token social o di un provider OIDC o un'asserzione SAML nel proprio token. Il token del pool di identità contiene un'attestazione `aud` che è l'ID del pool di identità.

Laddove la policy `Principal` di trust dei ruoli IAM sia il responsabile di un servizio di pool di identità`cognito-identity.amazonaws.com`, non è possibile creare o modificare policy di trust dei ruoli in modo tale da consentire a qualsiasi pool di identità di assumere il ruolo. Per quanto riguarda il principale del pool di identità, l'`Action`elemento deve avere un valore `Condition` che richieda che `AssumeRoleWithWebIndentity` venga eseguito solo dai pool di identità, come specificato da una chiave di condizione come`cognito-identity.amazonaws.com:aud`. Sono disponibili altre

chiavi di condizione, ma `aud` sono obbligatorie. Se si tenta di salvare una policy di attendibilità dei ruoli senza una condizione di questo tipo, IAM restituisce un errore.

Per ulteriori informazioni sulle chiavi di federazione OIDC (web identity), vedi Chiavi disponibili per la federazione AWS OIDC.

Di seguito sono riportate le chiavi delle condizioni di federazione OIDC disponibili per Amazon Cognito.

`cognito-identity.amazonaws.com:aud`

Limita il ruolo alle operazioni provenienti da uno o più pool di identità. Amazon Cognito indica il pool di identità di origine nel `aud` claim nel token del pool di identità.

`cognito-identity.amazonaws.com:amr`

Limita il ruolo a uno `authenticated` o agli utenti `unauthenticated` (ospiti). Amazon Cognito indica lo stato di autenticazione nel `amr` claim nel token del pool di identità.

`cognito-identity.amazonaws.com:sub`

Limita il ruolo a uno o più utenti in base all'UUID. Questo UUID è l'ID di identità dell'utente nel pool di identità. Questo valore non è il `sub` valore del provider di identità originale dell'utente. Amazon Cognito indica questo UUID nell'`sub`attestazione nel token del pool di identità.

Il seguente esempio di policy di trust dei ruoli consente al responsabile del servizio federato di `cognito-identity.amazonaws.com` chiamare l'API. AWS STS `AssumeRoleWithWebIdentity` La richiesta andrà a buon fine solo se il token del pool di identità nella richiesta API presenta le seguenti attestazioni.

1. Un'attestazione `aud` dell'ID del pool di identità `us-west-2:abcdefg-1234-5678-910a-0e8443553f95`.
2. Un'attestazione `amr` di `authenticated` viene aggiunta quando l'utente ha effettuato l'accesso e non è un utente ospite.

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-
west-2:abcdefg-1234-5678-910a-0e8443553f95"
                },
                "ForAnyValue:StringLike": {
                    "cognito-identity.amazonaws.com:amr": "authenticated"
                }
            }
        }
    ]
}
```

Politiche di fiducia per i ruoli IAM nell'autenticazione Basic (Classic)

Riepilogo

I pool di identità possono assumere ruoli per conto degli utenti nel flusso di autenticazione di base solo quando la politica di attendibilità del ruolo di destinazione contiene la `aud` condizione.

L'autenticazione di base presenta le stesse limitazioni relative alle politiche di attendibilità dei ruoli non sicure dell'autenticazione avanzata: non è possibile salvare una politica di fiducia dei ruoli che non limita i pool di identità supportati con una `aud` condizione. Questa limitazione non è stata applicata al momento dell'avvio del servizio. Prima dell'applicazione di questo requisito, era possibile creare politiche di fiducia dei ruoli che non prevedessero condizioni di sicurezza aggiuntive. Dopo l'applicazione di questo requisito, AWS STS consente alle identità web di assumere ruoli non protetti da condizioni, ma tali ruoli non possono essere modificati senza introdurre tali condizioni.

Enhanced-flow l'autenticazione richiede che il ruolo IAM appartenga allo stesso pool Account AWS di identità. Tuttavia, nell'autenticazione di base, in cui l'applicazione compone la `AssumeRoleWithWebIdentity` richiesta, l'applicazione può richiedere di assumere un ruolo in un account diverso. La tua richiesta di assunzione del ruolo tra account diversi avrà esito negativo, tuttavia, se il ruolo di destinazione ha una policy di fiducia legacy che non impone la condizione. `aud`

Il token emesso da un pool di identità per un'identità contiene informazioni sull'origine del pool Account AWS di identità. Quando presenti un token del pool di identità in una richiesta AssumeRoleWithWebIdentityAPI, AWS STS verifica se il pool di identità di origine ha lo stesso Account AWS ruolo di IAM. Se AWS STS determina che la richiesta è interaccount, verifica se la policy di fiducia dei ruoli presenta una `aud` condizione. La chiamata assume-role fallisce se tale condizione non è presente nella policy di fiducia dei ruoli. Se la richiesta non è interaccount, AWS STS non applica questa restrizione. Come procedura consigliata, applica sempre una condizione di questo tipo alle politiche di attendibilità dei ruoli del pool di identità.

Di seguito è riportato un esempio di policy di fiducia che soddisfa i requisiti minimi per un ruolo IAM per l'autenticazione di base con diversi pool di identità. Come best practice, consenti anche le identità autenticate con una `"cognito-identity.amazonaws.com:amr": "authenticated"` condizione.

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-west-2:abcdefg-1234-5678-910a-0e8443553f95",
                        "us-west-2:hijklmo-5678-9101-112b-0e4221776g96",
                        "us-west-2:pqrstuv-9101-1121-314c-0e2110887h97"
                    ]
                }
            }
        }
    ]
}
```

Condizioni aggiuntive relative alla politica di fiducia

Puoi utilizzare le seguenti condizioni di policy di fiducia per definire i pool di identità di origine, le identità e i provider che possono assumere ruoli IAM.

> **Note**
>
> Non implementate la chiave aws:SourceIpcondition nelle policy di fiducia per i ruoli IAM che i pool di identità assumono nel flusso di autenticazione avanzato. Poiché il flusso avanzato genera la AssumeRoleWithWebIdentityrichiesta per conto dell'applicazione, l'IP di origine della richiesta non sarà l'IP del client dell'applicazione e la condizione non sarà mai soddisfatta. Network-based le chiavi di condizione sono valide per i ruoli assunti dai pool di identità solo attraverso il flusso di base, che non dispone delle funzionalità lato servizio del flusso avanzato.

Riutilizzo dei ruoli tra i pool di identità

Per riutilizzare un ruolo tra più pool di identità, poiché condividono un set di autorizzazioni, puoi includere più pool di identità come segue:

```
"StringEquals": {
    "cognito-identity.amazonaws.com:aud": [
        "us-east-1:12345678-abcd-abcd-abcd-123456790ab",
        "us-east-1:98765432-dcba-dcba-dcba-123456790ab"
    ]
}
```

Limitazione dell'accesso a specifiche identità

Per creare una policy limitata a uno specifico set di app utenti, controlla il valore di `cognito-identity.amazonaws.com:sub`:

```
"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"
    ]
```

```
}
```

Limitazione dell'accesso a specifici provider

Per creare una policy limitata agli utenti che hanno effettuato l'accesso con un provider specifico (magari il tuo provider di accesso), controlla il valore di: `cognito-identity.amazonaws.com:amr`

```
"ForAnyValue:StringLike": {
    "cognito-identity.amazonaws.com:amr": "login.myprovider.myapp"
}
```

Ad esempio, un'applicazione che fa affidamento solo su Facebook avrebbe la seguente clausola amr:

```
"ForAnyValue:StringLike": {
    "cognito-identity.amazonaws.com:amr": "graph.facebook.com"
}
```

Policy di accesso

Le autorizzazioni collegate a un ruolo si applicano a tutti gli utenti che assumono tale ruolo. Per eseguire il partizionamento dell'accesso degli utenti, utilizza le condizioni e le variabili di policy. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag. Puoi utilizzare la condizione `sub` per limitare le operazioni agli ID di identità di Amazon Cognito nelle policy di accesso. Utilizza questa opzione con cautela, in particolare per identità non autenticate, prive di un ID utente coerente. Per ulteriori informazioni sulle variabili di policy IAM per la federazione web con Amazon Cognito, consulta IAM and AWS STS condition context keys nella Guida per l'AWS Identity and Access Management utente.

Per una maggiore protezione di sicurezza, Amazon Cognito applica una policy di riduzione dell'ambito alle credenziali assegnate agli utenti non autenticati nel flusso avanzato, utilizzando `GetCredentialsForIdentity`. La policy di riduzione dell'ambito aggiunge una Policy di sessione inline e una AWS politica di sessione gestita alle policy IAM applicate al ruolo non autenticato. Poiché è necessario concedere l'accesso in entrambe le policy IAM per le policy del ruolo e di sessione, la policy di riduzione dell'ambito limita l'accesso degli utenti a servizi diversi da quelli contenuti nell'elenco seguente.

> **Note**
>
> Nel flusso di base (classico), puoi creare la tua richiesta API AssumeRoleWithWebIdentity e applicare queste restrizioni alla richiesta. Come best practice di sicurezza, non assegnare autorizzazioni superiori a questa policy di riduzione dell'ambito a utenti non autenticati.

Amazon Cognito impedisce inoltre agli utenti autenticati e non autenticati di effettuare richieste API ai pool di identità di Amazon Cognito e Amazon Cognito Sync. Altri Servizi AWS potrebbero imporre restrizioni all'accesso ai servizi dalle identità web.

In caso di esito positivo della richiesta con il flusso avanzato, Amazon Cognito effettua una richiesta API `AssumeRoleWithWebIdentity` in background. Tra i parametri di questa richiesta, Amazon Cognito include quelli riportati di seguito.

1. L'ID identificativo dell'utente.
2. L'ARN del ruolo IAM che l'utente desidera assumere.
3. Un parametro `policy` che aggiunge una policy di sessione inline.
4. Un `PolicyArns.member.N` parametro il cui valore è una politica AWS gestita che concede autorizzazioni aggiuntive in Amazon. CloudWatch

Servizi cui possono accedere gli utenti non autenticati

Quando utilizzi il flusso avanzato, le policy di riduzione dell'ambito applicate da Amazon Cognito alla sessione dell'utente impediscono l'utilizzo di servizi diversi da quelli elencati nella tabella seguente. Per un sottoinsieme di servizi, sono consentite solo azioni specifiche.

Categoria	Servizio
Analisi	Amazon Data Firehose Servizio gestito da Amazon per Apache Flink
Integrazione di applicazioni	Amazon Simple Queue Service
Realtà aumentata e realtà virtuale	Amazon Sumerian
Applicazioni aziendali	Amazon Mobile Analytics

Categoria	Servizio
	Amazon Simple Email Service
Calcolo	AWS Lambda
Crittografia e PKI	AWS Key Management Service[1]
Database	Amazon DynamoDB Amazon SimpleDB
Front-end Web e dispositivi mobili	AWS AppSync Servizio di posizione Amazon Amazon Simple Notification Service Amazon Pinpoint Servizio di posizione Amazon
Sviluppo dei giochi	GameLift Server Amazon
Internet of Things (IoT)	AWS IoT

Categoria	Servizio
Machine Learning	Amazon CodeWhisperer Amazon Comprehend Amazon Lex Amazon Machine Learning Amazon Personalize Amazon Polly Amazon Rekognition Amazon SageMaker AI[1] Amazon Textract[1] Amazon Transcribe Amazon Translate
Gestione e governance	Amazon CloudWatch CloudWatch Registri Amazon
Reti e distribuzione di contenuti	Gateway Amazon API
Sicurezza, identità e conformità	Pool di utenti Amazon Cognito
Archiviazione	Amazon Simple Storage Service

[1] Per quanto riguarda la tabella seguente, la politica Servizi AWS in linea prevede un sottoinsieme di azioni. Nella tabella sono visualizzate le azioni disponibili in ciascuno di essi.

Servizio AWS	Autorizzazioni massime per gli utenti non autenticati del flusso avanzato

AWS Key Management Service	`Encrypt`
	`Decrypt`
	`ReEncryptTo`
	`ReEncryptFrom`
	`GenerateDataKey`
	`GenerateDataKeyPair`
	`GenerateDataKeyPair`
	`GenerateDataKeyPairWithoutPlaintext`
	`GenerateDataKeyWithoutPlaintext`

Amazon SageMaker AI	`InvokeEndpoint`
Amazon Textract	`DetectDocumentText` `AnalyzeDocument`
Amazon Sumerian	`View*`

Servizio di posizione Amazon	`SearchPlaceIndex*`
	`GetPlace`
	`CalculateRoute*`
	`*Geofence`
	`*Geofences`
	`*DevicePosition*`

Per concedere l'accesso a un numero Servizi AWS superiore a questo elenco, attiva il flusso di autenticazione di base (classico) nel tuo pool di identità. Se i tuoi utenti riscontrano `NotAuthorizedException` errori Servizi AWS dovuti alle policy assegnate al ruolo IAM per gli utenti non autenticati, valuta se puoi rimuovere quel servizio dal tuo caso d'uso. Se non è possibile, passa al flusso di base.

La politica di sessione in linea per gli utenti ospiti

Amazon Cognito applica innanzitutto una policy in linea nella richiesta di credenziali IAM. La policy di sessione inline limita le autorizzazioni effettive dell'utente per evitare di includere l'accesso a qualsiasi Servizi AWS diverso da quelli elencati di seguito. Inoltre, devi concedere le autorizzazioni a queste ultime Servizi AWS nelle politiche che applichi al ruolo IAM dell'utente. Le autorizzazioni effettive di un utente per una sessione del ruolo assunto sono l'intersezione delle policy assegnate al relativo ruolo e policy di sessione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di AWS Identity and Access Management .

Amazon Cognito aggiunge la seguente policy inline alle sessioni per gli utenti nelle Regioni AWS abilitate per impostazione predefinita. Per una panoramica dell'effetto netto della politica in linea e di altre politiche di sessione, consulta. Servizi cui possono accedere gli utenti non autenticati

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:*",
                "logs:*",
                "dynamodb:*",
                "kinesis:*",
                "mobileanalytics:*",
                "s3:*",
                "ses:*",
                "sns:*",
                "sqs:*",
                "lambda:*",
                "machinelearning:*",
                "execute-api:*",
                "iot:*",
```

```
                "gamelift:*",
                "cognito-identity:*",
                "cognito-idp:*",
                "lex:*",
                "polly:*",
                "comprehend:*",
                "translate:*",
                "transcribe:*",
                "rekognition:*",
                "mobiletargeting:*",
                "firehose:*",
                "appsync:*",
                "personalize:*",
                "sagemaker:InvokeEndpoint",
                "cognito-sync:*",
                "codewhisperer:*",
                "textract:DetectDocumentText",
                "textract:AnalyzeDocument",
                "sdb:*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Per tutte le altre regioni, la policy di riduzione dell'ambito inline include tutto ciò che è elencato nelle regioni predefinite ad eccezione delle seguenti istruzioni `Action`.

```
            "cognito-sync:*",
            "sumerian:View*",
            "codewhisperer:*",
            "textract:DetectDocumentText",
            "textract:AnalyzeDocument",
            "sdb:*"
```

Il AWS politica di sessione gestita per gli ospiti

Amazon Cognito applica anche una policy AWS gestita come policy di sessione alle sessioni a flusso avanzato di ospiti non autenticati. Questa policy limita l'ambito delle autorizzazioni degli utenti non autenticati alla policy. `AmazonCognitoUnAuthedIdentitiesSessionPolicy`

Questa autorizzazione deve essere specificata anche nelle policy che colleghi al ruolo IAM non autenticato. Le autorizzazioni effettive di un utente per una sessione con un ruolo presunto sono l'intersezione tra le politiche IAM assegnate al suo ruolo e le relative politiche di sessione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di AWS Identity and Access Management .

Per una panoramica dell'effetto netto di questa policy AWS gestita e di altre policy di sessione, consulta. Servizi cui possono accedere gli utenti non autenticati

La policy gestita `AmazonCognitoUnAuthedIdentitiesSessionPolicy` contiene le seguenti autorizzazioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "rum:PutRumEvents",
            "polly:*",
            "comprehend:*",
            "translate:*",
            "transcribe:*",
            "rekognition:*",
            "mobiletargeting:*",
            "firehose:*",
            "personalize:*",
            "sagemaker:InvokeEndpoint",
            "geo:GetMap*",
            "geo:SearchPlaceIndex*",
            "geo:GetPlace",
            "geo:CalculateRoute*",
            "geo:*Geofence",
            "geo:*Geofences",
            "geo:*DevicePosition*",
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncryptTo",
            "kms:ReEncryptFrom",
            "kms:GenerateDataKey",
            "kms:GenerateDataKeyPair",
```

```
            "kms:GenerateDataKeyPairWithoutPlaintext",
            "kms:GenerateDataKeyWithoutPlaintext"
        ],
        "Resource": "*"
    }]
}
```

Esempi di policy di accesso

In questa sezione, sono disponibili policy di accesso di Amazon Cognito di esempio che concedono agli utenti le autorizzazioni minime necessarie per eseguire un'operazione specifica. Puoi limitare ulteriormente le autorizzazioni per un determinato ID identità utilizzando variabili di policy, laddove possibile. Ad esempio, utilizzando ${cognito-identity.amazonaws.com:sub}. Per ulteriori informazioni, consulta Informazioni sull'autenticazione di Amazon Cognito parte 3: Ruoli e policy nel blog di AWS Mobile.

> **Note**
>
> Come best practice di sicurezza, le policy devono includere solo le autorizzazioni necessarie agli utenti per eseguire le loro attività. Ciò significa che dovresti cercare sempre di definire, laddove possibile, l'ambito dell'accesso a una singola identità per gli oggetti.

Autorizzazione per l'accesso in lettura di un'identità a un singolo oggetto in Amazon S3

La seguente policy di accesso concede autorizzazioni di lettura a un'identità per recuperare un singolo oggetto da un determinato bucket S3.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/assets/my_picture.jpg"]
```

```
    }
  ]
}
```

Autorizzazione per l'accesso in lettura e scrittura di un'identità a percorsi specifici in Amazon S3

La seguente policy di accesso concede le autorizzazioni di lettura e scrittura per accedere a una specifica "cartella" prefisso in un bucket S3 mappando il prefisso alla variabile `${cognito-identity.amazonaws.com:sub}`.

Con questa policy, un'identità come `us-east-1:12345678-1234-1234-1234-123456790ab` inserita tramite `${cognito-identity.amazonaws.com:sub}` sarà in grado di ottenere, inserire ed elencare oggetti in `arn:aws:s3:::amzn-s3-demo-bucket/us-east-1:12345678-1234-1234-1234-123456790ab`. Tuttavia, all'identità non verrà concesso l'accesso ad altri oggetti in `arn:aws:s3:::amzn-s3-demo-bucket`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket"],
      "Condition": {"StringLike": {"s3:prefix": ["${cognito-
identity.amazonaws.com:sub}/*"]}}
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/${cognito-
identity.amazonaws.com:sub}/*"]
    }
  ]
}
```

Un modello di accesso simile si ottiene con Amazon S3 Access Grants.

Assegnazione dell'accesso granulare delle identità ad Amazon DynamoDB

La seguente policy di accesso fornisce un controllo granulare dell'accesso alle risorse DynamoDB utilizzando variabili d'ambiente Amazon Cognito. Tali variabili concedono l'accesso agli elementi in DynamoDB attraverso l'ID identità. Per ulteriori informazioni, consulta Using IAM Policy Conditions for Fine-Grained Access Control nella Amazon DynamoDB Developer Guide.

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:us-west-2:123456789012:table/MyTable"
      ],
      "Condition": {
        "ForAllValues:StringEquals": {
          "dynamodb:LeadingKeys":  ["${cognito-identity.amazonaws.com:sub}"]
        }
      }
    }
  ]
}
```

Autorizzazione di un'identità a richiamare una funzione Lambda

La seguente policy di accesso concede a un'identità l'autorizzazione per richiamare una funzione Lambda.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
             "Effect": "Allow",
             "Action": "lambda:InvokeFunction",
             "Resource": [
                 "arn:aws:lambda:us-west-2:123456789012:function:MyFunction"
             ]
         }
     ]
 }
```

Autorizzazione di un'identità a pubblicare record in flussi di dati Kinesis

La policy di accesso seguente consente a un'identità di utilizzare l'operazione `PutRecord` con qualsiasi Kinesis Data Streams. Può essere applicata agli utenti che devono aggiungere record di dati a tutti i flussi in un account. Per ulteriori informazioni, consulta Controllo dell'accesso alle risorse Amazon Kinesis Data Streams che utilizzano IAM nella Guida per gli sviluppatori di Amazon Kinesis Data Streams.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kinesis:PutRecord",
            "Resource": [
                "arn:aws:kinesis:us-east-1:111122223333:stream/stream1"
            ]
        }
    ]
}
```

Autorizzazione di un'identità ad accedere ai propri dati nell'archivio di Amazon Cognito Sync

La seguente policy di accesso concede a un'identità le autorizzazioni solo per i dati nell'archivio di Amazon Cognito Sync.

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[{
      "Effect":"Allow",
      "Action":"cognito-sync:*",
      "Resource":["arn:aws:cognito-sync:us-east-1:123456789012:identitypool/
${cognito-identity.amazonaws.com:aud}/identity/${cognito-
identity.amazonaws.com:sub}/*"]
      }]
  }
```

Attendibilità del ruolo e autorizzazioni

Questi ruoli differiscono nelle loro relazioni di attendibilità. Di seguito viene riportato un esempio di policy di attendibilità per i ruoli non autenticati:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-corner-
cafe-123456790ab"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "unauthenticated"
```

```
        }
      }
    }
  ]
}
```

Questa policy concede l'autorizzazione agli utenti federati di `cognito-identity.amazonaws.com` (approvatore del token OpenID Connect) di assumere questo ruolo. Inoltre la policy limita l'`aud` del token, in questo caso l'ID dei pool di identità, in base al pool di identità. Infine, la policy specifica che uno dei membri dell'array dell'attestazione `amr` con più valori del token emesso dall'operazione API `GetOpenIdToken` Amazon Cognito ha il valore `unauthenticated`.

Quando Amazon Cognito crea un token, imposta l'`amr` del token o come `unauthenticated` o `authenticated`. Se `amr` è `authenticated`, il token include tutti i provider utilizzati durante l'autenticazione. Ciò significa che puoi creare un ruolo che si fidi solo degli utenti che hanno effettuato l'accesso tramite Facebook modificando la `amr` condizione come mostrato:

```
"ForAnyValue:StringLike": {
  "cognito-identity.amazonaws.com:amr": "graph.facebook.com"
}
```

Presta attenzione quando vengono modificate le relazioni di affidabilità sui tuoi ruoli, oppure quando si tenta di utilizzare ruoli tra pool di identità. Se il ruolo non è configurato per fare affidamento in maniera corretta sul tuo pool di identità, STS genera un'eccezione come la seguente:

```
AccessDenied -- Not authorized to perform sts:AssumeRoleWithWebIdentity
```

Se vedi questo mesaggio, verifica che il tuo pool di identità e il tipo di autenticazione siano appropriati per il ruolo.

Best practice di sicurezza per i pool di identità di Amazon Cognito

I pool di identità di Amazon Cognito forniscono AWS credenziali temporanee per la tua applicazione. Account AWS spesso contengono sia le risorse di cui gli utenti dell'applicazione hanno bisogno, sia risorse private di back-end. I ruoli e le policy IAM che costituiscono le AWS credenziali possono concedere l'accesso a qualsiasi di queste risorse.

La best practice principale per la configurazione del pool di identità consiste nel garantire che l'applicazione possa svolgere il lavoro senza privilegi eccessivi o involontari. Per evitare errori di

configurazione della sicurezza, consulta questi consigli prima del lancio di ogni applicazione che desideri rilasciare in produzione.

Argomenti

- Le migliori pratiche di configurazione IAM
- Best practice per la configurazione del pool di identità

Le migliori pratiche di configurazione IAM

Quando un ospite o un utente autenticato avvia una sessione nell'applicazione che richiede le credenziali del pool di identità, l'applicazione recupera le AWS credenziali temporanee per un ruolo IAM. Le credenziali potrebbero riguardare un ruolo predefinito, un ruolo scelto dalle regole nella configurazione del pool di identità o un ruolo personalizzato scelto dall'app. Con le autorizzazioni assegnate a ciascun ruolo, l'utente ottiene l'accesso alle risorse. AWS

Per ulteriori informazioni sulle best practice generali di IAM, consulta le best practice di IAM nella Guida per l' AWS Identity and Access Management utente.

Utilizza le condizioni delle policy di fiducia nei ruoli IAM

IAM richiede che i ruoli per i pool di identità abbiano almeno una condizione di policy di fiducia. Questa condizione può, ad esempio, impostare l'ambito del ruolo solo per gli utenti autenticati. AWS STS richiede inoltre che le richieste di autenticazione di base tra account abbiano due condizioni specifiche: `cognito-identity.amazonaws.com:aud` e. `cognito-identity.amazonaws.com:amr` Come best practice, applica entrambe queste condizioni a tutti i ruoli IAM che si affidano al principale `cognito-identity.amazonaws.com` del servizio di pool di identità.

- `cognito-identity.amazonaws.com:aud`: L'affermazione aud nel token del pool di identità deve corrispondere a un ID affidabile del pool di identità.
- `cognito-identity.amazonaws.com:amr`: L'attestazione amr nel token del pool di identità deve essere autenticata o non autenticata. Con questa condizione, puoi riservare l'accesso a un ruolo solo agli ospiti non autenticati o solo agli utenti autenticati. È possibile rifinire ulteriormente il valore di questa condizione per limitare il ruolo agli utenti di un provider specifico, ad esempio. `graph.facebook.com`

Il seguente esempio di politica di fiducia dei ruoli concede l'accesso a un ruolo nelle seguenti condizioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}
```

Elementi che si riferiscono ai pool di identità

- `"Federated": "cognito-identity.amazonaws.com"`: gli utenti devono provenire da un pool di identità.
- `"cognito-identity.amazonaws.com:aud": "us-east-1:a1b2c3d4-5678-90ab-cdef-example11111"`: Gli utenti devono provenire dal pool di identità specificous-east-1:a1b2c3d4-5678-90ab-cdef-example11111.
- `"cognito-identity.amazonaws.com:amr": "authenticated"`: Gli utenti devono essere autenticati. Gli utenti ospiti non possono assumere il ruolo.

Applica le autorizzazioni con privilegi minimi

Quando imposti le autorizzazioni con le policy IAM per l'accesso autenticato o l'accesso ospite, concedi solo le autorizzazioni specifiche necessarie per eseguire attività specifiche o le autorizzazioni

con privilegi minimi. La seguente policy IAM di esempio, se applicata a un ruolo, concede l'accesso in sola lettura a un singolo file di immagine in un bucket Amazon S3.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket/assets/my_picture.jpg"]
    }
  ]
}
```

Best practice per la configurazione del pool di identità

I pool di identità offrono opzioni flessibili per la generazione di AWS credenziali. Non utilizzare scorciatoie di progettazione quando l'applicazione può funzionare con i metodi più sicuri.

Comprendi gli effetti dell'accesso degli ospiti

L'accesso non autenticato come ospite consente agli utenti di recuperare i dati da te Account AWS prima di effettuare l'accesso. Chiunque conosca l'ID del tuo pool di identità può richiedere credenziali non autenticate. L'ID del tuo pool di identità non è un'informazione riservata. Quando attivi l'accesso come ospite, le AWS autorizzazioni concesse alle sessioni non autenticate sono disponibili per tutti.

Come procedura consigliata, lascia disattivato l'accesso degli ospiti e recupera le risorse necessarie solo dopo l'autenticazione degli utenti. Se l'applicazione richiede l'accesso alle risorse prima dell'accesso, prendi le seguenti precauzioni.

- Acquisite familiarità con le limitazioni automatiche imposte ai ruoli non autenticati.
- Monitora e modifica le autorizzazioni dei ruoli IAM non autenticati per soddisfare le esigenze specifiche della tua applicazione.
- Concedi l'accesso a risorse specifiche.
- Proteggi la policy di fiducia del tuo ruolo IAM predefinito non autenticato.

- Attiva l'accesso come ospite solo quando sei sicuro di poter concedere le autorizzazioni nel tuo ruolo IAM a chiunque su Internet.

Utilizza l'autenticazione avanzata per impostazione predefinita

Con l'autenticazione di base (classica), Amazon Cognito delega la selezione del ruolo IAM alla tua app. Al contrario, il flusso avanzato utilizza la logica centralizzata nel pool di identità per determinare il ruolo IAM. Fornisce inoltre una sicurezza aggiuntiva per le identità non autenticate con una politica mirata che stabilisce un limite massimo per le autorizzazioni IAM. Il flusso avanzato è la scelta più sicura con il livello più basso di impegno da parte degli sviluppatori. Per ulteriori informazioni su queste opzioni, consultaFlusso di autenticazione dei pool di identità.

Il flusso di base può esporre la logica lato client che riguarda la selezione dei ruoli e l'assemblaggio della richiesta di credenziali dell'API AWS STS. Il flusso migliorato nasconde sia la logica che la richiesta di assunzione del ruolo alla base dell'automazione del pool di identità.

Quando configuri l'autenticazione di base, applica le migliori pratiche IAM ai tuoi ruoli IAM e alle relative autorizzazioni.

Utilizza i provider di sviluppo in modo sicuro

Le identità autenticate dagli sviluppatori sono una funzionalità dei pool di identità per le applicazioni lato server. L'unica prova di autenticazione richiesta dai pool di identità per l'autenticazione degli sviluppatori sono le AWS credenziali di uno sviluppatore di pool di identità. I pool di identità non impongono alcuna restrizione alla validità degli identificatori sviluppatore-fornitore presenti in questo flusso di autenticazione.

Come best practice, implementa i provider di sviluppo solo nelle seguenti condizioni:

- Per creare la responsabilità per l'uso di credenziali autenticate dagli sviluppatori, create il nome e gli identificatori del provider di sviluppo in modo da indicare la fonte di autenticazione. Ad esempio: `"Logins" : {"MyCorp provider" : "[provider application ID]"}`.
- Evita le credenziali utente di lunga durata. Configura il tuo client lato server per richiedere identità con ruoli collegati ai servizi come i profili di istanza EC2 e i ruoli di esecuzione Lambda.
- Evita di mescolare fonti di fiducia interne ed esterne nello stesso pool di identità. Aggiungi il tuo provider di sviluppo e i provider Single Sign-On (SSO) in pool di identità separati.

Utilizzo di attributi per il controllo degli accessi

Gli attributi per il controllo degli accessi è l'implementazione dei pool di identità di Amazon Cognito del controllo degli accessi basato su attributi (ABAC). È possibile utilizzare le policy IAM per controllare l'accesso alle risorse AWS attraverso i pool di identità di Amazon Cognito in base agli attributi dell'utente. Questi attributi possono essere ricavati da provider di identità social e aziendali. È possibile eseguire la mappatura degli attributi all'interno dei token di accesso e ID dei provider o delle asserzioni SAML per i tag a cui è possibile fare riferimento nelle policy di autorizzazione IAM.

Puoi scegliere le mappature predefinite oppure creare le tue mappature personalizzate nei pool di identità di Amazon Cognito. Le mappature predefinite consentono di scrivere policy IAM in base a un set fisso di attributi utente. Le mappature personalizzate consentono invece di selezionare un set personalizzato di attributi utente a cui si fa riferimento nelle policy di autorizzazioni IAM. I nomi degli attributi nella console Amazon Cognito sono mappati alla chiave tag per principal, che sono i tag a cui si fa riferimento nella policy di autorizzazioni IAM.

Ad esempio, supponiamo che tu sia proprietario di un servizio di streaming multimediale con un abbonamento gratuito e uno pagamento. Puoi archiviare i file multimediali in Amazon S3 e taggarli con tag gratuiti o premium. Puoi utilizzare gli attributi per il controllo dell'accesso per consentire l'accesso a contenuti gratuiti e a pagamento in base al livello di appartenenza dell'utente, che è parte del profilo dell'utente. È possibile eseguire la mappatura dell'attributo di appartenenza per una chiave di tag per il principal da passare alla policy di autorizzazioni IAM. In questo modo puoi creare una singola policy di autorizzazioni e consentire in modo condizionale l'accesso al contenuto premium in base al valore del livello di appartenenza e del tag nei file di contenuto.

Argomenti

- Utilizzo degli attributi per il controllo dell'accesso con i pool di identità di Amazon Cognito
- Esempio di utilizzo di attributi per la policy di controllo degli accessi
- Disattivazione di attributi per il controllo degli accessi (console)
- Mappature di provider predefinite

L'uso degli attributi per controllare l'accesso offre molti vantaggi:

- La gestione delle autorizzazioni è molto più efficiente quando si utilizzano gli attributi per il controllo degli accessi. È possibile creare una policy di autorizzazioni di base che utilizzi attributi utente anziché creare più policy per funzioni di lavori diversi.

- Non è necessario aggiornare le policy ogni volta che si aggiungono o rimuovono risorse o utenti per l'applicazione. La politica delle autorizzazioni concede l'accesso solo agli utenti con gli attributi utente corrispondenti. Ad esempio, potrebbe essere necessario controllare l'accesso a determinati bucket S3 in base alla mansione degli utenti. In tal caso, puoi creare una policy di autorizzazioni per consentire l'accesso a questi file solo per gli utenti all'interno della mansione definita. Per ulteriori informazioni, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.
- Gli attributi possono essere passati come tag principali a una policy che consente o nega le autorizzazioni in base ai valori di tali attributi.

Utilizzo degli attributi per il controllo dell'accesso con i pool di identità di Amazon Cognito

Prima di utilizzare gli attributi per il controllo degli accessi, assicurati di soddisfare i seguenti prerequisiti:

- Un account AWS
- Pool di utenti
- Pool di identità
- Configurazione di un SDK
- Provider di identità integrati
- Credenziali

Per utilizzare attributi per il controllo degli accessi, la richiesta impostata come origine del set di dati imposta il valore della Chiave tag scelta. Amazon Cognito applica la chiave e il valore del tag alla sessione dell'utente. Le policy IAM possono valutare l'accesso dell'utente dalla condizione `${aws:PrincipalTag/tagkey}`. IAM valuta il valore del tag dell'utente rispetto alla policy.

È necessario preparare i ruoli IAM le cui credenziali si desidera passare agli utenti. La policy di attendibilità di questi ruoli deve consentire ad Amazon Cognito di assumere il ruolo per l'utente. Per quanto riguarda gli attributi per il controllo degli accessi, devi inoltre consentire ad Amazon Cognito di applicare i tag principali alla sessione temporanea dell'utente. Concedi il permesso di assumere il ruolo con l'azione AssumeRoleWithWebIdentity. Concedi l'autorizzazione per applicare tag alle sessioni degli utenti con l'azione basata solo sull'autorizzazione `sts:TagSession`. Per ulteriori informazioni, consulta Passare i tag di sessione in Servizio di token di sicurezza AWS nella Guida

per l'utente di AWS Identity and Access Management . Per una policy di attendibilità che concede le autorizzazioni `sts:AssumeRoleWithWebIdentity` e `sts:TagSession` al principale del servizio Amazon Cognito`cognito-identity.amazonaws.com`, consulta Esempio di utilizzo di attributi per la policy di controllo degli accessi.

Important

Se mappi gli attributi personalizzati da un pool di utenti di Amazon Cognito ai tag principali, assicurati che gli utenti finali non possano modificare tali attributi. Un attributo tramite il quale un utente può modificare non `UpdateUserAttributes` deve fungere da limite di sicurezza nelle politiche IAM. Per impedire la modifica da parte dell'utente, esegui una delle seguenti operazioni:

- Rimuovi l'attributo da ogni client WriteAttributesdi app nel tuo pool di utenti, non solo dal client di app che hai configurato nel tuo pool di identità.
- Definisci l'attributo come `Mutable: false` quando lo aggiungi allo schema del pool di utenti.

Gli attributi utente sono condivisi da tutti i client dell'app in un pool di utenti. Se un utente modifica un attributo tramite qualsiasi client di app, il nuovo valore viene visualizzato nei token ID di quell'utente provenienti da ogni client dell'app.
Quando non specificate un client `WriteAttributes` di app, il client dell'app ha accesso in scrittura a tutti gli attributi standard e personalizzati. Specificate `WriteAttributes` per ogni app client che create, compresi i client di app che aggiungerete in seguito.
Gli utenti non possono modificare un attributo immutabile dopo la creazione del loro profilo utente. Tuttavia, possono impostarne il valore al momento della registrazione. Se il tuo pool di utenti consente l'iscrizione in modalità self-service, rimuovi l'attributo da ogni `WriteAttributes` client dell'app. Poi esegui una delle seguenti operazioni:

- Imposta il valore quando crei utenti con. `AdminCreateUser`
- Rifiuta le richieste di iscrizione che contengono valori imprevisti. Per ulteriori informazioni, consulta the section called “Pre-iscrizione”.

Inoltre, assicurati che i valori degli attributi mappati ai tag principali non contengano caratteri che hanno un significato speciale nella valutazione delle politiche IAM, come * (asterisco) o ? (punto interrogativo). Negli `Resource` elementi delle policy IAM e negli operatori di `StringLike` condizione, questi caratteri funzionano come caratteri jolly. Se imposti il valore

di un attributo * e lo mappi su un tag principale, la ${aws:PrincipalTag/*tagkey*} valutazione potrebbe corrispondere a più risorse del previsto. Ciò può consentire l'accesso a risorse appartenenti ad altri inquilini o progetti. Convalida i valori degli attributi nella logica dell'applicazione. In alternativa, utilizza un trigger Lambda precedente alla generazione di token per rifiutare o disinfettare i valori che contengono questi caratteri prima che l'applicazione li passi come tag principali.
Se l'applicazione richiede che l'attributo rimanga scrivibile dall'utente per altri scopi, utilizza un trigger Lambda precedente alla generazione del token per convalidare o sostituire l'attestazione prima che venga inclusa nel token ID.

Per configurare gli attributi per il controllo degli accessi nella console

1. Accedi alla console di Amazon Cognito e seleziona Pool di identità. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Individua Provider di identità. Scegli il provider di identità da modificare. Se desideri aggiungere un nuovo IdP, seleziona Aggiungi provider di identità.
4. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, scegli Modifica in Attributi per il controllo degli accessi.
    a. Per non applicare alcun tag principale, scegli Inattivo.
    b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.
    c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.
5. Seleziona Salva modifiche.

Esempio di utilizzo di attributi per la policy di controllo degli accessi

Consideriamo uno scenario in cui un dipendente dell'ufficio legale di un'azienda deve riportare nei bucket tutti i file che appartengono al suo reparto e classificarli con il loro livello di sicurezza. Si supponga che il token che questo dipendente ottiene dal provider di identità contenga le seguenti attestazioni.

Attestazioni

```
{
            "...": "other claims",
            "sub" : "57e7b692-4f66-480d-98b8-45a6729b4c88",
            "department" : "legal",
            "clearance" : "confidential",
            "...": "other claims"
        }
```

È possibile eseguire la mappatura di questi attributi per i tag e si può fare riferimento ad essi nelle policy di autorizzazioni IAM come tag principal. Adesso puoi possibile gestire l'accesso modificando il profilo utente sul provider di identità. In alternativa, puoi modificare gli attributi sul lato risorsa utilizzando nomi o tag senza modificare la policy stessa.

La seguente policy di autorizzazioni esegue due operazioni:

- Consente l'accesso all'elenco a tutti i bucket S3 che terminano con un prefisso che corrisponde al nome del reparto dell'utente.
- Consente l'accesso in lettura ai file in questi bucket, purché il tag di autorizzazione sul file corrisponda all'attributo di autorizzazione dell'utente.

Policy delle autorizzazioni

JSON

```
 {
  "Version":"2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": "s3:List*",
          "Resource": "arn:aws:s3:::*-${aws:PrincipalTag/department}"
      },
      {
          "Effect": "Allow",
          "Action": "s3:GetObject*",
          "Resource": "arn:aws:s3:::*-${aws:PrincipalTag/department}/*",
          "Condition": {
```

```
                "StringEquals": {
                    "s3:ExistingObjectTag/clearance": "${aws:PrincipalTag/
clearance}"
                }
            }
        }
    ]
}
```

La policy di attendibilità determina chi può assumere questo ruolo. La policy di relazione di attendibilità consente l'utilizzo di `sts:AssumeRoleWithWebIdentity` e `sts:TagSession` per consentire l'accesso. Aggiunge condizioni per limitare la policy al pool di identità creato e garantisce che sia per un ruolo autenticato.

Policy di trust

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRoleWithWebIdentity",
        "sts:TagSession"
      ],
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "IDENTITY-POOL-ID"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
```

```
  ]
}
```

Disattivazione di attributi per il controllo degli accessi (console)

Segui questa procedura per disattivare gli attributi per il controllo degli accessi.

Per disattivare gli attributi per il controllo degli accessi nella console

1. Accedi alla console di Amazon Cognito e seleziona Pool di identità. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Individua Provider di identità. Scegli il provider di identità da modificare.
4. Scegli Modifica in Attributi per il controllo degli accessi.
5. Per non applicare alcun tag principale, scegli Inattivo.
6. Seleziona Salva modifiche.

Mappature di provider predefinite

Nella tabella seguente sono riportate le informazioni di mappatura predefinite per i provider di autenticazione supportati da Amazon Cognito.

Provider	Tipo di token	Valori dei tag principal	Esempio
Bacino d'utenza di Amazon Cognito	Token ID	aud(client ID) and sub(user ID)	"6jk8ltokc7ac9es6jrtg9q572f", "57e7b692-4f66-480d-98b8-45a6729b4c88"
Facebook	Token di accesso	aud(app_id), sub(user_id)	"492844718097981", "112177216992379"
Google	Token ID	aud(client ID) and sub(user ID)	"620493171733-eebk7c0hcp5lj3e1tlqp1gntt3k0rncv.apps.googleusercontent.com",

Provider	Tipo di token	Valori dei tag principal	Esempio
			"109220063452404746097"
SAML	Asserzioni	"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier" , "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"	"auth0|5e28d196f8f55a0eaaa95de3", "user123@gmail.com"
Apple	Token ID	aud(client ID) and sub (user ID)	"com.amazonaws.ec2-54-80-172-243.compute-1.client", "001968.a6ca34e9c1e742458a26cf8005854be9.0733"
Amazon	Token di accesso	aud (Client ID on Amzn Dev Ac), user_id(user ID)	"amzn1.application-oa2-client.9d70d9382d3446108aaee3dd763a0fa6", "amzn1.account.AGHNIFJQMFSBG3G6XCPVB35ORQAA"
Provider OIDC standard	Token ID e token di accesso	aud (as client_id), sub (as user ID)	"620493171733-eebk7c0hcp5lj3e1tlqp1gntt3k0rncv.apps.googleusercontent.com", "109220063452404746097"

Provider	Tipo di token	Valori dei tag principal	Esempio
Twitter	Token di accesso	aud (app ID; app Secret), sub (user ID)	«DfwifTtKEX1FiIBRn OTIR0CFK; «, Xgj5xb8xIrIVCPjXgL IdkW7fXmw cJJrFvnoK9gwZkLexo 1y5z1 «12690038 84292222976"
DevAuth	Eseguire la mappatura	Non applicabile	"tag1", "tag2"

> **Note**
>
> L'opzione di mappatura degli attributi predefinita viene popolata automaticamente per Chiave tag per principal e Nomi attributo. Non è possibile modificare le mappature predefinite.

Utilizzo del controllo degli accessi basato su ruoli

I pool di identità di Amazon Cognito assegnano agli utenti autenticati una serie di credenziali temporanee con privilegi limitati per accedere alle tue risorse. AWS Le autorizzazioni di ciascun utente vengono controllate attraverso i ruoli IAM che hai creato. Puoi definire le regole per scegliere il ruolo di ogni utente sulla base delle richieste nel token ID dell'utente. Puoi definire un ruolo di default per gli utenti autenticati. Puoi inoltre definire un ruolo IAM separato con autorizzazioni limitate per gli utenti guest non autenticati.

Creazione dei ruoli per la mappatura dei ruoli

È importante aggiungere la policy di attendibilità appropriata per ogni ruolo, in modo che possa essere assunta da Amazon Cognito solo per gli utenti autenticati nel pool di identità. Di seguito viene riportato un esempio di suddetta policy di attendibilità:

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-corner-
cafe-123456790ab"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}
```

Questa policy consente agli utenti federati di `cognito-identity.amazonaws.com` (approvatore del token OpenID Connect) di assumere questo ruolo. Inoltre la policy limita l'`aud` del token, in questo caso l'ID dei pool di identità, in base al pool di identità. Infine, la policy specifica che uno dei membri dell'array dell'attestazione `amr` con più valori del token emesso dall'operazione API `GetOpenIdToken` Amazon Cognito ha il valore `authenticated`.

Concessione delle autorizzazioni per il passaggio di ruoli

Per consentire a un utente di impostare ruoli con autorizzazioni superiori rispetto alle autorizzazioni dell'utente esistenti su un pool di identità, concedere loro l'autorizzazione `iam:PassRole` per trasferire il ruolo all'API `set-identity-pool-roles`. Ad esempio, se l'utente non è in grado di scrivere su Amazon S3 ma il ruolo IAM che l'utente imposta nel pool di identità concede l'autorizzazione per scrivere su Amazon S3, l'utente può impostare questo ruolo solo se a quest'ultimo viene concessa l'autorizzazione `iam:PassRole`. Il seguente esempio di policy mostra come concedere l'autorizzazione `iam:PassRole`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::123456789012:role/myS3WriteAccessRole"
            ]
        }
    ]
}
```

In questo esempio di policy viene concessa l'autorizzazione `iam:PassRole` al ruolo `myS3WriteAccessRole`. Il ruolo viene specificato usando l'Amazon Resource Name (ARN) del ruolo. È necessario, inoltre, collegare questa policy all'utente. Per ulteriori informazioni, consulta la sezione relativa all'utilizzo di policy gestite.

> **Note**
>
> Le funzioni Lambda usano una policy basata sulle risorse collegata direttamente alla funzione Lambda stessa. Quando crei una regola che invoca una funzione Lambda, non trasferisci un ruolo; pertanto, l'utente che crea la regola non richiede l'autorizzazione `iam:PassRole`. Per ulteriori informazioni sull'autorizzazione della funzione Lambda, consulta Gestione delle autorizzazioni: uso di una policy della funzione Lambda.

Utilizzo dei token per l'assegnazione dei ruoli agli utenti

Per gli utenti che effettuano l'accesso tramite il bacino d'utenza Amazon Cognito, i ruoli possono essere trasferiti nel token ID assegnato dal bacino d'utenza. I ruoli vengono visualizzati nelle seguenti attestazioni nel token ID:

- L'attestazione `cognito:preferred_role` è il ruolo ARN.

- L'attestazione `cognito:roles` è una stringa separata da virgole contenente una serie di ARN del ruolo consentite.

Le attestazioni sono impostate come segue:

- L'attestazione `cognito:preferred_role` è impostata sul ruolo del gruppo con il valore `Precedence` migliore (più basso). Se viene consentito un solo ruolo, `cognito:preferred_role` è impostato su tale ruolo. Se sono presenti più ruoli e nessun ruolo singolo ha la migliore precedenza, l'attestazione non è stata impostata.
- L'attestazione `cognito:roles` è impostata se è presente almeno un ruolo.

Quando si utilizzano i token per l'assegnazione dei ruoli, se sono presenti più ruoli che possono essere assegnati all'utente, i pool di identità di Amazon Cognito (identità federate) selezionano il ruolo come segue:

- Utilizza il GetCredentialsForIdentity`CustomRoleArn`parametro se è impostato e corrisponde a un ruolo nel claim. `cognito:roles` Se questo parametro non corrisponde a un ruolo in `cognito:roles`, nega l'accesso.
- Se l'attestazione `cognito:preferred_role` è impostata, utilizzala.
- Se l'`cognito:preferred_role`attestazione non è impostata, viene impostata e non `CustomRoleArn` è specificata nella chiamata a`GetCredentialsForIdentity`, quindi l'impostazione della risoluzione del ruolo nella console o nel `AmbiguousRoleResolution` campo (nel `RoleMappings` parametro dell'SetIdentityPoolRolesAPI) viene utilizzata per determinare il ruolo da assegnare. `cognito:roles`

Utilizzo di una mappatura basata su regole per assegnare i ruoli agli utenti

Le regole ti permettono di eseguire la mappatura delle attestazioni del token di un provider di identità per i ruoli IAM.

Ogni regola specifica l'attestazione di un token (ad esempio un attributo utente nel token ID di un bacino d'utenza di Amazon Cognito), il tipo di corrispondenza, un valore e un ruolo IAM. Il tipo di corrispondenza può essere `Equals`, `NotEqual`, `StartsWith`, o `Contains`. Se un utente dispone di un valore corrispondente all'attestazione, tale utente può assumere quel ruolo nel momento in cui ottiene le credenziali. Ad esempio, puoi creare una regola che assegna un determinato ruolo IAM agli utenti con un valore di attributo personalizzato `custom:dept` di `Sales`.

> **Note**
>
> Nelle impostazioni delle regole, gli attributi personalizzati richiedono il prefisso `custom:` per distinguersi dagli attributi standard.

Le regole vengono valutate in ordine e viene utilizzato il ruolo IAM per la prima regola corrispondente, a meno che non sia specificato `CustomRoleArn` per sovrascrivere l'ordine. Per ulteriori informazioni sugli attributi utente nei bacini d'utenza di Amazon Cognito, consulta Utilizzo degli attributi utente.

Puoi impostare più regole per un provider di autenticazione nella console del pool di identità (identità federate). Le regole vengono applicate in ordine. Puoi trascinare le regole per modificarne l'ordine. La prima regola corrispondente ha la priorità. Se il tipo di corrispondenza è `NotEqual` e l'attestazione non esiste, la regola non viene valutata. Se nessuna delle regole corrisponde, l'impostazione Risoluzione del ruolo viene applicata su Utilizza ruolo autenticato predefinito o Nega richiesta.

Nell'API e nella CLI, puoi specificare il ruolo da assegnare quando nessuna regola corrisponde nel `AmbiguousRoleResolution` campo del RoleMappingtipo, che è specificato nel `RoleMappings` parametro dell'SetIdentityPoolRolesAPI.

Per aggiungere una mappatura basata su regole a un provider di identità nella console Amazon Cognito, aggiungi o aggiorna un IdP e seleziona Scegli ruolo con regole in Selezione ruolo. Da lì, puoi aggiungere regole che mappano le dichiarazioni del provider ai ruoli IAM.

Puoi configurare una mappatura basata su regole per i provider di identità nell'API AWS CLI o con il `RulesConfiguration` campo del tipo. RoleMapping È possibile specificare questo campo nel `RoleMappings` parametro dell'API. SetIdentityPoolRoles

Ad esempio, il AWS CLI comando seguente aggiunge una regola che assegna il ruolo `arn:aws:iam::123456789012:role/Sacramento_team_S3_admin` agli utenti nella tua sede di Sacramento che sono stati autenticati da OIDC IdP: `arn:aws:iam::123456789012:oidc-provider/myOIDCIdP`

```
aws cognito-identity set-identity-pool-roles --region us-east-1 --cli-input-json
 file://role-mapping.json
```

Contenuto di **`role-mapping.json`**:

```
{
```

```
    "IdentityPoolId": "us-east-1:12345678-corner-cafe-123456790ab",
    "Roles": {
        "authenticated": "arn:aws:iam::123456789012:role/myS3WriteAccessRole",
        "unauthenticated": "arn:aws:iam::123456789012:role/myS3ReadAccessRole"
    },
    "RoleMappings": {
        "arn:aws:iam::123456789012:oidc-provider/myOIDCIdP": {
            "Type": "Rules",
            "AmbiguousRoleResolution": "AuthenticatedRole",
            "RulesConfiguration": {
                "Rules": [
                    {
                        "Claim": "locale",
                        "MatchType": "Equals",
                        "Value": "Sacramento",
                        "RoleARN": "arn:aws:iam::123456789012:role/
Sacramento_team_S3_admin"
                    }
                ]
            }
        }
    }
}
```

Per ogni bacino d'utenza o un altro provider di autenticazione configurato per un pool di identità, puoi creare fino a 25 regole. Questo limite non è regolabile. Per ulteriori informazioni, consulta la sezione Quote in Amazon Cognito.

Attestazioni dei token da utilizzare nella mappatura basata su regole

Amazon Cognito

Un token ID Amazon Cognito è rappresentato come token web JSON (JWT). Il token contiene attestazioni relative all'identità dell'utente autenticato, ad esempio `name` `family_name`, e `phone_number`. Per ulteriori informazioni sulle attestazioni standard, consulta la specifica OpenID Connect. Oltre alle attestazioni standard, di seguito sono riportate le attestazioni aggiuntive specifiche di Amazon Cognito:

- `cognito:groups`
- `cognito:roles`
- `cognito:preferred_role`

Amazon

Le seguenti attestazioni, insieme ai valori possibili per dette attestazioni, possono essere utilizzate con Login with Amazon:

- `iss`: www.amazon.com
- `aud`: Id app
- `sub`: `sub` dal token di Login with Amazon

Facebook

Le seguenti attestazioni, insieme ai valori possibili per dette attestazioni, possono essere utilizzate con Facebook:

- `iss`: graph.facebook.com
- `aud`: Id app
- `sub`: `sub` dal token di Facebook

Google

Un token di Google contiene attestazioni standard della specifica OpenID Connect. Tutte le attestazioni nel token di OpenID sono disponibili per la mappatura basata su regole. Consulta il sito OpenID Connect di Google per scoprire quali attestazioni sono disponibili nel token di Google.

Apple

Un token Apple contiene attestazioni standard della specifica OpenID Connect. Consulta Authenticating Users with Sign in with Apple (Autenticazione degli utenti con Accedi con Apple) nella documentazione di Apple per ulteriori informazioni sulle attestazioni disponibili tramite token Apple. Il token di Apple non sempre contiene `email`.

OpenID

Tutte le attestazioni nel token di Open Id sono disponibili per la mappatura basata su regole. Per ulteriori informazioni sulle attestazioni standard, consulta la specifica OpenID Connect. Consulta la documentazione relativa al provider di OpenID per scoprire le altre attestazioni aggiuntive disponibili.

SAML

Le attestazioni vengono ricavate analizzando l'asserzione SAML ricevuta. Tutte le attestazioni disponibili nell'asserzione SAML possono essere utilizzate nella mappatura basata su regole.

Best practice per il controllo accessi basato sui ruoli

> **Important**
>
> Se l'attestazione che si sta mappando a un ruolo può essere modificata dall'utente finale, qualsiasi utente finale può assumere il ruolo e impostare la policy di conseguenza. Esegui la mappatura solo delle attestazioni che non possono essere direttamente impostate dall'utente finale ai ruoli con autorizzazioni elevate. In un bacino d'utenza di Amazon Cognito puoi impostare le autorizzazioni di lettura e di scrittura per app per ogni attributo utente.

> **Important**
>
> Se imposti i ruoli per i gruppi in un bacino d'utenza di Amazon Cognito, tali ruoli vengono trasmessi tramite il token ID dell'utente. Per utilizzare questi ruoli, è necessario impostare anche Choose role from token (Scegli ruolo dal token) per la selezione del ruolo autenticato per il pool di identità.
>
> Puoi utilizzare l'impostazione della risoluzione dei ruoli nella console e il `RoleMappings` parametro dell'SetIdentityPoolRolesAPI per specificare qual è il comportamento predefinito quando il ruolo corretto non può essere determinato dal token.

Ottenere le credenziali

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione, in modo che gli utenti possano accedere alle risorse. AWS In questa sezione viene descritto come ottenere le credenziali e come recuperare un'identità di Amazon Cognito da un pool di identità.

Amazon Cognito supporta sia le identità autenticate che le identità non autenticate. L'identità degli utenti non autenticati non è verificata. Ciò rende questo ruolo appropriato per utenti guest dell'app o per i casi in cui non importa se l'identità degli utenti è verificata. Gli utenti autenticati accedono all'applicazione tramite un provider di identità di terza parte o un bacino d'utenza, che verifica la loro identità. Assicurati di creare l'ambito delle autorizzazioni di risorse in modo appropriato per evitare che utenti non autenticati possano accedere a esse.

Le identità di Amazon Cognito non sono credenziali. Vengono scambiate con credenziali utilizzando il supporto per la federazione delle identità Web disponibile in (). Servizio di token di sicurezza AWS AWS STS Per ottenere AWS le credenziali per gli utenti della tua app, utilizza il provider di credenziali del pool di identità Cognito del tuo AWS SDK. Ad esempio, nell' AWS SDK for JavaScript (v3), utilizzate la funzione inclusa nel pacchetto. `fromCognitoIdentityPool` `@aws-sdk/credential-providers` L'identità viene quindi scambiata con credenziali utilizzando. AWS STS

I provider di credenziali di identità web fanno parte della catena di provider di credenziali predefinita negli SDK AWS . Per impostare il token del pool di identità in un `config` file locale per un AWS SDK o il AWS CLI, aggiungi una `web_identity_token_file` voce di profilo. Consulta Assume il ruolo di fornitore di credenziali nella Guida di riferimento agli AWS SDK e agli strumenti.

Per ulteriori informazioni su come compilare le credenziali di identità web nell'SDK, fai riferimento alla guida per gli sviluppatori di SDK. Per ottenere i migliori risultati, inizia il tuo progetto con l'integrazione del pool di identità integrata in. AWS Amplify

AWS Risorse SDK per ottenere e impostare credenziali con pool di identità

- Federazione del pool di identità (Android) in Amplify Dev Center
- Federazione del pool di identità (iOS) in Amplify Dev Center
- Utilizzo di Amazon Cognito Identity per autenticare gli utenti nella Developer Guide AWS SDK per JavaScript
- Provider di credenziali Amazon Cognito nella Developer Guide AWS SDK per .NET
- Specificate le credenziali in modo programmatico nella Guida per gli sviluppatori AWS SDK per Go
- Fornisci credenziali temporanee nel codice contenuto nella Guida per gli sviluppatori AWS SDK for Java 2.x
- assumi come RoleWithWebIdentityCredentialProvider provider nella AWS SDK per PHP Developer Guide
- Assumere il ruolo con provider di identità web nella documentazione AWS SDK per Python (Boto3)
- Specificare le credenziali e la regione predefinita nella Developer Guide AWS SDK per Rust

Le sezioni seguenti forniscono esempi di codice in alcuni SDK precedenti AWS .

Android

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione, in modo che gli utenti possano accedere alle risorse. AWS Amazon Cognito supporta

sia le identità autenticate sia le identità non autenticate. Per fornire AWS le credenziali alla tua app, segui questi passaggi.

Per utilizzare un pool di identità di Amazon Cognito in un'app Android, configura. AWS Amplify Per ulteriori informazioni, consultare Autenticazione in Amplify Dev Center.

Recupero di una identità di Amazon Cognito

Se autorizzi gli utenti non autenticati, puoi recuperare immediatamente un identificatore univoco di Amazon Cognito (ID identità) per l'utente finale. Se esegui l'autenticazione degli utenti, puoi recuperare l'ID identità dopo aver impostato i token di accesso nel provider di credenziali:

```
String identityId = credentialsProvider.getIdentityId();
Log.d("LogTag", "my ID is " + identityId);
```

> **Note**
>
> Non chiamare `getIdentityId()`, `refresh()` o `getCredentials()` nel thread principale dell'applicazione. A partire da Android 3.0 (API Level 11), l'app fallirà automaticamente e genererà un errore NetworkOnMainThreadExceptionse esegui una rete I/O sul thread principale dell'applicazione. È necessario spostare il codice in un thread in background utilizzando `AsyncTask`. Per ulteriori informazioni, consulta la documentazione di Android. Puoi inoltre chiamare `getCachedIdentityId()` per recuperare un ID, ma solo se ce n'è già uno memorizzato nella cache in locale. In caso contrario, il metodo restituirà un valore nullo.

iOS - Objective-C

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione, in modo che gli utenti possano accedere alle risorse. AWS I pool di identità di Amazon Cognito supportano sia le identità autenticate che le identità non autenticate. Per fornire AWS le credenziali alla tua app, completa i seguenti passaggi.

Per utilizzare un pool di identità di Amazon Cognito in un'app iOS, configura. AWS Amplify Per ulteriori informazioni, consultare Swift Authentication e Flutter Authentication nella Amplify Dev Center.

Recupero di una identità di Amazon Cognito

Puoi recuperare immediatamente un identificatore di identità univoco di Amazon Cognito (ID identità) per l'utente finale se autorizzi gli utenti non autenticati o dopo aver impostato i token di accesso nel fornitore di credenziali se autentichi gli utenti:

```
// Retrieve your Amazon Cognito ID
[[credentialsProvider getIdentityId] continueWithBlock:^id(AWSTask *task) {
    if (task.error) {
        NSLog(@"Error: %@", task.error);
    }
    else {
        // the task result will contain the identity id
        NSString *cognitoId = task.result;
    }
    return nil;
}];
```

> **Note**
>
> `getIdentityId` è una chiamata asincrona. Se sul tuo provider è già impostato un ID identità, puoi chiamare `credentialsProvider.identityId` per recuperare tale identità, la quale è memorizzata nella cache in locale. Tuttavia, se sul tuo provider non è impostato un ID di identità, la chiamata `credentialsProvider.identityId` ritorna`nil`. Per ulteriori informazioni, consulta la documentazione di riferimento del'SDK per iOS.

iOS - Swift

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione in modo che gli utenti possano accedere alle risorse. AWS Amazon Cognito supporta sia le identità autenticate sia le identità non autenticate. Per fornire AWS le credenziali alla tua app, segui questi passaggi.

Per utilizzare un pool di identità di Amazon Cognito in un'app iOS, configura. AWS Amplify Per ulteriori informazioni, consultare Swift Authentication in Amplify Dev Center.

Recupero di una identità di Amazon Cognito

Puoi recuperare immediatamente un identificatore di identità univoco di Amazon Cognito (ID identità) per l'utente finale se autorizzi gli utenti non autenticati o dopo aver impostato i token di accesso nel fornitore di credenziali se autentichi gli utenti:

```
// Retrieve your Amazon Cognito ID
credentialsProvider.getIdentityId().continueWith(block: { (task) -> AnyObject? in
    if (task.error != nil) {
        print("Error: " + task.error!.localizedDescription)
    }
    else {
        // the task result will contain the identity id
        let cognitoId = task.result!
        print("Cognito id: \(cognitoId)")
    }
    return task;
})
```

> **Note**
>
> `getIdentityId` è una chiamata asincrona. Se sul tuo provider è già impostato un ID identità, puoi chiamare `credentialsProvider.identityId` per recuperare tale identità, la quale è memorizzata nella cache in locale. Tuttavia, se sul tuo provider non è impostato un ID di identità, la chiamata `credentialsProvider.identityId` ritorna`nil`. Per ulteriori informazioni, consulta la documentazione di riferimento del'SDK per iOS.

JavaScript

Se non lo hai già fatto, crea un pool di identità nella console Amazon Cognito prima di utilizzare `fromCognitoIdentityPool`.

Dopo aver configurato un pool di identità con i tuoi provider di identità, puoi utilizzare la `fromCognitoIdentityPool` funzione inclusa nel `@aws-sdk/credential-providers` pacchetto per autenticare gli utenti. Anziché impostare un valore globale`AWS.config.credentials`, collega il provider di credenziali direttamente a ciascun client di servizio tramite la relativa `credentials` proprietà. L'SDK v3 risolve pigramente il provider la prima volta che il client effettua una richiesta, quindi non è necessaria una chiamata separata per ottenere le credenziali. L'esempio seguente collega il provider a un: `S3Client`

```
import { S3Client } from "@aws-sdk/client-s3";
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";

// Set the region where your identity pool exists (for example, us-east-1 or eu-west-1)
const region = "REGION";
```

```
// Attach the credentials provider to your service client
const client = new S3Client({
  region,
  credentials: fromCognitoIdentityPool({
    identityPoolId: "IDENTITY_POOL_ID",
    logins: { // optional tokens, used for authenticated login
      "graph.facebook.com": "FBTOKEN",
      "www.amazon.com": "AMAZONTOKEN",
      "accounts.google.com": "GOOGLETOKEN",
      "appleid.apple.com": "APPLETOKEN"
    },
    clientConfig: { region },
  }),
});

// Credentials are resolved lazily when the client first makes a request.
// To inspect them directly, call the provider and await the result
// (from inside an async function):
const credentials = await client.config.credentials();
const { accessKeyId, secretAccessKey, sessionToken } = credentials;
```

La proprietà `logins` opzionale è una mappa di nomi di provider di identità ai token di identità per tali provider. Il modo in cui ottieni il token dal provider di identità dipende dal provider utilizzato. Ad esempio, se Facebook è uno dei provider di identità, puoi utilizzare la funzione `FB.login` di SDK di Facebook per ottenere un token del provider di identità:

```
FB.login(function (response) {
    if (response.authResponse) { // logged in
        const region = "us-east-1";
        const client = new S3Client({
          region,
          credentials: fromCognitoIdentityPool({
            identityPoolId: "us-east-1:1699ebc0-7900-4099-b910-2df94f52a030",
            logins: {
              "graph.facebook.com": response.authResponse.accessToken
            },
            clientConfig: { region },
          }),
        });

        console.log('You are now logged in.');
    } else {
```

```
        console.log('There was a problem logging you in.');
    }
});
```

Recupero di una identità di Amazon Cognito

Puoi recuperare un identificatore Amazon Cognito (ID di identità) univoco per il tuo utente finale. Se consenti utenti non autenticati, puoi recuperarlo immediatamente. Se autentichi gli utenti, recuperali dopo aver impostato i token di accesso. Il `fromCognitoIdentityPool` provider espone le `identityId` credenziali che risolve, quindi puoi leggerle dalle credenziali risolte di un client che utilizza il provider:

```
// Inside an async function:
const { identityId } = await client.config.credentials();
```

Unità

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione, in modo che gli utenti possano accedere alle risorse. AWS Amazon Cognito supporta sia le identità autenticate sia le identità non autenticate. Per fornire AWS le credenziali alla tua app, segui questi passaggi.

L'SDK AWS per Xamarin fa ora parte di SDK per .NET. Per iniziare a usare Amazon Cognito in SDK per .NET, consulta il fornitore di credenziali Amazon Cognito nella Developer Guide. AWS SDK per .NET Oppure consulta Amplify Dev Center per le opzioni con cui creare un'app. AWS Amplify

Recupero di una identità di Amazon Cognito

Se autorizzi gli utenti non autenticati, puoi recuperare immediatamente un identificatore di identità univoco di Amazon Cognito (ID identità) per l'utente finale. Se autentichi gli utenti, lo puoi recuperare dopo aver impostato i token di accesso nel fornitore di credenziali:

```
string identityId = await credentials.GetIdentityIdAsync();
```

Accesso Servizi AWS con credenziali temporanee

Il risultato di un'autenticazione riuscita con un pool di identità è un set di AWS credenziali. Con queste credenziali, l'applicazione può effettuare richieste a AWS risorse protette con l'autenticazione IAM. Con i vari AWS SDK che puoi aggiungere alle tue applicazioni per accedere alle operazioni API dei pool di identità, puoi effettuare richieste API non autenticate che producono credenziali temporanee.

Quindi puoi aggiungere SDK per altri utenti Servizi AWS al tuo client e firmare le richieste con quelle credenziali temporanee. Le autorizzazioni IAM concesse al tuo ruolo temporary-credentials devono consentire le operazioni che richiedi ad altri servizi.

Dopo aver configurato il provider di credenziali Amazon Cognito e recuperato le AWS credenziali, crea un client. Servizio AWS Di seguito sono riportati alcuni esempi tratti dalla documentazione SDK. AWS

AWS Risorse SDK per la creazione di un client

- AWS Configurazione del client nella Guida per gli AWS SDK per C++ sviluppatori
- Utilizzo della AWS SDK per Go versione 2 con la Servizi AWS Guida per gli AWS SDK per Go sviluppatori
- Configurazione dei client HTTP nella Developer Guide AWS SDK for Java 2.x
- Creazione e chiamata di oggetti di servizio nella AWS SDK per JavaScript Developer Guide
- Creazione di client nella AWS SDK per Python (Boto3) documentazione
- Creazione di un client di servizio nella Guida per AWS SDK per Rust gli sviluppatori
- Utilizzo dei client nella Guida per AWS SDK per Swift gli sviluppatori

Ad esempio, il seguente frammento inizializza un client Amazon DynamoDB:

Android

Per utilizzare un pool di identità di Amazon Cognito in un'app Android, configura. AWS Amplify Per ulteriori informazioni, consultare Autenticazione in Amplify Dev Center.

```
// Create a service client with the provider
AmazonDynamoDB client = new AmazonDynamoDBClient(credentialsProvider);
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

iOS - Objective-C

Per utilizzare un pool di identità di Amazon Cognito in un'app iOS, configura. AWS Amplify Per ulteriori informazioni, consultare Swift Authentication e Flutter Authentication nella Amplify Dev Center.

```
// create a configuration that uses the provider
AWSServiceConfiguration *configuration = [AWSServiceConfiguration
 configurationWithRegion:AWSRegionUSEast1 provider:credentialsProvider];
// get a client with the default service configuration
AWSDynamoDB *dynamoDB = [AWSDynamoDB defaultDynamoDB];
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

iOS - Swift

Per utilizzare un pool di identità di Amazon Cognito in un'app iOS, configura. AWS Amplify Per ulteriori informazioni, consultare Swift Authentication in Amplify Dev Center.

```
// get a client with the default service configuration
let dynamoDB = AWSDynamoDB.default()

// get a client with a custom configuration
AWSDynamoDB.register(with: configuration!, forKey: "USWest2DynamoDB");
let dynamoDBCustom = AWSDynamoDB(forKey: "USWest2DynamoDB")
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

JavaScript

```
// Create a service client with the provider
var dynamodb = new AWS.DynamoDB({region: 'us-west-2'});
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

Unità

L'SDK AWS per Xamarin fa ora parte di SDK per .NET. Per iniziare a usare Amazon Cognito in SDK per .NET, consulta il fornitore di credenziali Amazon Cognito nella Developer Guide. AWS SDK per .NET Oppure consulta Amplify Dev Center per le opzioni con cui creare un'app. AWS Amplify

```
// create a service client that uses credentials provided by Cognito
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials, REGION);
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

Pool di identità, provider di identità di terze parti

Con i pool di identità di Amazon Cognito, puoi integrarti con una varietà di provider di identità esterni (IdPs) per fornire AWS credenziali temporanee tramite l'autenticazione federata nella tua applicazione. Configurando il tuo pool di identità in modo che funzioni con queste risorse esterne IdPs, puoi autorizzare l'accesso alle AWS risorse di back-end per i tuoi utenti con l'autenticazione da parte di pool di utenti di Amazon Cognito, provider social, provider OIDC o provider SAML. Questa sezione illustra i passaggi per la configurazione e l'integrazione IdPs con il tuo pool di identità Amazon Cognito.

Con la proprietà `logins` puoi impostare le credenziali ricevute da un provider di identità. Puoi anche associare un pool di identità a più IdPs di uno. Ad esempio, puoi impostare i token di Facebook e Google nella proprietà `logins` per associare l'identità univoca di Amazon Cognito agli accessi di entrambi i provider di identità. L'utente può eseguire l'autenticazione con un dei due account, ma Amazon Cognito restituisce lo stesso identificatore utente.

Le seguenti istruzioni ti guidano nell'autenticazione con i pool di identità IdPs supportati dai pool di identità di Amazon Cognito.

Argomenti

- Configurazione di Facebook come pool di identità IdP
- Configurazione di Login with Amazon come IdP di pool di identità
- Configurazione di Google come IdP del pool di identità
- Configurazione di Accedi con Apple come IdP del pool di identità

- Configurazione di un provider OIDC come IdP del pool di identità
- Configurazione di un provider SAML come IdP del pool di identità

Configurazione di Facebook come pool di identità IdP

I pool di identità di Amazon Cognito funzionano con Facebook per fornire l'autenticazione federata agli utenti delle tue applicazioni. Questa sezione illustra come registrare e impostare la tua applicazione con Facebook come provider di identità.

Impostazione di Facebook

Prima di autenticare gli utenti di Facebook e interagire con le API di Facebook, è necessario registrare la tua applicazione su Facebook.

Il portale per gli sviluppatori di Facebook ti aiuta a configurare l'applicazione. Esegui questa procedura prima di integrare Facebook nel tuo pool di identità Amazon Cognito:

> **Note**
>
> La federazione dei pool di identità di Amazon Cognito non è compatibile con Facebook Limited Login. Per ulteriori informazioni su come configurare Facebook Login per iOS senza superare le autorizzazioni impostate per l'accesso limitato, vedi Facebook Login per iOS - Quickstart at Meta for Developers.

Configurazione di Facebook

1. Nel portale per sviluppatori di Facebook, esegui l'accesso con le credenziali di Facebook.
2. Nel menu Apps (App), scegli Add a New App (Aggiungi una nuova app).
3. Scegli una piattaforma e completa la procedura di avvio rapido.

Android

Per ulteriori informazioni su come integrare le app Android con Facebook Login, consulta la guida Primi passi di Facebook.

iOS - Objective-C

Per ulteriori informazioni su come integrare Objective-C le app iOS con Facebook Login, consulta la Guida introduttiva di Facebook.

iOS - Swift

Per ulteriori informazioni su come integrare le app iOS Swift con Facebook Login, consulta la guida Primi passi di Facebook.

JavaScript

Per ulteriori informazioni su come integrare le app JavaScript Web con Facebook Login, consulta la procedura di accesso di Facebook per il Web.

Configurazione di un provider di identità nella console dei pool di identità di Amazon Cognito

Per configurare il provider di identità, utilizza la procedura seguente.

Per aggiungere un gestore dell'identità digitale Facebook

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Facebook.
5. Inserisci l'ID app del progetto OAuth creato su Meta per sviluppatori. Per ulteriori informazioni, consultare Facebook Login nella documentazione Meta per sviluppatori.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

    - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

        i. Se hai scelto Scegli ruolo con regole, inserisci la dichiarazione di origine ricavata dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare l'affermazione, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

        ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Utilizzo di Facebook

Android

Per aggiungere l'autenticazione di Facebook, segui innanzitutto la guida di Facebook per integrare l'SDK di Facebook nella tua applicazione. Quindi aggiungi un pulsante Accedi con Facebook alla tua interfaccia utente di Android. L'SDK di Facebook utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito utilizza il token di accesso di questo oggetto di sessione per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

Una volta autenticato l'utente con l'SDK di Facebook, aggiungi il token di sessione al provider di credenziali di Amazon Cognito.

SDK di Facebook 4.0 o versione successiva:

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("graph.facebook.com", AccessToken.getCurrentAccessToken().getToken());
credentialsProvider.setLogins(logins);
```

Versione precedente a SDK di Facebook 4.0:

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("graph.facebook.com", Session.getActiveSession().getAccessToken());
credentialsProvider.setLogins(logins);
```

Il processo di accesso di Facebook inizializza una sessione singleton nell'SDK. L'oggetto sessione di Facebook contiene un token OAuth che Amazon Cognito utilizza per AWS generare credenziali per l'utente finale autenticato. Amazon Cognito usa inoltre il token per controllare la presenza di un utente corrispondente a questa particolare identità di Facebook nel tuo database degli utenti. Se l'utente esiste già, l'API restituisce l'identificatore esistente. In caso contrario, l'API restituisce un nuovo identificatore. Gli identificatori vengono automaticamente memorizzati nella cache dall'SDK client sul dispositivo locale.

> **Note**
>
> Dopo aver impostato la mappa degli accessi, effettua una chiamata o recupera `refresh` le credenziali. `get` AWS

iOS - Objective-C

Per aggiungere l'autenticazione di Facebook, segui innanzitutto la guida di Facebook per integrare l'SDK di Facebook nella tua applicazione. Quindi aggiungi un pulsante Accedi con Facebook alla tua interfaccia utente. L'SDK di Facebook utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito usa il token di accesso da questo oggetto di sessione per autenticare l'utente e associarlo a un pool di identità di Amazon Cognito univoco (identità federate).

Per fornire il token di accesso di Facebook ad Amazon Cognito, implementa il protocollo AWSIdentityProviderManager.

Quando implementi il metodo `logins`, viene restituito un dizionario contenente `AWSIdentityProviderFacebook`. Questo dizionario funge da chiave e il token di accesso corrente dell'utente Facebook autenticato funge da valore, come mostrato nell'esempio di codice seguente.

```
- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    FBSDKAccessToken* fbToken = [FBSDKAccessToken currentAccessToken];
    if(fbToken){
        NSString *token = fbToken.tokenString;
        return [AWSTask taskWithResult: @{ AWSIdentityProviderFacebook : token }];
    }else{
            return [AWSTask taskWithError:[NSError errorWithDomain:@"Facebook Login"
                                                              code:-1
                                                          userInfo:@{@"error":@"No current
 Facebook access token"}]];
```

```
    }
}
```

Quando crei istanze di `AWSCognitoCredentialsProvider`, trasmetti la classe che implementa `AWSIdentityProviderManager` come il valore di `identityProviderManager` nel costruttore. Per ulteriori informazioni, vai alla pagina AWSCognitoCredentialsProviderdi riferimento e scegli init WithRegionType:identityPoolId:identityProviderManager.

iOS - Swift

Per aggiungere l'autenticazione di Facebook, segui innanzitutto la guida di Facebook per integrare l'SDK di Facebook nella tua applicazione. Quindi aggiungi un pulsante Accedi con Facebook alla tua interfaccia utente. L'SDK di Facebook utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito usa il token di accesso da questo oggetto di sessione per autenticare l'utente e associarlo a un pool di identità di Amazon Cognito univoco (identità federate).

> **Note**
>
> La federazione dei pool di identità di Amazon Cognito non è compatibile con Facebook Limited Login. Per ulteriori informazioni su come configurare Facebook Login per iOS senza superare le autorizzazioni impostate per l'accesso limitato, vedi Facebook Login per iOS - Quickstart at Meta for Developers.

Per fornire il token di accesso di Facebook ad Amazon Cognito, implementa il protocollo AWSIdentityProviderManager.

Quando implementi il metodo `logins`, viene restituito un dizionario contenente `AWSIdentityProviderFacebook`. Questo dizionario funge da chiave e il token di accesso corrente dell'utente Facebook autenticato funge da valore, come mostrato nell'esempio di codice seguente.

```
class FacebookProvider: NSObject, AWSIdentityProviderManager {
    func logins() -> AWSTask<NSDictionary> {
        if let token = AccessToken.current?.authenticationToken {
            return AWSTask(result: [AWSIdentityProviderFacebook:token])
        }
        return AWSTask(error:NSError(domain: "Facebook Login", code: -1 , userInfo:
 ["Facebook" : "No current Facebook access token"]))
```

```
    }
}
```

Quando crei istanze di `AWSCognitoCredentialsProvider`, trasmetti la classe che implementa `AWSIdentityProviderManager` come il valore di `identityProviderManager` nel costruttore. Per ulteriori informazioni, vai alla pagina di AWSCognitoCredentialsProviderriferimento e scegli init. WithRegionType:identityPoolId:identityProviderManager

JavaScript

Per fornire l'autenticazione di Facebook, segui le istruzioni in Facebook Login per il web per aggiungere il pulsante Accedi con Facebook sul tuo sito Web. L'SDK di Facebook utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito utilizza il token di accesso di questo oggetto di sessione per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

Una volta autenticato l'utente con l'SDK di Facebook, aggiungi il token di sessione al provider di credenziali di Amazon Cognito.

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

FB.login(function (response) {

  // Check if the user logged in successfully.
  if (response.authResponse) {

    console.log('You are now logged in.');

    // Attach a Cognito identity pool credentials provider to your service client,
    // passing the Facebook access token in the logins map.
    const client = new S3Client({
      region: 'REGION',
      credentials: fromCognitoIdentityPool({
        identityPoolId: 'IDENTITY_POOL_ID',
        logins: {
          'graph.facebook.com': response.authResponse.accessToken
        },
        clientConfig: { region: 'REGION' }
      })
    });
```

```
    // The provider resolves and caches AWS credentials the first time the
    // client makes a request. Access AWS resources with client here.

  } else {
    console.log('There was a problem logging you in.');
  }

});
```

L'SDK di Facebook ottiene un token OAuth che Amazon Cognito utilizza per generare AWS credenziali per l'utente finale autenticato. Amazon Cognito usa inoltre il token per controllare la presenza di un utente corrispondente a questa particolare identità di Facebook nel tuo database utenti. Se l'utente esiste già, l'API restituisce l'identificatore esistente. In caso contrario, viene restituito un nuovo identificatore. Gli identificatori vengono automaticamente memorizzati nella cache dall'SDK client sul dispositivo locale.

> **Note**
>
> Il provider di credenziali risolve le credenziali pigramente la prima volta che il client di servizio AWS effettua una richiesta. Per ulteriori informazioni sull'utilizzo dei pool di identità di Amazon Cognito con l' AWS SDK for JavaScript, consulta la Guida per gli sviluppatori AWS SDK for JavaScript v3.

Unità

Per aggiungere l'autenticazione di Facebook, segui innanzitutto la guida di Facebook per integrare l'SDK di Facebook nella tua applicazione. Amazon Cognito usa il token di accesso di Facebook dall'oggetto FB per generare un identificatore utente univoco associato a un'identità di Amazon Cognito.

Una volta autenticato l'utente con l'SDK di Facebook, aggiungi il token di sessione al provider di credenziali di Amazon Cognito:

```
void Start()
{
    FB.Init(delegate() {
        if (FB.IsLoggedIn) { //User already logged in from a previous session
            AddFacebookTokenToCognito();
        } else {
```

```
            FB.Login ("email", FacebookLoginCallback);
        }
    });
}

void FacebookLoginCallback(FBResult result)
{
    if (FB.IsLoggedIn)
    {
        AddFacebookTokenToCognito();
    }
    else
    {
        Debug.Log("FB Login error");
    }
}

void AddFacebookTokenToCognito()
{
    credentials.AddLogin ("graph.facebook.com",
 AccessToken.CurrentAccessToken.TokenString);
}
```

Prima di utilizzare `FB.AccessToken`, chiama `FB.Login()` e assicurati che `FB.IsLoggedIn` sia true.

Configurazione di Login with Amazon come IdP di pool di identità

I pool di identità di Amazon Cognito funzionano con Login with Amazon per fornire l'autenticazione federata agli utenti delle app per dispositivi mobili e Web. Questa sezione spiega come registrare e impostare la tua applicazione con Login with Amazon come provider di identità.

Configura Login with Amazon per l'uso di Amazon Cognito nel portale per sviluppatori. Per ulteriori informazioni, consulta Configurazione di Login with Amazon nelle domande frequenti (FAQ) di Login with Amazon.

> **Note**
>
> Non puoi integrare in modo nativo Login with Amazon nella piattaforma Unity. Puoi utilizzare invece una visualizzazione Web e seguire il flusso di accesso del browser.

Configurazione di Login with Amazon

Implementazione di Login with Amazon

Nel portale per gli sviluppatori Amazon, puoi configurare un'applicazione OAuth da integrare con il tuo pool di identità, trovare la documentazione Login with Amazon e scaricare gli SDK. Scegli Developer console (Console per sviluppatori), quindi Login with Amazon nel portale per gli sviluppatori. Puoi creare un profilo di sicurezza per la tua applicazione e quindi costruire meccanismi di autenticazione Login with Amazon nella tua app. Consulta Ottenere le credenziali per ulteriori informazioni su come integrare l'autenticazione Login with Amazon con la tua app.

Amazon genera un ID client OAuth 2.0 per il tuo nuovo profilo di sicurezza. Puoi trovare l'ID client nella scheda Web Settings (Impostazioni Web) del profilo di sicurezza. Inserisci l'ID del profilo di sicurezza nel campo ID app dell'IdP Login with Amazon nel pool di identità.

> **Note**
>
> Inserisci l'ID del profilo di sicurezza nel campo ID app dell'IdP Login with Amazon nel pool di identità. Ciò differisce dai pool di utenti, che utilizzano l'ID client.

Configurazione del provider esterno nella console di Amazon Cognito

Per aggiungere un gestore dell'identità digitale Login with Amazon

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Scegli Login with Amazon.
5. Inserisci l'ID app del progetto OAuth creato su Login with Amazon. Per ulteriori informazioni, consultare la documentazione di Login with Amazon.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

    - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

        i. Se hai scelto Scegli ruolo con regole, inserisci la dichiarazione di origine ricavata dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare l'affermazione, il

valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Utilizzo di Login with Amazon: Android

Dopo aver autenticato Amazon Login, puoi passare il token al provider di credenziali Amazon Cognito nel metodo OnSuccess dell'interfaccia. TokenListener Il codice sarà il seguente:

```
@Override
public void onSuccess(Bundle response) {
    String token = response.getString(AuthzConstants.BUNDLE_KEY.TOKEN.val);
    Map<String, String> logins = new HashMap<String, String>();
    logins.put("www.amazon.com", token);
    credentialsProvider.setLogins(logins);
}
```

Usa Login with Amazon: iOS - Objective-C

Dopo aver autenticato l'accesso ad Amazon, puoi passare il token al provider di credenziali Amazon Cognito con il metodo di DidSucceed richiesta di: AMZNAccessTokenDelegate

```
- (void)requestDidSucceed:(APIResult \*)apiResult {
    if (apiResult.api == kAPIAuthorizeUser) {
```

```
        [AIMobileLib getAccessTokenForScopes:[NSArray arrayWithObject:@"profile"]
 withOverrideParams:nil delegate:self];
    }
    else if (apiResult.api == kAPIGetAccessToken) {
        credentialsProvider.logins = @{ @(AWSCognitoLoginProviderKeyLoginWithAmazon):
 apiResult.result };
    }
}}
```

Utilizzo di Login with Amazon: iOS - Swift

Una volta autenticato l'accesso ad Amazon, puoi passare il token al provider di credenziali di Amazon Cognito nel metodo `requestDidSucceed` di `AMZNAccessTokenDelegate`:

```
func requestDidSucceed(apiResult: APIResult!) {
    if apiResult.api == API.AuthorizeUser {
        AIMobileLib.getAccessTokenForScopes(["profile"], withOverrideParams: nil,
 delegate: self)
    } else if apiResult.api == API.GetAccessToken {
        credentialsProvider.logins =
 [AWSCognitoLoginProviderKey.LoginWithAmazon.rawValue: apiResult.result]
    }
}
```

Usa Login with Amazon: JavaScript

Dopo che l'utente ha effettuato l'autenticazione con Login with Amazon e viene reindirizzato al tuo sito Web, il token di accesso di Login with Amazon viene fornito nella stringa di query. Passa tale token alla mappa degli accessi delle credenziali.

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({
   region: 'REGION',
   credentials: fromCognitoIdentityPool({
       identityPoolId: 'IDENTITY_POOL_ID',
       logins: {
           'www.amazon.com': 'Amazon Access Token'
       },
       clientConfig: { region: 'REGION' }
   })
```

```
});
```

Configurazione di Google come IdP del pool di identità

I pool di identità di Amazon Cognito collaborano con Google per fornire l'autenticazione federata agli utenti delle tue applicazioni mobili. Questa sezione spiega come registrare e impostare la tua applicazione con Google come provider di identità.

Android

> **Note**
>
> Se l'app utilizza Google ed è disponibile su più piattaforme per dispositivi mobili, è necessario configurarla come provider OpenID Connect. Aggiungi tutti gli ID client creati come valori di destinatari aggiuntivi per consentire una migliore integrazione. Per ulteriori informazioni sul modello di identità cross-client di Google, consulta Identità. Cross-client

Configurazione di Google

Per attivare Google Sign-in per Android, crea un progetto di console Google Developers per la tua applicazione.

1. Vai alla console di Google Developers e crea un nuovo progetto.
2. Scegli APIs & Services (API e servizi), quindi OAuth consent screen (Schermata del consenso OAuth). Personalizza le informazioni che Google mostra agli utenti quando richiede il consenso per condividere i dati del profilo con la tua app.
3. Scegli Credentials (Credenziali), quindi Create credentials (Crea credenziali). Scegli OAuth client ID (ID client OAuth). Seleziona Android per Application type (Tipo di applicazione). Crea un ID client separato per ogni piattaforma in cui sviluppi la tua app.
4. Da Credentials (Credenziali), scegli Manage service accounts (Gestione degli account del servizio). Scegli Create service account (Crea account del servizio). Inserisci i dettagli dell'account del servizio, quindi scegli Create and continue (Crea e continua).
5. Concedi all'account del servizio l'accesso al tuo progetto. Concedi agli utenti l'accesso all'account del servizio come richiesto dall'app.
6. Scegli il tuo nuovo account del servizio, seleziona la scheda Keys (Chiavi) e Add key (Aggiungi chiave). Crea e scarica una nuova chiave JSON.

Per ulteriori informazioni su come utilizzare la console di Google Developers, consulta Creating and managing projects (Creazione e gestione di progetti) nella documentazione di Google Cloud.

Per ulteriori informazioni su come integrare Google nella tua app Android, consulta Autenticare gli utenti con Accedi con Google nella documentazione di Google Identity.

Per aggiungere un gestore dell'identità digitale di Google

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Google.
5. Inserisci l'ID cliente del progetto OAuth creato su Google Cloud Platform. Per ulteriori informazioni, consultare Configurazione di OAuth 2.0 della Guida in linea della console di Google Cloud Platform.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.
   - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.
     i. Se hai scelto Scegli ruolo con regole, inserisci la dichiarazione di origine dall'autenticazione dell'utente, l'operatore con cui desideri confrontare l'affermazione, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.
     ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.
7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.
   a. Per non applicare alcun tag principale, scegli Inattivo.
   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Utilizzo di Google

Per abilitare l'accesso con Google nella tua applicazione, segui le istruzioni della documentazione di Google per Android. Quando un utente esegue l'accesso, viene richiesto un token di autenticazione OpenID Connect a Google. Quindi, Amazon Cognito utilizza il token per autenticare l'utente e generare un identificatore univoco.

Il seguente codice di esempio illustra come recuperare il token di autenticazione da Google Play:

```
GooglePlayServicesUtil.isGooglePlayServicesAvailable(getApplicationContext());
AccountManager am = AccountManager.get(this);
Account[] accounts = am.getAccountsByType(GoogleAuthUtil.GOOGLE_ACCOUNT_TYPE);
String token = GoogleAuthUtil.getToken(getApplicationContext(), accounts[0].name,
        "audience:server:client_id:YOUR_GOOGLE_CLIENT_ID");
Map<String, String> logins = new HashMap<String, String>();
logins.put("accounts.google.com", token);
credentialsProvider.setLogins(logins);
```

iOS - Objective-C

> **Note**
>
> Se l'app utilizza Google ed è disponibile su più piattaforme per dispositivi mobili, è necessario configurare Google come provider OpenID Connect. Aggiungi tutti gli ID client creati come valori di destinatari aggiuntivi per consentire una migliore integrazione. Per ulteriori informazioni sul modello di identità cross-client di Google, consulta Cross-clientIdentità.

Configurazione di Google

Per abilitare Google Sign-in per iOS, crea un progetto di console Google Developers per la tua applicazione.

1. Vai alla console di Google Developers e crea un nuovo progetto.

2. Scegli APIs & Services (API e servizi), quindi OAuth consent screen (Schermata del consenso OAuth). Personalizza le informazioni che Google mostra agli utenti quando richiede il consenso per condividere i dati del profilo con la tua app.
3. Scegli Credentials (Credenziali), quindi Create credentials (Crea credenziali). Scegli OAuth client ID (ID client OAuth). Seleziona iOS per Application type (Tipo di applicazione). Crea un ID client separato per ogni piattaforma in cui sviluppi la tua app.
4. Da Credentials (Credenziali), scegli Manage service accounts (Gestione degli account del servizio). Scegli Create service account (Crea account del servizio). Inserisci i dettagli dell'account del servizio e scegli Create and continue (Crea e continua).
5. Concedi all'account del servizio l'accesso al tuo progetto. Concedi agli utenti l'accesso all'account del servizio come richiesto dall'app.
6. Scegli il tuo nuovo account del servizio. Scegli la scheda Keys (Chiavi) e Add key (Aggiungi chiave). Crea e scarica una nuova chiave JSON.

Per ulteriori informazioni su come utilizzare la console di Google Developers, consulta Creating and managing projects (Creazione e gestione di progetti) nella documentazione di Google Cloud.

Per ulteriori informazioni su come integrare Google nella tua app iOS, consulta Google Sign-In per iOS nella documentazione di Google Identity.

Per aggiungere un gestore dell'identità digitale di Google

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Google.
5. Inserisci l'ID cliente del progetto OAuth creato su Google Cloud Platform. Per ulteriori informazioni, consultare Configurazione di OAuth 2.0 della Guida in linea della console di Google Cloud Platform.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

    - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

i. Se hai scelto Scegli ruolo con regole, inserisci la dichiarazione di origine ricavata dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare l'affermazione, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Utilizzo di Google

Per abilitare l'accesso con Google nella tua applicazione, segui la documentazione Google per iOS. In caso di autenticazione corretta, si genera un token di autenticazione di OpenID Connect, utilizzato da Amazon Cognito per autenticare l'utente e generare un identificatore univoco.

In caso di autenticazione corretta, viene generato un oggetto `GTMOAuth2Authentication` contenente un `id_token`, utilizzato da Amazon Cognito per autenticare l'utente e generare un identificatore univoco:

```
- (void)finishedWithAuth: (GTMOAuth2Authentication *)auth error: (NSError *) error {
        NSString *idToken = [auth.parameters objectForKey:@"id_token"];
        credentialsProvider.logins = @{ @(AWSCognitoLoginProviderKeyGoogle): idToken };
    }
```

iOS - Swift

> **Note**
>
> Se l'app utilizza Google ed è disponibile su più piattaforme per dispositivi mobili, è necessario configurare Google come provider OpenID Connect. Aggiungi tutti gli ID client creati come valori di destinatari aggiuntivi per consentire una migliore integrazione. Per ulteriori informazioni sul modello di identità cross-client di Google, consulta Identità. Cross-client

Configurazione di Google

Per abilitare Google Sign-in per iOS, crea un progetto di console Google Developers per la tua applicazione.

1. Vai alla console di Google Developers e crea un nuovo progetto.
2. Scegli APIs & Services (API e servizi), quindi OAuth consent screen (Schermata del consenso OAuth). Personalizza le informazioni che Google mostra agli utenti quando richiede il consenso per condividere i dati del profilo con la tua app.
3. Scegli Credentials (Credenziali), quindi Create credentials (Crea credenziali). Scegli OAuth client ID (ID client OAuth). Seleziona iOS per Application type (Tipo di applicazione). Crea un ID client separato per ogni piattaforma in cui sviluppi la tua app.
4. Da Credentials (Credenziali), scegli Manage service accounts (Gestione degli account del servizio). Scegli Create service account (Crea account del servizio). Inserisci i dettagli dell'account del servizio e scegli Create and continue (Crea e continua).
5. Concedi all'account del servizio l'accesso al tuo progetto. Concedi agli utenti l'accesso all'account del servizio come richiesto dall'app.
6. Scegli il tuo nuovo account del servizio, seleziona la scheda Keys (Chiavi) e Add key (Aggiungi chiave). Crea e scarica una nuova chiave JSON.

Per ulteriori informazioni su come utilizzare la console di Google Developers, consulta Creating and managing projects (Creazione e gestione di progetti) nella documentazione di Google Cloud.

Per ulteriori informazioni su come integrare Google nella tua app iOS, consulta Google Sign-In per iOS nella documentazione di Google Identity.

Scegli Gestisci pool di identità dalla home page della console Amazon Cognito:

Configurazione del provider esterno nella console Amazon Cognito

1. Scegli il nome del pool di identità per cui vuoi abilitare Google come provider esterno. Viene visualizzata la pagina Dashboard (Pannello di controllo) per il tuo pool di identità.
2. Nell'angolo in alto a destra della pagina Dashboard (Pannello di controllo), scegli Edit identity pool (Modifica pool di identità). Viene visualizzata la pagina Edit identity pool (Modifica pool di identità).
3. Scorri verso il basso e scegli Authentication providers (Provider di autenticazione) per espandere la sezione.
4. Scegli la scheda Google .
5. Scegli Unlock (Sblocca).
6. Inserisci l'ID client ottenuto da Google, quindi scegli Save Changes (Salva le modifiche).

Utilizzo di Google

Per abilitare l'accesso con Google nella tua applicazione, segui la documentazione Google per iOS. In caso di autenticazione corretta, viene generato un token di autenticazione OpenID Connect, utilizzato da Amazon Cognito per autenticare l'utente e generare un identificatore univoco.

L'autenticazione completata restituisce un oggetto `GTMOAuth2Authentication` contenente un `id_token`. Amazon Cognito utilizza questo token per autenticare l'utente e generare un identificatore univoco:

```
func finishedWithAuth(auth: GTMOAuth2Authentication!, error: NSError!) {
    if error != nil {
      print(error.localizedDescription)
    }
    else {
      let idToken = auth.parameters.objectForKey("id_token")
      credentialsProvider.logins = [AWSCognitoLoginProviderKey.Google.rawValue:
 idToken!]
    }
}
```

JavaScript

> **Note**
>
> Se l'app utilizza Google ed è disponibile su più piattaforme per dispositivi mobili, è necessario configurare Google come provider OpenID Connect. Aggiungi tutti gli ID client creati come valori di destinatari aggiuntivi per consentire una migliore integrazione. Per ulteriori informazioni sul modello di identità cross-client di Google, consulta Cross-clientIdentità.

Configurazione di Google

Per abilitare Google Sign-in per un'app JavaScript web, crea un progetto di console Google Developers per la tua applicazione.

1. Vai alla console di Google Developers e crea un nuovo progetto.
2. Scegli APIs & Services (API e servizi), quindi OAuth consent screen (Schermata del consenso OAuth). Personalizza le informazioni che Google mostra agli utenti quando chiede il consenso a condividere i dati del profilo con la tua app.
3. Scegli Credentials (Credenziali), quindi Create credentials (Crea credenziali). Scegli OAuth client ID (ID client OAuth). Seleziona Web application (Applicazione Web) per Application type (Tipo di applicazione). Crea un ID client separato per ogni piattaforma in cui sviluppi la tua app.
4. Da Credentials (Credenziali), scegli Manage service accounts (Gestione degli account del servizio). Scegli Create service account (Crea account del servizio). Inserisci i dettagli dell'account del servizio e scegli Create and continue (Crea e continua).
5. Concedi all'account del servizio l'accesso al tuo progetto. Concedi agli utenti l'accesso all'account del servizio come richiesto dall'app.
6. Scegli il tuo nuovo account del servizio, seleziona la scheda Keys (Chiavi) e Add key (Aggiungi chiave). Crea e scarica una nuova chiave JSON.

Per ulteriori informazioni su come utilizzare la console di Google Developers, consulta Creating and managing projects (Creazione e gestione di progetti) nella documentazione di Google Cloud.

Per ulteriori informazioni su come integrare Google nella tua app Web, consulta Sign In With Google nella documentazione di Google Identity.

Configurazione del provider esterno nella console Amazon Cognito

Per aggiungere un gestore dell'identità digitale di Google

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Google.
5. Inserisci l'ID cliente del progetto OAuth creato su Google Cloud Platform. Per ulteriori informazioni, consultare Configurazione di OAuth 2.0 della Guida in linea della console di Google Cloud Platform.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.
   - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.
     i. Se hai scelto Scegli ruolo con regole, inserisci la dichiarazione di origine tratta dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare l'affermazione, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.
     ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.
7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.
   a. Per non applicare alcun tag principale, scegli Inattivo.
   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.
   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.
8. Seleziona Salva modifiche.

Utilizzo di Google

Per abilitare l'accesso con Google nella tua applicazione, segui la documentazione Google per Web.

In caso di autenticazione corretta, viene generato un oggetto di risposta contenente un `id_token`, utilizzato da Amazon Cognito per autenticare l'utente e generare un identificatore univoco:

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

function signinCallback(authResult) {
  if (authResult['status']['signed_in']) {

     // Attach a Cognito identity pool credentials provider to your service client,
     // passing the Google ID token in the logins map.
     const client = new S3Client({
        region: 'REGION',
        credentials: fromCognitoIdentityPool({
           identityPoolId: 'IDENTITY_POOL_ID',
           logins: {
              'accounts.google.com': authResult['id_token']
           },
           clientConfig: { region: 'REGION' }
        })
     });

     // The provider resolves and caches AWS credentials the first time the
     // client makes a request. Access AWS resources with client here.
  }
}
```

Configurazione di Accedi con Apple come IdP del pool di identità

I pool di identità di Amazon Cognito funzionano con Sign in with Apple per fornire l'autenticazione federata per gli utenti delle applicazioni mobili e delle applicazioni Web. Questa sezione spiega come registrare e configurare l'applicazione con Accedi con Apple come provider di identità.

Per aggiungere Accedi con Apple come provider di autenticazione a un pool di identità, devi completare due procedure. Innanzitutto devi integrare Accedi con Apple in un'applicazione, quindi configurare Accedi con Apple nei pool di identità. Per le informazioni più aggiornate sulla configurazione della funzionalità di accesso con Apple, consulta la sezione relativa alla configurazione dell'ambiente per la funzionalità di accesso con Apple nella documentazione per gli sviluppatori Apple.

Configurazione di Accedi con Apple

Per configurare Accedi con Apple come provider di identità, è necessario registrare l'applicazione con Apple per ricevere l'ID client.

1. Creazione di un account sviluppatore con Apple.
2. Accedi con le tue credenziali Apple.
3. Nel riquadro di navigazione a sinistra, scegliere Certificates, IDs & Profiles (Certificati, identificatori e profili).
4. Nel riquadro di navigazione a sinistra, scegliere Inputs (Input).
5. Nella pagina Identificatori scegli l'icona +.
6. Nella pagina Register a New Identifier (Registra un nuovo identificatore), scegli App IDs (ID app), quindi scegli Continue (Continua).
7. Nella pagina Register an App ID (Registra un'ID app), procedi come indicato di seguito:
    a. In Description (Descrizione), digitare una descrizione.
    b. In Bundle ID (ID pacchetto) digitare un identificatore. Prendi nota dell'ID bundle poiché avrai bisogno di questo valore per configurare Apple come provider nel pool di identità.
    c. In Capabilities (Funzionalità), scegli Sign In with Apple (Accedi con Apple), quindi seleziona Edit (Modifica).
    d. Nella pagina Accedi con Apple: configurazione dell'ID Apple, seleziona l'impostazione appropriata per l'app. Quindi scegli Save (Salva).
    e. Scegli Continua.
8. Nella pagina Confirm your App ID (Conferma l'ID app), scegli Register (Registra).
9. Passa alla fase 10 se desideri integrare Accedi con Apple con un'applicazione iOS nativa. La fase 11 è per le applicazioni che si desidera integrare con Accedi con Apple JS.
10. Nella pagina Identifiers (Identificatori), scegli il menu App IDs (ID app), quindi Services IDs (ID servizio). Scegli l'icona +.
11. Nella pagina Register a New Identifier (Registra un nuovo identificatore), seleziona Services IDs (ID servizi), quindi scegli Continue (Continua).
12. Nella pagina Register a Services ID (Registra un ID servizi), procedi nel modo seguente:
    a. In Description (Descrizione), digitare una descrizione.
    b. In Identifier (Identificatore), digitare un identificatore. Prendi nota dell'ID servizio poiché avrai bisogno di questo valore per configurare Apple come provider nel pool di identità.

c. Seleziona Accedi con Apple, quindi scegli Configura.

d. Nella pagina Web Authentication Configuration (Configurazione autenticazione Web), scegli un Primary App ID (ID app principale). In Website URLs (URL del sito Web), scegli l'icona +. Per Domini e sottodomini, inserire il nome di dominio dell'app. In Return URLs (URL restituiti) inserisci l'URL di callback a cui l'autorizzazione reindirizza l'utente dopo l'autenticazione con Accedi con Apple.

e. Scegli Avanti.

f. Scegli Continue (Continua), quindi scegli Register (Registra).

13. Nel riquadro di navigazione a sinistra, scegliere Chiavi.
14. Nella pagina Keys (Chiavi), scegli l'icona +.
15. Nella pagina Register a New Key, (Registra una nuova chiave) procedi nel modo seguente:

a. In Key Name (Nome chiave), digitare un nome della chiave.

b. Scegli Accedi con Apple, quindi scegli Configura.

c. Nella pagina Configura chiave, scegliere un ID app principale, quindi scegli Salva.

d. Scegli Continue (Continua), quindi scegli Register (Registra).

> **Note**
>
> Per integrare Accedi con Apple con un'applicazione iOS nativa, consulta Implementing User Authentication with Sign in with Apple (Implementazione dell'autenticazione utente con Accedi con Apple).
> Per integrare Accedi con Apple in una piattaforma diversa da iOS nativo, consulta Sign in with Apple JS (Accedi con Apple JS).

Configurazione del provider esterno nella console delle identità federate di Amazon Cognito

Per configurare il provider esterno, utilizzare la procedura seguente.

Per aggiungere un gestore dell'identità digitale Accedi con Apple

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.

4. Scegli Accedi con Apple.
5. Inserisci l'ID servizi del progetto OAuth creato con Apple Developer. Per ulteriori informazioni, consultare la pagina relativa all'autenticazione degli utenti con Accedi con Apple nella documentazione di Accedi con Apple.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

    - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

        i. Se hai scelto Scegli ruolo con regole, inserisci la dichiarazione di origine ricavata dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare l'affermazione, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

        ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.
7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

    a. Per non applicare alcun tag principale, scegli Inattivo.

    b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

    c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.
8. Seleziona Salva modifiche.

Accesso con Apple come provider negli esempi della CLI di identità federate di Amazon Cognito

In questo esempio viene creato un pool di identità denominato `MyIdentityPool` con Accedi con Apple come provider di identità.

```
aws cognito-identity create-identity-pool --identity-
pool-name MyIdentityPool --supported-login-providers
appleid.apple.com="sameple.apple.clientid"
```

Per ulteriori informazioni, consulta l'argomento relativo alla creazione di un pool di identità

Genera un ID identità di Amazon Cognito

Questo esempio genera (o recupera) un ID Amazon Cognito. Questa è un'API pubblica, quindi non sono necessarie credenziali per richiamare questa API.

```
aws cognito-identity get-id --identity-pool-id SampleIdentityPoolId --
logins appleid.apple.com="SignInWithAppleIdToken"
```

Per ulteriori informazioni, consulta get-id.

Ottenimento delle credenziali per un ID identità di Amazon Cognito

In questo esempio vengono restituite le credenziali per l'ID di identità fornito e per l'accesso con Accedi con Apple. Questa è un'API pubblica, quindi non sono necessarie credenziali per richiamare questa API.

```
aws cognito-identity get-credentials-for-identity --identity-id
SampleIdentityId --logins appleid.apple.com="SignInWithAppleIdToken"
```

Per ulteriori informazioni, consulta get-credentials-for-identity

Utilizzo di Accedi con Apple: Android

Apple non fornisce un SDK che supporta Accedi con Apple per Android. È invece possibile utilizzare il flusso Web in una visualizzazione Web.

- Per configurare Accedi con Apple nella tua applicazione, segui Configurazione della pagina Web per Accedi con Apple nella documentazione Apple.
- Per aggiungere un pulsante Accedi con Apple all'interfaccia utente Android, segui la sezione Visualizzazione dei pulsanti Accedi con Apple sul Web nella documentazione Apple.
- Per autenticare in modo sicuro gli utenti utilizzando il servizio Accedi con Apple, segui le indicazioni in Autenticazione degli utenti con Accedi con Apple nella documentazione Apple (lingua italiana non garantita).

L'accesso con Apple utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito utilizza il token ID di questo oggetto sessione per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

```
@Override
public void onSuccess(Bundle response) {
    String token = response.getString("id_token");
    Map<String, String> logins = new HashMap<String, String>();
    logins.put("appleid.apple.com", token);
    credentialsProvider.setLogins(logins);
}
```

Usa Accedi con Apple: iOS - Objective-C

Apple ha fornito il supporto SDK per Accedi con Apple nelle applicazioni iOS native. Per implementare l'autenticazione utente con Accedi con Apple nei dispositivi iOS nativi, seguire le indicazioni riportate in Implementing User Authentication with Sign in with Apple (Implementazione dell'autenticazione utente con Accedi con Apple) nella documentazione di Apple.

Amazon Cognito utilizza il token ID per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

```
(void)finishedWithAuth: (ASAuthorizationAppleIDCredential *)auth error: (NSError *)
 error {
        NSString *idToken = [ASAuthorizationAppleIDCredential
 objectForKey:@"identityToken"];
        credentialsProvider.logins = @{ "appleid.apple.com": idToken };
    }
```

Utilizzo di Accedi con Apple: iOS - Swift

Apple ha fornito il supporto SDK per Accedi con Apple nelle applicazioni iOS native. Per implementare l'autenticazione utente con Accedi con Apple nei dispositivi iOS nativi, seguire le indicazioni riportate in Implementing User Authentication with Sign in with Apple (Implementazione dell'autenticazione utente con Accedi con Apple) nella documentazione di Apple.

Amazon Cognito utilizza il token ID per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

Per ulteriori informazioni sulla configurazione di Accedi con Apple in iOS, consulta Configurazione di Accedi con Apple (lingua italiana non garantita).

```
func finishedWithAuth(auth: ASAuthorizationAppleIDCredential!, error: NSError!) {
    if error != nil {
```

```
      print(error.localizedDescription)
    }
    else {
      let idToken = auth.identityToken,
      credentialsProvider.logins = ["appleid.apple.com": idToken!]
    }
}
```

Usa Accedi con Apple: JavaScript

Apple non fornisce un SDK che supporti l'accesso con Apple per JavaScript. È invece possibile utilizzare il flusso Web in una visualizzazione Web.

- Per configurare Accedi con Apple nella tua applicazione, segui Configurazione della pagina Web per Accedi con Apple nella documentazione Apple.
- Per aggiungere un pulsante Accedi con Apple all'interfaccia JavaScript utente, segui la sezione Visualizzazione dei pulsanti Accedi con Apple sul Web nella documentazione Apple.
- Per autenticare in modo sicuro gli utenti utilizzando il servizio Accedi con Apple, segui le indicazioni in Autenticazione degli utenti con Accedi con Apple nella documentazione Apple (lingua italiana non garantita).

L'accesso con Apple utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito utilizza il token ID di questo oggetto sessione per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

function signinCallback(authResult) {
     // Attach a Cognito identity pool credentials provider to your service client,
     // passing the Apple ID token in the logins map.
     const client = new S3Client({
        region: 'REGION',
        credentials: fromCognitoIdentityPool({
           identityPoolId: 'IDENTITY_POOL_ID',
           logins: {
              'appleid.apple.com': authResult['id_token']
           },
           clientConfig: { region: 'REGION' }
        })
```

```
    });

    // The provider resolves and caches AWS credentials the first time the
    // client makes a request. Access AWS resources with client here.
}
```

Configurazione di un provider OIDC come IdP del pool di identità

OpenID Connect è uno standard aperto di autenticazione supportato da una serie di provider di accesso. Con Amazon Cognito, puoi collegare le identità ai provider OpenID Connect tramite i quali configuri. AWS Identity and Access Management

Aggiunta di un provider OpenID Connect

Per informazioni su come creare un provider OpenID Connect, consultare Creazione di provider di identità OpenID Connect (OIDC) nella Guida per l'utente di AWS Identity and Access Management .

Associazione di un fornitore ad Amazon Cognito

Per aggiungere un gestore dell'identità digitale OIDC

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Scegli OpenID Connect (OIDC).
5. Scegli un provider di identità OIDC dall'IAM del tuo. IdPs Account AWS Se desideri aggiungere un nuovo provider SAML, scegli Crea nuovo provider per accedere alla console IAM.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

    - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

        i. Se hai scelto Scegli ruolo con regole, inserisci il claim di origine tratto dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare l'attestazione, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

È possibile associare più provider OpenID Connect a un singolo pool di identità.

Utilizzo di OpenID Connect

Consulta la documentazione del tuo provider per sapere come accedere e ricevere un token ID.

Dopo aver ottenuto un token, aggiungilo alla mappa degli accessi. Usa l'URI del tuo provider come chiave.

Convalida di un token di OpenID Connect

Alla prima integrazione con Amazon Cognito, potresti ricevere un'eccezione `InvalidToken`. È importante comprendere in che modo Amazon Cognito convalida i token OpenID Connect (OIDC).

> **Note**
>
> Come specificato qui (https://tools.ietf.org/html/rfc7523), Amazon Cognito offre un periodo di prova di 5 minuti per gestire qualsiasi inclinazione del clock tra i sistemi.

1. Il parametro `iss` deve corrispondere alla chiave usata nella mappa degli accessi (ad esempio login.provider.com).
2. La firma deve essere valida. La firma deve essere verificabile mediante una chiave pubblica RSA.

> **Note**
>
> I pool di identità mantengono una cache della chiave di firma OIDC IdP per un breve periodo. Se il provider modifica la chiave di firma, Amazon Cognito potrebbe restituire un `NoKeyFound` errore fino all'aggiornamento della cache. Se riscontri questo errore, attendi una decina di minuti affinché il pool di identità aggiorni la chiave di firma.

3. L'impronta digitale della chiave pubblica del certificato corrisponde all'impronta digitale impostata in IAM al momento della creazione del provider OIDC.
4. In presenza del parametro `azp`, verifica questo valore rispetto agli ID client elencati nel provider OIDC.
5. In assenza del parametro `azp`, verifica il parametro `aud` rispetto agli ID client elencati nel provider OIDC.

Il sito Web jwt.io è una risorsa preziosa per la decodifica dei token e la verifica di questi valori.

Android

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("login.provider.com", token);
credentialsProvider.setLogins(logins);
```

iOS - Objective-C

```
credentialsProvider.logins = @{ "login.provider.com": token }
```

JavaScript

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({
 region: 'REGION',
 credentials: fromCognitoIdentityPool({
    identityPoolId: 'IDENTITY_POOL_ID',
    logins: {
       'login.provider.com': token
    },
```

```
    clientConfig: { region: 'REGION' }
 })
});
```

Configurazione di un provider SAML come IdP del pool di identità

Con i pool di identità di Amazon Cognito, puoi autenticare gli utenti con provider di identità (IdPs) tramite SAML 2.0. Puoi utilizzare un provider di identità che supporti il linguaggio SAML con Amazon Cognito per garantire un flusso di onboarding semplice per i tuoi utenti. Il tuo SAML-supporting IdP specifica i ruoli IAM che gli utenti possono assumere. In questo modo, diversi utenti possono ricevere diversi set di autorizzazioni.

Configurazione del pool di identità per un provider di identità SAML

I passaggi seguenti descrivono come configurare il pool di identità per l'utilizzo di un SAML-based IdP.

> **Note**
>
> Prima di configurare il pool di identità per supportare un provider SAML, è necessario configurare il provider di identità SAML nella console IAM. Per ulteriori informazioni, consulta Integrazione di provider di soluzioni SAML di terze parti con AWS nella Guida per l'utente di IAM.

Per aggiungere un gestore dell'identità digitale SAML

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Scegli SAML.
5. Scegli un provider di identità SAML dall'IAM del tuo IdPs . Account AWS Se desideri aggiungere un nuovo provider SAML, scegli Crea nuovo provider per accedere alla console IAM.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

    - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

i. Se hai scelto Scegli ruolo con regole, inserisci la dichiarazione di origine ricavata dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare l'affermazione, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Configurazione di un provider di identità SAML

Una volta creato, configura il provider di identità SAML in modo da aggiungere una relazione di trust tra il provider di identità e AWS. Con molti IdPs, puoi specificare un URL che l'IdP può utilizzare per leggere le informazioni e i certificati del relying party da un documento XML. Perché AWS, puoi usare. https://signin.aws.amazon.com/static/saml-metadata.xml Il passaggio successivo consiste nel configurare la risposta all'asserzione SAML del tuo IdP per compilare le attestazioni necessarie. AWS Per maggiori dettagli sulla configurazione delle richieste, consulta l'articolo sulla configurazione di asserzioni SAML per la risposta di autenticazione.

Quando il tuo IdP SAML include più di un certificato di firma nei metadati SAML, al momento dell'accesso il tuo pool di identità determina che l'asserzione SAML è valida se corrisponde a qualsiasi certificato nei metadati SAML.

Personalizzazione del ruolo utente con SAML

L'utilizzo di SAML con Amazon Cognito Identity ti consente la personalizzazione del ruolo per l'utente finale. Amazon Cognito supporta solo il flusso avanzato con l' SAML-basedIdP. Non è

necessario specificare un ruolo autenticato o non autenticato affinché il pool di identità utilizzi un IdP. SAML-based L'attributo `https://aws.amazon.com/SAML/Attributes/Role` di registrazione specifica una o più coppie composte da un ARN di provider e da un ruolo delimitato da virgole. Questi sono i ruoli che l'utente può assumere. Puoi configurare il provider di identità SAML per popolare gli attributi di ruolo in base alle informazioni di attributo utente disponibili dal provider di identità. Se nell'asserzione SAML vengono ricevuti più ruoli, il parametro `customRoleArn` opzionale deve essere popolato quando chiami `getCredentialsForIdentity`. L'utente assume questo `customRoleArn` se il ruolo corrisponde a uno dell'attestazione nell'asserzione SAML.

Autenticazione di utenti con un provider di identità SAML

Per eseguire la federazione con l' SAML-based IdP, determina l'URL in cui l'utente avvia l'accesso. AWS la federazione utilizza il login. IdP-initiated In AD FS 2.0, l'URL prende la forma di `https://<fqdn>/adfs/ls/IdpInitiatedSignOn.aspx?loginToRp=urn:amazon:webservices`.

Per aggiungere il supporto per il provider di identità SAML in Amazon Cognito, è necessario prima autenticare gli utenti con il provider di identità SAML dalla tua applicazione iOS o Android. Il codice utilizzato per integrare e autenticare con il provider di identità SAML è specifico dei provider SAML. Dopo l'autenticazione dell'utente, puoi fornire l'asserzione SAML risultante ad Amazon Cognito Identity usando le API di Amazon Cognito.

Non puoi ripetere o riprodurre un'asserzione SAML nella mappa `Logins` della tua richiesta API del pool di identità. Un'asserzione SAML riprodotta dispone di un ID asserzione che duplica l'ID di una richiesta API precedente. Le operazioni API che possono accettare un'asserzione SAML nella `Logins` mappa includono GetId, GetCredentialsForIdentityGetOpenIdToken, e. GetOpenIDTokenForDeveloperIdentity Puoi riprodurre un ID asserzione SAML una volta per richiesta API in un flusso di autenticazione del pool di identità. Ad esempio, puoi fornire la stessa asserzione SAML in una richiesta `GetId` e in una richiesta `GetCredentialsForIdentity` successiva, ma non in una seconda richiesta `GetId`.

Developer-authenticated identità

Amazon Cognito supporta identità autenticate dagli sviluppatori oltre alla federazione delle identità web tramite Configurazione di Facebook come pool di identità IdP, Configurazione di Google come IdP del pool di identità, Configurazione di Login with Amazon come IdP di pool di identità e Configurazione di Accedi con Apple come IdP del pool di identità. Con le identità autenticate dagli sviluppatori, puoi registrare e autenticare gli utenti tramite il tuo processo di autenticazione esistente, continuando a utilizzare Amazon Cognito per sincronizzare i dati degli utenti e accedere alle

risorse. AWS L'utilizzo di identità autenticate dagli sviluppatori implica l'interazione tra il dispositivo dell'utente finale, il backend per l'autenticazione e Amazon Cognito. Per maggiori dettagli, consulta Understanding Amazon Cognito Authentication Part 2: Developer Authenticated Identities nel blog. AWS

Informazioni sul flusso di autenticazione

Il funzionamento dell'GetOpenIdTokenForDeveloperIdentityAPI può avviare l'autenticazione degli sviluppatori sia per l'autenticazione avanzata che per quella di base. Questa API autentica una richiesta con credenziali amministrative. La `Logins` mappa è il nome di un provider di sviluppatori di pool di identità, ad esempio `login.mydevprovider` abbinato a un identificatore personalizzato.

Esempio:

```
"Logins": {
        "login.mydevprovider": "my developer identifier"
    }
```

Autenticazione avanzata

Chiama l'operazione GetCredentialsForIdentityAPI con una `Logins` mappa in cui la chiave è l'emittente del token restituito da `GetOpenIdTokenForDeveloperIdentity` e il valore è il token stesso.

L'emittente (e quindi la chiave di accesso richiesta) dipende dalla regione del pool di identità. Consulta Emittente del token per regione l'elenco completo delle regioni e dei rispettivi emittenti di token.

Esempio di un pool di identità in una regione che utilizza l'emittente globale:

```
"Logins": {
        "cognito-identity.amazonaws.com": "eyJra12345EXAMPLE"
    }
```

Esempio di un pool di identità in una regione più recente (come eu-south-2):

```
"Logins": {
        "cognito-identity.eu-south-2.amazonaws.com": "eyJra12345EXAMPLE"
    }
```

> **Important**
>
> La chiave della mappa di accesso deve corrispondere esattamente all'attestazione `iss` (emittente) contenuta nel token. Se non corrispondono, `GetCredentialsForIdentity` restituisce: `NotAuthorizedException: Invalid login token. Issuer doesn't match providerName`

> **Tip**
>
> Per determinare la chiave di accesso corretta a livello di codice, decodifica il token (un JWT) e leggi il valore del claim. `iss` Usa quel valore come chiave della mappa degli accessi.

`GetCredentialsForIdentity`with developer authenticated identities restituisce credenziali temporanee per il ruolo autenticato predefinito del pool di identità.

Autenticazione di base

Chiama l'operazione AssumeRoleWithWebIdentityAPI e richiedi l'attivazione `RoleArn` di qualsiasi ruolo IAM che abbia definito una relazione di trust appropriata. Imposta il valore di `WebIdentityToken` per il token ottenuto da`GetOpenIdTokenForDeveloperIdentity`.

Per informazioni sull'authflow delle identità autenticate dagli sviluppatori e su come si differenziano dalle identità dei provider esterni, consulta. Flusso di autenticazione dei pool di identità

Definisci un nome del provider per sviluppatori e associalo con un pool di identità

Per usare le identità autenticate dagli sviluppatori, sarà necessario un pool di identità associato al provider degli sviluppatori. A tale scopo, procedi nel modo seguente:

Per aggiungere un provider degli sviluppatori personalizzato

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Scegli Provider degli sviluppatori personalizzato.
5. Inserisci un Nome del provider di sviluppatori. Non è possibile modificare o eliminare il provider degli sviluppatori dopo averlo aggiunto.

6. Seleziona Salva modifiche.

Nota: una volta impostato, il nome del provider non può essere modificato.

Implementa un provider di identità

Android

Per utilizzare le identità autenticate dagli sviluppatori, implementa la tua classe di provider di identità che estende `AWSAbstractCognitoIdentityProvider`. La tua classe di provider di identità dovrebbe restituire un oggetto di risposta contenente il token come attributo.

Di seguito è riportato un esempio di base di provider di identità.

```
public class DeveloperAuthenticationProvider extends
 AWSAbstractCognitoDeveloperIdentityProvider {

  private static final String developerProvider = "<Developer_provider_name>";

  public DeveloperAuthenticationProvider(String accountId, String identityPoolId,
 Regions region) {
    super(accountId, identityPoolId, region);
    // Initialize any other objects needed here.
  }

  // Return the developer provider name which you choose while setting up the
  // identity pool in the &COG; Console

  @Override
  public String getProviderName() {
    return developerProvider;
  }

  // Use the refresh method to communicate with your backend to get an
  // identityId and token.

  @Override
  public String refresh() {

    // Override the existing token
    setToken(null);
```

```
    // Get the identityId and token by making a call to your backend
    // (Call to your backend)

    // Call the update method with updated identityId and token to make sure
    // these are ready to be used from Credentials Provider.

    update(identityId, token);
    return token;

  }

  // If the app has a valid identityId return it, otherwise get a valid
  // identityId from your backend.

  @Override
  public String getIdentityId() {

    // Load the identityId from the cache
    identityId = cachedIdentityId;

    if (identityId == null) {
       // Call to your backend
    } else {
       return identityId;
    }

  }
}
```

Per utilizzare questo provider di identità, è necessario passarlo in `CognitoCachingCredentialsProvider`. Ecco un esempio:

```
DeveloperAuthenticationProvider developerProvider = new
 DeveloperAuthenticationProvider( null, "IDENTITYPOOLID", Regions.USEAST1);
CognitoCachingCredentialsProvider credentialsProvider = new
 CognitoCachingCredentialsProvider( context, developerProvider, Regions.USEAST1);
```

> **Note**
>
> Le versioni precedenti dell' AWS SDK for Android (2.x e precedenti) venivano `cognito-identity.amazonaws.com` utilizzate come nome del provider interno per le identità autenticate dagli sviluppatori. Se il tuo pool di identità si trova in una regione più recente

che utilizza un emittente regionalizzato, esegui l'upgrade a una versione SDK corrente. Per ulteriori informazioni, consulta Formato regionale dell'emittente del token.

iOS - Objective-C

Per utilizzare le identità autenticate dagli sviluppatori, implementa la tua classe di provider di identità che estende AWSCognitoCredentialsProviderHelper. La tua classe di provider di identità dovrebbe restituire un oggetto di risposta contenente il token come attributo.

```
@implementation DeveloperAuthenticatedIdentityProvider
/*
 * Use the token method to communicate with your backend to get an
 * identityId and token.
 */

- (AWSTask <NSString*> *) token {
    //Write code to call your backend:
    //Pass username/password to backend or some sort of token to authenticate user
    //If successful, from backend call getOpenIdTokenForDeveloperIdentity with logins
 map
    //containing "your.provider.name":"enduser.username"
    //Return the identity id and token to client
    //You can use AWSTaskCompletionSource to do this asynchronously

    // Set the identity id and return the token
    self.identityId = response.identityId;
    return [AWSTask taskWithResult:response.token];
}

@end
```

Per utilizzare questo provider di identità, è necessario passarlo in `AWSCognitoCredentialsProvider`, come illustrato nell'esempio seguente:

```
DeveloperAuthenticatedIdentityProvider * devAuth =
 [[DeveloperAuthenticatedIdentityProvider alloc]
 initWithRegionType:AWSRegionYOUR_IDENTITY_POOL_REGION
                                      identityPoolId:@"YOUR_IDENTITY_POOL_ID"
                                     useEnhancedFlow:YES
                                identityProviderManager:nil];
```

```
AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider
 alloc]

 initWithRegionType:AWSRegionYOUR_IDENTITY_POOL_REGION
                                                identityProvider:devAuth];
```

Se intendi supportare sia le identità non autenticate sia le identità autenticate dagli sviluppatori, sostituisci il metodo `logins` nell'implementazione di `AWSCognitoCredentialsProviderHelper`.

```
- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    }else{
        return [super logins];
    }
}
```

Se intendi supportare le identità autenticate dagli sviluppatori e i provider social, devi indicare chi è il provider attuale nell'implementazione `logins` di `AWSCognitoCredentialsProviderHelper`.

```
- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    }else if (/*logic to determine if user is Facebook*/){
        return [AWSTask taskWithResult: @{ AWSIdentityProviderFacebook :
 [FBSDKAccessToken currentAccessToken] }];
    }else {
        return [super logins];
    }
}
```

> **Note**
>
> Le versioni precedenti dell' AWS SDK per iOS (2.x e precedenti) venivano `cognito-identity.amazonaws.com` utilizzate come nome del provider interno per le identità autenticate dagli sviluppatori. Se il tuo pool di identità si trova in una regione più recente che utilizza un emittente regionalizzato, esegui l'upgrade a una versione SDK corrente. Per ulteriori informazioni, consulta Formato regionale dell'emittente del token.

iOS - Swift

Per utilizzare le identità autenticate dagli sviluppatori, implementa la tua classe di provider di identità che estende AWSCognitoCredentialsProviderHelper. La tua classe di provider di identità dovrebbe restituire un oggetto di risposta contenente il token come attributo.

```
import AWSCore
/*
 * Use the token method to communicate with your backend to get an
 * identityId and token.
 */
class DeveloperAuthenticatedIdentityProvider : AWSCognitoCredentialsProviderHelper {
    override func token() -> AWSTask<NSString> {
    //Write code to call your backend:
    //pass username/password to backend or some sort of token to authenticate user, if
 successful,
    //from backend call getOpenIdTokenForDeveloperIdentity with logins map containing
 "your.provider.name":"enduser.username"
    //return the identity id and token to client
    //You can use AWSTaskCompletionSource to do this asynchronously

    // Set the identity id and return the token
    self.identityId = resultFromAbove.identityId
    return AWSTask(result: resultFromAbove.token)
}
```

Per utilizzare questo provider di identità, è necessario passarlo in `AWSCognitoCredentialsProvider`, come illustrato nell'esempio seguente:

```
let devAuth =
 DeveloperAuthenticatedIdentityProvider(regionType: .YOUR_IDENTITY_POOL_REGION,
 identityPoolId: "YOUR_IDENTITY_POOL_ID", useEnhancedFlow: true,
 identityProviderManager:nil)
let credentialsProvider =
 AWSCognitoCredentialsProvider(regionType: .YOUR_IDENTITY_POOL_REGION,
 identityProvider:devAuth)
let configuration = AWSServiceConfiguration(region: .YOUR_IDENTITY_POOL_REGION,
 credentialsProvider:credentialsProvider)
AWSServiceManager.default().defaultServiceConfiguration = configuration
```

Se intendi supportare sia le identità non autenticate sia le identità autenticate dagli sviluppatori, sostituisci il metodo `logins` nell'implementazione di `AWSCognitoCredentialsProviderHelper`.

```
override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else {
        return super.logins()
    }
}
```

Se intendi supportare le identità autenticate dagli sviluppatori e i provider social, devi indicare chi è il provider attuale nell'implementazione `logins` di `AWSCognitoCredentialsProviderHelper`.

```
override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else if (/*logic to determine if user is Facebook*/){
        if let token = AccessToken.current?.authenticationToken {
            return AWSTask(result: [AWSIdentityProviderFacebook:token])
        }
        return AWSTask(error:NSError(domain: "Facebook Login", code: -1 , userInfo:
 ["Facebook" : "No current Facebook access token"]))
    }else {
        return super.logins()
    }
}
```

JavaScript

Dopo aver ottenuto un ID di identità e un token di sessione dal back-end, li trasmetterai al provider di. `fromCognitoIdentity` `@aws-sdk/credential-providers` La chiave di accesso deve corrispondere alla `iss` richiesta del token. Per i pool di identità nelle regioni che utilizzano l'emittente globale, utilizza. `cognito-identity.amazonaws.com` Per i pool di identità in altre regioni, usa`cognito-identity.`*`region`*`.amazonaws.com`. Per ulteriori informazioni, consulta Formato regionale dell'emittente del token.

```
import { fromCognitoIdentity } from "@aws-sdk/credential-providers";

const token = 'TOKEN_RETURNED_FROM_YOUR_PROVIDER';
const region = 'IDENTITY_POOL_REGION';

// Extract the issuer from the token
const payload = token.split('.')[1];
```

```
const claims = JSON.parse(Buffer.from(payload, 'base64url').toString());
const providerName = claims.iss;

// Your backend already established the identity, so use fromCognitoIdentity
// (which calls GetCredentialsForIdentity directly for the identity ID you pass)
// rather than fromCognitoIdentityPool (which would call GetId and resolve a new
 identity).
const credentials = fromCognitoIdentity({
   identityId: 'IDENTITY_ID_RETURNED_FROM_YOUR_PROVIDER',
   logins: {
      [providerName]: token
   },
   clientConfig: { region }
});

// Attach the provider to each service client that needs credentials.
const client = new SomeServiceClient({ region, credentials });
```

Aggiornamento della mappa degli accessi (solo Android e iOS)

Android

Una volta completata correttamente l'autenticazione dell'utente con il sistema di autenticazione, aggiorna la mappa degli accessi con il nome del provider degli sviluppatori e un identificatore utente per gli sviluppatori. Questa è una stringa alfanumerica che identifica in maniera univoca un utente nel sistema di autenticazione. Assicurati di chiamare il metodo `refresh` dopo aver aggiornato la mappa degli accessi, poiché `identityId` potrebbe essere cambiato:

```
HashMap<String, String> loginsMap = new HashMap<String, String>();
loginsMap.put(developerAuthenticationProvider.getProviderName(),
 developerUserIdentifier);

credentialsProvider.setLogins(loginsMap);
credentialsProvider.refresh();
```

iOS - objective-C

L'SDK per iOS chiama solo il metodo `logins` per ottenere le mappe degli accessi più recenti se non sono disponibili le credenziali o se sono scadute. Se vuoi forzare l'SDK a ottenere nuove credenziali (ad esempio, l'utente è passato dall'essere non autenticato a essere autenticato e desideri credenziali rispetto all'utente autenticato), chiama `clearCredentials` su `credentialsProvider`.

```
[credentialsProvider clearCredentials];
```

iOS - Swift

L'SDK per iOS chiama solo il metodo `logins` per ottenere le mappe degli accessi più recenti se non sono disponibili le credenziali o se sono scadute. Se vuoi forzare l'SDK a ottenere nuove credenziali (ad esempio, se il tuo utente è passato dall'essere non autenticato a essere autenticato e vuoi avere delle credenziali dell'utente autenticato), chiama `clearCredentials` sul tuo `credentialsProvider`.

```
credentialsProvider.clearCredentials()
```

Ottenimento di un token (lato server)

È possibile ottenere un token chiamando GetOpenIdTokenForDeveloperIdentity. Questa API deve essere richiamata dal back-end utilizzando le credenziali AWS dello sviluppatore. Non deve essere invocata dall'SDK client. L'API riceve l'ID pool di identità di Cognito, una mappa degli accessi contenente il nome del provider di identità come chiave e l'identificatore come valore, e, facoltativamente, un ID identità di Cognito (ad esempio, se stai rendendo autenticato un utente non autenticato). L'identificatore può essere il nome utente dell'utente, un indirizzo e-mail o un valore numerico. L'API risponde alla tua chiamata con un ID di Cognito univoco per il tuo utente e con un token Cognito OpenID Connect per l'utente finale.

Qualche consiglio da tenere a mente relativamente al token restituito da `GetOpenIdTokenForDeveloperIdentity`:

- Puoi specificare un periodo di scadenza personalizzato per il token in modo da memorizzarlo nella cache. Se non fornisci alcun periodo di scadenza personalizzato, il token è valido per 15 minuti.
- È possibile impostare un periodo massimo di durata del token di 24 ore.
- Sii consapevole delle implicazioni di sicurezza relative all'aumento della durata del token. Se un utente malintenzionato ottiene questo token, può scambiarlo con AWS credenziali per l'utente finale per la durata del token.

I seguenti frammenti di codice Java mostrano come inizializzare un client Amazon Cognito e recuperare un token per un'identità autenticata dagli sviluppatori.

```
// authenticate your end user as appropriate
```

```
// ....

// if authenticated, initialize a cognito client with your AWS developer credentials
AmazonCognitoIdentity identityClient = new AmazonCognitoIdentityClient(
  new BasicAWSCredentials("access_key_id", "secret_access_key")
);

// create a new request to retrieve the token for your end user
GetOpenIdTokenForDeveloperIdentityRequest request =
  new GetOpenIdTokenForDeveloperIdentityRequest();
request.setIdentityPoolId("YOUR_COGNITO_IDENTITY_POOL_ID");

request.setIdentityId("YOUR_COGNITO_IDENTITY_ID"); //optional, set this if your client
 has an
                                                    //identity ID that you want to link
 to this
                                                    //developer account

// set up your logins map with the username of your end user
HashMap<String,String> logins = new HashMap<>();
logins.put("YOUR_IDENTITY_PROVIDER_NAME","YOUR_END_USER_IDENTIFIER");
request.setLogins(logins);

// optionally set token duration (in seconds)
request.setTokenDuration(60 * 15l);
GetOpenIdTokenForDeveloperIdentityResult response =
  identityClient.getOpenIdTokenForDeveloperIdentity(request);

// obtain identity id and token to return to your client
String identityId = response.getIdentityId();
String token = response.getToken();

//code to return identity id and token to client
//...
```

Seguendo i passaggi precedenti, sarai in grado di integrare le identità autenticate dagli sviluppatori nell'app. Se hai problemi o domande, scrivi nei nostri forum.

Formato regionale dell'emittente del token

Il token OpenID Connect restituito da `GetOpenIdTokenForDeveloperIdentity` contiene un'attestazione `iss` (emittente). Il formato di questa dichiarazione dipende dalla AWS regione del pool di identità.

Quando si chiama `GetCredentialsForIdentity` (enhanced flow), la chiave della mappa di accesso deve corrispondere esattamente alla `iss` richiesta del token.

Emittente del token per regione

Il valore dell'emittente del token dipende dalla regione del pool di identità.

- Regioni che utilizzano l'emittente globale (us-east-1, us-east-2, us-west-1, us-west-2, ap-northeast-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-southeast-1 ap-southeast-2, ca-central-1, eu-central-1, eu-central-1, eu-nord-1, eu-west-1, eu-west-2, eu-west-3, sa-east-1): `cognito-identity.amazonaws.com`
- Le regioni della Cina (cn-north-1): `cognito-identity.cn-north-1.amazonaws.com.cn`
- Tutte le restanti regioni commerciali: AWS `cognito-identity.`*`region`*`.amazonaws.com`

Determinazione della chiave di accesso a livello di codice

Invece di codificare la chiave di accesso, puoi estrarla dal token:

Java

```
import java.util.Base64;
import org.json.JSONObject;

String[] parts = token.split("\\.");
String payload = new String(Base64.getUrlDecoder().decode(parts[1]));
String issuer = new JSONObject(payload).getString("iss");
// Use 'issuer' as the logins map key in GetCredentialsForIdentity
```

Python

```
import base64, json

payload = token.split('.')[1]
# Add padding if needed
payload += '=' * (4 - len(payload) % 4)
claims = json.loads(base64.urlsafe_b64decode(payload))
issuer = claims['iss']
# Use 'issuer' as the logins map key
```

JavaScript

```
const payload = token.split('.')[1];
const claims = JSON.parse(Buffer.from(payload, 'base64url').toString());
const issuer = claims.iss;
// Use 'issuer' as the logins map key
```

Considerazioni sulla migrazione

Se stai migrando un pool di identità in una regione che utilizza un emittente regionalizzato, tieni presente quanto segue:

1. I client che utilizzano l'hardcode `cognito-identity.amazonaws.com` come chiave di accesso falliranno nel pool di identità della nuova regione.
2. Il codice client deve essere aggiornato per utilizzare l'emittente regionalizzato (`cognito-identity.`*`new-region`*`.amazonaws.com`) o per leggere dinamicamente la `iss` dichiarazione dal token (scelta consigliata).
3. Le versioni precedenti dei AWS Mobile SDK codificano l'emittente globale. Ciò include l' AWS SDK per Android 2.x e versioni precedenti e AWS l'SDK per iOS 2.x e versioni precedenti. Se il tuo pool di identità utilizza un emittente regionale, esegui l'upgrade a una versione SDK corrente.

Connessione a un'identità social esistente

Tutti i collegamenti dei provider quando si utilizzano identità autenticate dagli sviluppatori devono essere eseguiti dal back-end. Per collegare un'identità personalizzata all'identità social di un utente (Login with Amazon, Accedi con Apple, Facebook o Google), aggiungi il token del provider di identità alla mappa degli accessi quando GetOpenIdTokenForDeveloperIdentitychiami. Per rendere possibile ciò, quando richiami il back-end dall'SDK del client per autenticare l'utente finale, trasmetti anche il token del provider social dell'utente finale.

Ad esempio, se stai cercando di collegare un'identità personalizzata a Facebook, oltre all'identificatore del provider di identità, sarà necessario aggiungere alla mappa degli accessi anche il token di Facebook quando chiami `GetOpenIdTokenForDeveloperIdentity`.

```
logins.put("YOUR_IDENTITY_PROVIDER_NAME","YOUR_END_USER_IDENTIFIER");
logins.put("graph.facebook.com","END_USERS_FACEBOOK_ACCESSTOKEN");
```

Supporto delle transizioni tra provider

Android

L'applicazione potrebbe richiedere il supporto di identità non autenticate o di identità autenticate tramite provider pubblici (Login with Amazon, Accedi con Apple, Facebook o Google) oltre alle identità autenticate dagli sviluppatori. La differenza principale tra le identità autenticate dagli sviluppatori e altre identità (identità non autenticate e identità autenticate tramite provider pubblici) risiede nella modalità in cui identityId e token vengono ottenuti. Per altre identità, l'applicazione mobile interagisce direttamente con Amazon Cognito anziché contattare il tuo sistema di autenticazione. Pertanto, l'applicazione mobile deve essere in grado di supportare due flussi distinti, a seconda della scelta dell'utente dell'app. Per questo motivo sarà necessario apportare alcune modifiche al provider di identità personalizzato.

Il metodo `refresh` controlla la mappa degli accessi. Se la mappa non è vuota e ha una chiave con il nome del provider di sviluppo, chiama il back-end. Altrimenti, chiama il IdentityId metodo get e restituisci null.

```
public String refresh() {

   setToken(null);

   // If the logins map is not empty make a call to your backend
   // to get the token and identityId
   if (getProviderName() != null &&
      !this.loginsMap.isEmpty() &&
      this.loginsMap.containsKey(getProviderName())) {

      /**
       * This is where you would call your backend
       **/

      // now set the returned identity id and token in the provider
      update(identityId, token);
      return token;

   } else {
      // Call getIdentityId method and return null
      this.getIdentityId();
      return null;
   }
```

```
}
```

Allo stesso modo, il `getIdentityId` metodo ha due flussi a seconda del contenuto della mappa degli accessi:

```
public String getIdentityId() {

   // Load the identityId from the cache
   identityId = cachedIdentityId;

   if (identityId == null) {

      // If the logins map is not empty make a call to your backend
      // to get the token and identityId

      if (getProviderName() != null && !this.loginsMap.isEmpty()
         && this.loginsMap.containsKey(getProviderName())) {

         /**
           * This is where you would call your backend
          **/

         // now set the returned identity id and token in the provider
         update(identityId, token);
         return token;

      } else {
         // Otherwise call &COG; using getIdentityId of super class
         return super.getIdentityId();
      }

   } else {
      return identityId;
   }

}
```

iOS - Objective-C

L'applicazione potrebbe richiedere il supporto di identità non autenticate o di identità autenticate tramite provider pubblici (Login with Amazon, Accedi con Apple, Facebook o Google) oltre alle identità autenticate dagli sviluppatori. Per fare ciò, sovrascrivi il

AWSCognitoCredentialsProviderHelper`logins`metodo per poter restituire la mappa di accesso corretta in base all'attuale provider di identità. Questo esempio illustra come muoversi tra un'identità non autenticata, un'identità autenticata da Facebook e dagli sviluppatori.

```
- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    }else if (/*logic to determine if user is Facebook*/){
        return [AWSTask taskWithResult: @{ AWSIdentityProviderFacebook :
 [FBSDKAccessToken currentAccessToken] }];
    }else {
        return [super logins];
    }
}
```

Quando effettui la transizione da non autenticato ad autenticato, è necessario chiamare `[credentialsProvider clearCredentials];` per forzare l'SDK a ricevere nuove credenziali autenticate. Quando si passa tra due provider autenticati e non si cerca di collegarli (ad esempio, se non vengono forniti token per più provider nel dizionario degli accessi), chiama `[credentialsProvider clearKeychain];`. Ciò cancella sia le credenziali che l'identità e costringe l'SDK a ottenerne di nuove.

iOS - Swift

L'applicazione potrebbe richiedere il supporto di identità non autenticate o di identità autenticate tramite provider pubblici (Login with Amazon, Accedi con Apple, Facebook o Google) oltre alle identità autenticate dagli sviluppatori. A tale scopo, sostituisci il AWSCognitoCredentialsProviderHelper`logins`metodo per poter restituire la mappa di accesso corretta in base all'attuale provider di identità. Questo esempio illustra come muoversi tra un'identità non autenticata, un'identità autenticata da Facebook e dagli sviluppatori.

```
override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else if (/*logic to determine if user is Facebook*/){
        if let token = AccessToken.current?.authenticationToken {
            return AWSTask(result: [AWSIdentityProviderFacebook:token])
        }
        return AWSTask(error:NSError(domain: "Facebook Login", code: -1 , userInfo:
 ["Facebook" : "No current Facebook access token"]))
    }else {
```

```
        return super.logins()
    }
}
```

Quando effettui la transizione da non autenticato ad autenticato, è necessario chiamare `credentialsProvider.clearCredentials()` per forzare l'SDK a ricevere nuove credenziali autenticate. Quando alterni tra due provider autenticati e non cerchi di collegarli (ad esempio, se non fornisci token per più provider nel tuo dizionario degli accessi), è necessario chiamare `credentialsProvider.clearKeychain()`. Ciò cancella sia le credenziali che l'identità e costringe l'SDK a ottenerne di nuove.

Unità

L'applicazione potrebbe richiedere il supporto di identità non autenticate o di identità autenticate tramite provider pubblici (Login with Amazon, Accedi con Apple, Facebook o Google) oltre alle identità autenticate dagli sviluppatori. La differenza principale tra le identità autenticate dagli sviluppatori e altre identità (identità non autenticate e identità autenticate tramite provider pubblici) risiede nella modalità in cui identityId e token vengono ottenuti. Per altre identità, l'applicazione mobile interagisce direttamente con Amazon Cognito anziché contattare il tuo sistema di autenticazione. L'applicazione per dispositivi mobili deve essere in grado di supportare due flussi distinti, a seconda della scelta dell'utente dell'app. Per questo motivo sarà necessario apportare alcune modifiche al provider di identità personalizzato.

Il metodo consigliato per eseguire questa operazione in Unity consiste nell'estendere il proprio provider di AbstractCognitoIdentityProvider identità AmazonCognitoEnhancedIdentityProvide anziché utilizzare il RefreshAsync metodo principale anziché il proprio nel caso in cui l'utente non sia autenticato con il proprio backend. Se l'utente viene autenticato, puoi utilizzare lo stesso flusso spiegato precedentemente.

Passaggio da utenti non autenticati a utenti autenticati

I pool di identità di Amazon Cognito supportano sia utenti autenticati che non autenticati. Gli utenti non autenticati ricevono l'accesso alle tue AWS risorse anche se non hanno effettuato l'accesso con nessuno dei tuoi provider di identità (). IdPs Questo livello di accesso è utile per mostrare i contenuti agli utenti prima che effettuino l'accesso. Ogni utente non autenticato ha un'identità unica nel pool di identità, anche se non è stato effettuato l'accesso e l'autenticazione individualmente.

Questa sezione descrive il caso in cui l'utente sceglie di passare dall'accesso con un'identità non autenticata all'utilizzo di un'identità autenticata.

Android

Gli utenti possono accedere all'applicazione come ospiti non autenticati. Alla fine potrebbero decidere di accedere utilizzando uno dei supporti. IdPs Amazon Cognito verifica che una vecchia identità usi lo stesso identificatore univoco della nuova identità e i dati del profilo vengono uniti automaticamente.

La tua applicazione viene informata di un'unione dei profili tramite l'interfaccia `IdentityChangedListener`. Implementa il metodo `identityChanged` nell'interfaccia per ricevere questi messaggi:

```
@override
public void identityChanged(String oldIdentityId, String newIdentityId) {
    // handle the change
}
```

iOS - Objective-C

Gli utenti possono accedere alla tua applicazione come ospiti non autenticati. Alla fine potrebbero decidere di accedere utilizzando uno dei supporti. IdPs Amazon Cognito verifica che una vecchia identità usi lo stesso identificatore univoco della nuova identità e i dati dei due profili vengono uniti automaticamente.

`NSNotificationCenter` informa la tua applicazione di un'unione dei profili:

```
[[NSNotificationCenter defaultCenter] addObserver:self
                                      selector:@selector(identityIdDidChange:)
                                      name:AWSCognitoIdentityIdChangedNotification
                                      object:nil];

-(void)identityDidChange:(NSNotification*)notification {
    NSDictionary *userInfo = notification.userInfo;
    NSLog(@"identity changed from %@ to %@",
        [userInfo objectForKey:AWSCognitoNotificationPreviousId],
        [userInfo objectForKey:AWSCognitoNotificationNewId]);
}
```

iOS - Swift

Gli utenti possono accedere alla tua applicazione come ospiti non autenticati. Alla fine potrebbero decidere di accedere utilizzando uno dei supporti. IdPs Amazon Cognito verifica che una vecchia

identità usi lo stesso identificatore univoco della nuova identità e i dati dei due profili vengono uniti automaticamente.

`NSNotificationCenter` informa la tua applicazione di un'unione dei profili:

```
[NSNotificationCenter.defaultCenter().addObserver(observer: self
   selector:"identityDidChange"
   name:AWSCognitoIdentityIdChangedNotification
   object:nil)

func identityDidChange(notification: NSNotification!) {
  if let userInfo = notification.userInfo as? [String: AnyObject] {
    print("identity changed from: \(userInfo[AWSCognitoNotificationPreviousId])
    to: \(userInfo[AWSCognitoNotificationNewId])")
  }
}
```

JavaScript

Utente inizialmente non autenticato

Gli utenti in genere iniziano con il ruolo non autenticato. Per questo ruolo, crei un fornitore di `fromCognitoIdentityPool` credenziali senza una `logins` proprietà e lo colleghi ai tuoi clienti di servizio. In questo caso, la configurazione predefinita potrebbe essere simile alla seguente:

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";

// Create a credentials provider without a logins map for the unauthenticated role.
const region = "us-east-1";
const credentials = fromCognitoIdentityPool({
    identityPoolId: "us-east-1:1699ebc0-7900-4099-b910-2df94f52a030",
    clientConfig: { region }
});

// Attach the provider to each service client that needs credentials.
const client = new SomeServiceClient({ region, credentials });
```

Cambio a utente autenticato

Quando un utente non autenticato accede a un IdP e disponi di un token, puoi passare da non autenticato a autenticato chiamando una funzione personalizzata che crea un nuovo `fromCognitoIdentityPool` provider con la mappa che contiene il token: `logins`

```
// Called when an identity provider has a token for a logged in user
function userLoggedIn(providerName, token) {
    // Re-create the provider with a logins map to switch to the authenticated role.
    const credentials = fromCognitoIdentityPool({
        identityPoolId: "us-east-1:1699ebc0-7900-4099-b910-2df94f52a030",
        logins: {
            [providerName]: token
        },
        clientConfig: { region }
    });

    // Re-create any service clients so they pick up the new credentials.
    // SomeServiceClient is a placeholder: import the client for the service you're
 calling
    // (for example, S3Client from @aws-sdk/client-s3).
    const client = new SomeServiceClient({ region, credentials });
}
```

Poiché il `fromCognitoIdentityPool` provider è collegato direttamente a ciascun client di servizio, passare dal ruolo non autenticato a quello autenticato significa creare un nuovo provider con la `logins` mappa e passarla a tutti i client di servizio creati successivamente. A differenza dell' AWS SDK per la JavaScript v2, l'SDK v3 non utilizza un oggetto di configurazione globale, quindi non ci sono credenziali condivise da reimpostare.

Per ulteriori informazioni sul `fromCognitoIdentityPool` provider, consulta AWS SDK per i provider di credenziali v3. JavaScript

Guida introduttiva ai pool di identità di Amazon Cognito

I bacini d'utenza di Amazon Cognito consentono di creare identità univoche e assegnare le autorizzazioni agli utenti. Il tuo pool di identità può importare identità dai seguenti tipi di servizi di autenticazione:

- Utenti in un bacino d'utenza di Amazon Cognito

- Utenti che eseguono l'autenticazione con un provider di identità esterno come Facebook, Google, Apple o un provider di identità OIDC o SAML.
- Utenti autenticati tramite il processo di autenticazione esistente

Dopo che gli utenti si sono autenticati con il proprio provider e hanno presentato l'autorizzazione a un pool di identità, ottengono credenziali temporanee AWS . Le credenziali degli utenti dispongono di autorizzazioni definite dall'utente per l'accesso ad altri utenti. Servizi AWS

Argomenti

- Creazione di un pool di identità in Amazon Cognito
- Configurazione di un SDK
- Integrazione dei provider di identità
- Ottenere le credenziali
- Esempio di applicazione per pool di identità

Creazione di un pool di identità in Amazon Cognito

Puoi creare un bacino d'utenza tramite la console di Amazon Cognito oppure puoi utilizzare l' AWS Command Line Interface (CLI) o le API di Amazon Cognito. La procedura seguente è una guida generale per creare un nuovo pool di identità nella console. Puoi anche passare direttamente alla console e seguire l'esperienza guidata e i contenuti di aiuto in linea.

Per creare un nuovo pool di identità nella console

1. Accedi alla console di Amazon Cognito e seleziona Pool di identità. Per assegnare le autorizzazioni al tuo responsabile IAM in modo che possa creare e gestire risorse Amazon Cognito, consulta. AWS politiche gestite per Amazon Cognito La `AmazonCognitoPowerUser` policy è sufficiente per la creazione di pool di identità.
2. Scegli Crea pool di identità.
3. In Configurazione dell'attendibilità del pool di identità, scegli di configurare il pool di identità per Accesso autenticato, Accesso guest o entrambi.
    - Se hai scelto Accesso autenticato, seleziona uno o più Tipi di identità che desideri impostare come origine delle identità autenticate nel pool di identità. Se configuri un Provider degli sviluppatori personalizzato, non puoi modificarlo né eliminarlo dopo aver creato il pool di identità.

4. In Configura le autorizzazioni, scegli un ruolo IAM predefinito per gli utenti autenticati o guest nel pool di identità.

   a. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza documento di policy per esaminare le autorizzazioni che Amazon Cognito assegna al tuo nuovo ruolo IAM.

   b. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo Account AWS che desideri utilizzare. Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.

5. In Connect identity providers, inserisci i dettagli dei provider di identità (IdPs) che hai scelto in Configure identity pool trust. È possibile che ti venga chiesto di fornire informazioni sul client dell'app OAuth, scegliere un pool di utenti Amazon Cognito, scegliere un IdP IAM o inserire un identificatore personalizzato per un provider degli sviluppatori.

   a. Scegli le impostazioni del ruolo per ogni IdP. Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure puoi selezionare l'opzione Scegli ruolo con regole. Con un IdP del pool di utenti Amazon Cognito, puoi anche scegliere un ruolo con attestazione preferred_role nei token. Per ulteriori informazioni sulla richiesta `cognito:preferred_role`, consultare Assegnazione dei valori di priorità ai gruppi.

      i. Se hai scelto Scegli ruolo con regole, inserisci la dichiarazione di origine ricavata dall'autenticazione dell'utente, l'operatore con cui vuoi confrontare l'attestazione, il valore che determina una corrispondenza con questa scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

      ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

   b. Configura Attributi per il controllo degli accessi per ciascun IdP. L'opzione Attributi per il controllo degli accessi associa le richieste dell'utente ai tag principali applicati da Amazon Cognito alla relativa sessione temporanea. Puoi creare policy IAM per filtrare l'accesso utente in base ai tag applicati alla relativa sessione.

i. Per non applicare alcun tag principale, scegli Inattivo.

ii. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

iii. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

6. In Configura proprietà, inserisci un Nome in Nome del pool di identità.
7. In Autenticazione di base (classica), scegli Attiva flusso di base, se desiderato. Con il flusso di base attivo, puoi ignorare le selezioni di ruolo che hai effettuato per te e chiamare direttamente IdPs . AssumeRoleWithWebIdentity Per ulteriori informazioni, consulta Flusso di autenticazione dei pool di identità.
8. In Tag, scegli Aggiungi tag se desideri applicare tag al pool di identità.
9. In Esamina e crea, conferma le selezioni effettuate per il nuovo pool di identità. Seleziona Modifica per tornare alla procedura guidata e modificare le eventuali impostazioni. Al termine, seleziona Crea un pool di identità.

Configurazione di un SDK

Per utilizzare i pool di identità di Amazon Cognito, configura AWS Amplify AWS SDK per Java, o il. SDK per .NET Per ulteriori informazioni, consulta i seguenti argomenti.

- Configurazione dell'SDK per JavaScript nella Guida per gli sviluppatori AWS SDK per JavaScript
- Documentazione di Amplify in Amplify Dev Center
- Provider di credenziali Amazon Cognito nella Guida per gli sviluppatori di SDK per .NET

Integrazione dei provider di identità

In pool di identità di Amazon Cognito (identità federate) supportano l'autenticazione degli utenti tramite i pool di utenti Amazon Cognito, i provider di identità federate (inclusi i provider di identità Amazon, Facebook, Google e SAML) e le identità non autenticate. Supporta anche questa funzionalitàDeveloper-authenticated identità, che consente di registrare e autenticare gli utenti tramite il proprio processo di autenticazione back-end.

Per ulteriori informazioni sull'utilizzo di un bacino d'utenza di Amazon Cognito per la creazione della tua directory utente, consulta Autenticazione degli utenti con pool di utenti di Amazon Cognito e Accesso Servizi AWS utilizzo di un pool di identità dopo l'accesso.

Per ulteriori informazioni su come utilizzare provider di identità esterni, consulta Pool di identità, provider di identità di terze parti.

Per ulteriori informazioni su come integrare il tuo processo di autenticazione di back-end, consulta Developer-authenticated identità.

Ottenere le credenziali

I pool di identità di Amazon Cognito forniscono AWS credenziali temporanee per gli utenti ospiti (non autenticati) e per gli utenti che si sono autenticati e hanno ricevuto un token. Con queste AWS credenziali, la tua app può accedere in modo sicuro a un backend interno AWS o esterno AWS tramite Amazon API Gateway. Consulta Ottenere le credenziali.

Esempio di applicazione per pool di identità

Il caso d'uso più comune dei pool di identità di Amazon Cognito consiste nel federare gli utenti di più sistemi di accesso e fornire credenziali temporanee ad accesso limitato direttamente AWS al client. Ciò elimina la necessità di creare un broker di credenziali per le autorizzazioni di accesso alle risorse. AWS Ad esempio, potresti dover consentire agli utenti di accedere con i propri account di social media e accedere alle risorse dell'app da Amazon S3 per la tua applicazione mobile. I pool di identità forniscono inoltre le credenziali agli utenti che accedono con i pool di utenti.

In questo tutorial, creerai un'applicazione web in cui potrai ottenere credenziali temporanee autenticate e guest nei flussi di autenticazione avanzati e di base con provider di identità supportati (IdPs) nei pool di identità. Se hai già esperienza nello sviluppo web, scarica l'app di esempio da. GitHub

Scarica l'applicazione di esempio da GitHub

Questa applicazione di esempio dimostra le seguenti funzionalità dei pool di identità di Amazon Cognito:

Flussi di autenticazione nei pool di identità

- Flusso di autenticazione migliorato con suddivisioni dettagliate delle richieste API
- Flusso di autenticazione di base con suddivisioni dettagliate delle richieste API

Implementazione dell'accesso guest (non autenticato)

- Fornisci un Servizio AWS accesso limitato senza richiedere l'accesso

Integrazione con i provider di identità supportati

- Social IdPs (Facebook, Amazon, Twitter, Apple e Google) per l'accesso dei consumatori
- Enterprise IdPs (tramite OpenID Connect o SAML) per utenti aziendali
- Pool di utenti Amazon Cognito

AWS gestione delle credenziali

- Scambio di token di provider di identità con credenziali temporanee AWS
- Utilizzo di credenziali temporanee per accedere ai servizi in modo sicuro AWS

Dopo aver configurato l'applicazione sul server web di sviluppo e avervi effettuato l'accesso in un browser, vengono visualizzate le seguenti opzioni.



Argomenti

- Prerequisiti
- Configurazione del provider di autenticazione
- Implementa l'applicazione demo
- Esplora i metodi di autenticazione nel tuo pool di identità
- Fasi successive

Prerequisiti

Prima di iniziare, è necessario configurare le seguenti risorse.

- Un AWS account con accesso ad Amazon Cognito. Se non disponi di un AWS account, segui le istruzioni riportate inNozioni di base su AWS.
- Python 3.11 o successivo installato sulla tua macchina di sviluppo.
- GitHub accesso.
- AWS credenziali configurate con autorizzazioni per effettuare richieste autenticate alle API di Amazon Cognito. Queste credenziali sono necessarie per l'autenticazione degli sviluppatori.

Per ulteriori informazioni sull'implementazione delle AWS credenziali e sulla federazione dei pool di identità nel tuo SDK specifico, consulta. the section called “Ottenere le credenziali”

Configurazione del provider di autenticazione

Per ottenere i migliori risultati con questa applicazione, configura e integra uno o più provider di identità di terze parti (IdPs) o pool di utenti Amazon Cognito con il tuo pool di identità Amazon Cognito. Dopo aver completato i prerequisiti e prima di eseguire questa applicazione demo, scegli quali provider di identità configurare. La console Amazon Cognito ti guida attraverso il processo di configurazione dei pool di identità e dei provider.

Pool di utenti Amazon Cognito

- the section called “Autenticazione”
- the section called “Client dell'app”

Fornitori di identità social

- Google: the section called “Google”
- Facebook: the section called “Facebook”
- Amazon: the section called “Login with Amazon”

Provider OpenID Connect (OIDC)

- the section called “Provider Open ID Connect”

fornitori SAML

- the section called “Provider di identità SAML”

> **Note**
>
> Per questa applicazione demo, non è necessario configurare tutti i provider di identità supportati. Puoi iniziare con uno che corrisponda al tuo caso d'uso. Ogni collegamento fornisce istruzioni di configurazione dettagliate.

Implementa l'applicazione demo

Clona il repository

1. Apri una finestra del terminale.
2. Clona il `aws-doc-sdk-examples` repository o recupera in altro modo questa cartella nel repository.

```
git clone https://github.com/awsdocs/aws-doc-sdk-examples.git
```

3. Passa alla directory del progetto .

```
cd python/example_code/cognito/scenarios/identity_pools_example_demo/web
```

Crea un pool di identità di

Per creare un pool di identità Amazon Cognito per la tua applicazione, segui le istruzioni riportate in. the section called “Configurazione dei pool di identità”

Per configurare un pool di identità per l'applicazione demo

1. Apri la console Amazon Cognito.
2. Dal menu di navigazione a sinistra, scegli Identity pool. Scegli un pool di identità esistente o creane uno nuovo.
3. In Accesso utente, abilita Accesso autenticato e Accesso ospite. Configura un ruolo IAM nuovo o esistente e assegnagli le autorizzazioni che desideri concedere a ciascun tipo di utente.
4. In Accesso utente, configura tutti i provider di identità che desideri configurare.
5. In Proprietà del pool di identità, abilita l'autenticazione di base (classica).
6. Tieni il browser aperto alla console del tuo pool di identità. Utilizzerai l'ID del pool di identità e altre informazioni di configurazione nella configurazione dell'applicazione.

Configura ed esegui l'applicazione

I seguenti passaggi guidano l'utente nella configurazione iniziale dell'applicazione demo.

Per configurare l'applicazione demo

1. Apri una riga di comando `python/example_code/cognito/scenarios/identity_pools_example_demo/web` nel tuo `aws-doc-sdk-examples` clone.
2. Crea un `.env` file copiando il file di ambiente di esempio.

```
cp .env.example .env
```

3. Apri il file `.env` in un editor di testo. Sostituite i valori di esempio nel file con i vostri valori di configurazione.
4. Installa le dipendenze del backend.

```
pip install -r requirements.txt
```

5. Avvia il server di backend:

```
cd backend
python oauth_server.py
```

6. Apri una nuova finestra di terminale, vai alla directory del progetto e avvia il server frontend:

```
cd frontend
python -m http.server 8001
```

7. Apri il browser all'applicazione all'indirizzo: 8001. http://localhost Il browser visualizza l'interfaccia dimostrativa dell'applicazione, pronta per testare l'autenticazione dei pool di identità.

Esplora i metodi di autenticazione nel tuo pool di identità

Questa sezione ti guida attraverso i flussi di autenticazione di base e avanzati utilizzando l'applicazione demo dei pool di identità Amazon Cognito. Con questa demo, scoprirai come funzionano i pool di identità con vari provider di identità per fornire AWS credenziali temporanee agli utenti delle tue applicazioni.

Nella sezione demo interattiva dell'applicazione di esempio, sceglierai innanzitutto tra due tipi di accesso supportati dai pool di identità.

[Accesso non autenticato (ospite)]

Fornisci AWS le credenziali agli utenti che non si sono ancora autenticati.

Accesso autenticato

Scambia i token del provider di identità in cambio di AWS credenziali con una gamma completa di autorizzazioni disponibili. Scegli un provider di identità tra quelli configurati nel tuo file. `.env`

Accesso non autenticato (ospite)

Questo passaggio dimostra come ottenere AWS credenziali temporanee per gli utenti non autenticati (ospiti) tramite la funzionalità di accesso ospite del pool di identità. Nell'applicazione demo, testerai sia i flussi avanzati che quelli di base per vedere come i pool di identità emettono le credenziali senza richiedere l'accesso dell'utente. L'accesso ospite utilizza la stessa sequenza API dell'accesso autenticato, ma senza fornire token di provider di identità (come i token OAuth di Google, Facebook o le asserzioni SAML dei provider aziendali).

Continua a leggere se stai cercando informazioni su come fornire un accesso limitato AWS agli utenti senza richiedere l'autenticazione. Dopo aver implementato l'accesso come ospite, imparerai come fornire in modo sicuro AWS le credenziali agli utenti anonimi e a comprendere le differenze tra i due flussi di autenticazione.

> **Important**
>
> L'accesso non autenticato può rilasciare credenziali a chiunque abbia accesso a Internet, quindi è meglio utilizzarlo per AWS risorse che richiedono una sicurezza minima, come API pubbliche e risorse grafiche. Prima di procedere con questo passaggio, controlla se hai configurato il tuo pool di identità con l'accesso degli ospiti abilitato e assicurati che siano in atto politiche IAM adeguate per limitare le autorizzazioni.

Guest access with enhanced flow

Il flusso avanzato è un approccio semplificato all'ottenimento di AWS credenziali per utenti non autenticati con due richieste API.

Per testare l'accesso degli ospiti con il flusso avanzato

1. Nell'applicazione demo, vai alla sezione Demo interattiva

2. Scegli la scheda Accesso ospite.
3. Scegli la scheda Flusso avanzato.
4. Scegli Test guest access.
5. L'app ottiene AWS credenziali temporanee dai tuoi pool di identità senza ulteriori richieste di autenticazione.
6. Una volta completata l'autenticazione, vedrai l'interfaccia web che mostra il pannello Risultati e avrai due opzioni per esplorarli:

   a. Pulsante Visualizza solo le credenziali: scegliete questo pulsante se desiderate visualizzare direttamente AWS le credenziali temporanee generate senza i dettagli del flusso API.

   ```
   {
     "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
     "Credentials": {
       "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
       "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
       "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
       "Expiration": "2025-08-07T00:58:21-07:00"
     }
   }
   ```

   b. Pulsante Visualizza il flusso API dettagliato: scegli questo pulsante se desideri visualizzare le richieste API dettagliate.

   - `GetId()`Richiesta API con il tuo. `identityPoolId` Non sono richiesti token di autenticazione per l'accesso degli ospiti

     ```
     {
       "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
     }
     ```

     Se valido, trova o crea e restituisce quello dell'`IdentityID`utente. Un esempio di risposta è simile al seguente:

     ```
     {
       "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
     }
     ```

   - `GetCredentialsForIdentity()`con il restituito`identityPoolId`.

```
POST GetCredentialsForIdentity
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Cognito convalida l'accesso degli ospiti, assume internamente il ruolo non autenticato e restituisce credenziali AWS temporanee AWS STS. (Nessuna autenticazione IAM su questa chiamata; la fiducia nei ruoli deve consentirla). `cognito-identity-amazonzaws.com`

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-07T00:58:21-07:00"
  }
}
```

Guest access with basic flow

Il flusso di base fornisce un controllo granulare sul processo di autenticazione con richieste API separate per il recupero dell'identità e la generazione di credenziali.

Per testare l'accesso degli ospiti con il flusso di base

1. Nell'applicazione demo, vai alla sezione Demo interattiva
2. Scegli la scheda Accesso ospite.
3. Scegli la scheda Basic flow.
4. Scegli Test guest access.
5. L'app ottiene AWS credenziali temporanee dai tuoi pool di identità senza ulteriori richieste di autenticazione.
6. Una volta completata l'autenticazione, vedrai l'interfaccia web che mostra il pannello Risultati e avrai due opzioni per esplorarli.

a. Pulsante Visualizza solo le credenziali: scegli questo pulsante se desideri visualizzare direttamente AWS le credenziali temporanee generate senza i dettagli del flusso API.

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. Pulsante Visualizza il flusso API dettagliato: scegli questo pulsante se desideri visualizzare le richieste API dettagliate.

- `GetId()`Richiesta API con il tuo ID del pool di identità. Non sono richiesti token di autenticazione per l'accesso degli ospiti.

  ```
  POST GetId
  {
    "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
  ```

  Se valido, trova o crea e restituisce quello dell'`IdentityID`utente. Un esempio di risposta è simile al seguente:

  ```
  {
     "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
  }
  ```

- `GetOpenIdToken()`con la stessa `Logins` mappa restituita `IdentityID`

  ```
  POST GetOpenIdToken
  {
    "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
  }
  ```

  Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Token": "eyJraWQiOiJFWAMPLE......"
}
```

Cosa succede in questa fase: Amazon Cognito emette un token di identità web OpenID Connect di breve durata da cognito-identity.amazonaws.com che lo rappresenta. `IdentityId` Il token include attestazioni OIDC che AWS STS effettuano valutazioni, tra cui aud (ID del pool di identità) e amr (autenticato o non autenticato). La policy di fiducia del tuo ruolo IAM deve richiedere tali attestazioni.

- `AssumeRoleWithWebIdentity()`- La tua app chiama AWS STS direttamente per scambiare il token OpenID di Amazon Cognito con credenziali temporanee AWS

```
POST sts:AssumeRoleWithWebIdentity
{
  "RoleArn": "arn:aws:iam::111122223333:role/
Cognito_IdentityPoolUnauth_Role",
  "WebIdentityToken": "eyJraWQiOiJFWAMPLE......"
}
```

Risposta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "FwoGZXIvYXdzEEXAMPLE......"
  }
}
```

Cosa succede in questa fase: Una volta convalidata: restituisce le credenziali temporanee AWS

Usa le credenziali temporanee

Queste credenziali temporanee funzionano come AWS credenziali standard ma con autorizzazioni limitate definite dal ruolo IAM non autenticato del pool di identità. Puoi usarle con qualsiasi SDK o. AWS AWS CLI Per ulteriori informazioni sulla configurazione degli AWS SDK con credenziali,

consulta Fornitori di credenziali standardizzati nella Guida di riferimento agli SDK e agli AWS strumenti.

Gli esempi riportati di seguito non sono un elenco completo, ma mostrano alcuni modi comuni in cui la funzionalità guest di un pool di identità può migliorare l'esperienza utente.

Contenuti pubblici di sola lettura

I seguenti esempi configurano i provider di credenziali per l'accesso limitato ad Amazon S3 come utente ospite.

Python

```
# Example: Using credentials with boto3
import boto3

# Configure client with temporary credentials
s3_client = boto3.client(
    's3',
    aws_access_key_id='AKIAIOSFODNN7EXAMPLE',
    aws_secret_access_key='wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
    aws_session_token='IQoJb3JpZ2luX2VjEEXAMPLE......'
)

# Make API requests within IAM role permissions
response = s3_client.list_objects_v2(Bucket='my-public-bucket')

# Access public content
for obj in response.get('Contents', []):
    print(f"File: {obj['Key']}, Size: {obj['Size']} bytes")
```

JavaScript

```
// Example: Accessing public content
import { S3Client, GetObjectCommand } from "@aws-sdk/client-s3";

const s3Client = new S3Client({
    region: "us-east-1",
    credentials: {
        accessKeyId: 'AKIAIOSFODNN7EXAMPLE',
        secretAccessKey: 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
        sessionToken: 'IQoJb3JpZ2luX2VjEEXAMPLE......'
    }
```

```
});

// Access public images or documents
const response = await s3Client.send(new GetObjectCommand({
    Bucket: 'my-public-content',
    Key: 'product-catalog.pdf'
}));
```

"" caratteristiche Try-before-login

Gli esempi seguenti utilizzano l'accesso in sola lettura ad Amazon DynamoDB come utente guest.

Python

```
# Example: Limited app functionality for trial users
import boto3

dynamodb = boto3.client(
    'dynamodb',
    aws_access_key_id='AKIAIOSFODNN7EXAMPLE',
    aws_secret_access_key='wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
    aws_session_token='IQoJb3JpZ2luX2VjEEXAMPLE......'
)

# Allow guest users to view sample data (limited to 5 items)
response = dynamodb.scan(TableName='SampleProducts', Limit=5)
```

JavaScript

```
// Example: Limited app functionality for trial users
import { DynamoDBClient, ScanCommand } from "@aws-sdk/client-dynamodb";

const dynamodbClient = new DynamoDBClient({
    region: "us-east-1",
    credentials: {
        accessKeyId: 'AKIAIOSFODNN7EXAMPLE',
        secretAccessKey: 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
        sessionToken: 'IQoJb3JpZ2luX2VjEEXAMPLE......'
    }
});

// Allow guest users to view sample data (limited to 5 items)
```

```
const response = await dynamodbClient.send(new ScanCommand({
    TableName: 'SampleProducts',
    Limit: 5
}));
```

Autenticazione tramite provider di identità social

Questo passaggio esplora il flusso generale per l'utilizzo dei provider di identità social con i pool di identità di Amazon Cognito. L'autenticazione social offre un'esperienza di accesso familiare, pur mantenendo la sicurezza attraverso la gestione delle identità federate. Puoi accedere da un provider di identità social (IdP) come Google, Facebook e Amazon, quindi scambiare il token IdP con credenziali temporanee. AWS L'integrazione tra Twitter e Apple è supportata anche dai pool di identità, ma non è supportata nell'applicazione di esempio.

Il pool di identità in sé non è una directory utente. Non memorizza password o campi del profilo. Si affida invece IdPs all'esterno per l'autenticazione dell'utente e si concentra sull'autorizzazione dell'utente già autenticato a chiamare direttamente i AWS servizi, vendendo credenziali per i ruoli IAM.

Social identity provider with enhanced flow

Questa sezione mostra come utilizzare un provider di identità social per accedere a un utente e, utilizzando il flusso avanzato, scambiare il token del provider in un pool di identità di Amazon Cognito con credenziali temporanee per richiedere risorse. AWS

Usa l'accesso tramite social con il flusso migliorato dell'applicazione di esempio

1. Nell'applicazione demo, vai alla sezione Demo interattiva
2. Scegli la scheda Accesso autenticato.
3. Scegli la scheda Flusso avanzato.
4. Scegli un social provider supportato che hai configurato, ad esempio Accedi con Google, Accedi con Facebook o Login with Amazon.
5. Accedi e acconsenti a condividere i dati dell'utente con l'applicazione.
6. Il provider reindirizza nuovamente all'URI di reindirizzamento dell'app
7. L'app invia il token del provider al tuo pool di identità e recupera le credenziali temporanee AWS
8. L'app visualizza il pannello Risultati nell'interfaccia web.

Una volta completata l'autenticazione, vedrai l'interfaccia web che mostra il pannello Risultati e avrai due opzioni per esplorarli:

a. Pulsante Visualizza solo le credenziali: scegliete questo pulsante se desiderate visualizzare direttamente AWS le credenziali temporanee generate senza i dettagli del flusso API.

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. Pulsante Visualizza il flusso API dettagliato: scegli questo pulsante se desideri visualizzare le richieste API dettagliate.

- L'app accede all'utente con un IdP social e ottiene il token del provider. I pool di identità accettano questi artefatti dei provider di social network:

Gestore dell'identità digitale	Chiave del provider Cognito	Scopo
Google	`accounts.google.com`	Token OAuth 2.0 di Google Sign-In
Facebook	`graph.facebook.com`	Accedi ai token da Facebook Login
Amazon	`www.amazon.com`	Token OAuth da Login with Amazon

Dopo aver eseguito correttamente l'autenticazione con il social provider, l'app riceve una risposta OAuth contenente il token di accesso e altri dettagli di autenticazione:

```
{
```

```
   "access_token": "ya29.A0AS3H6NEXAMPLE......",
   "expires_in": 3599,
   "scope": "openid https://www.examplesocial....",
   "token_type": "Bearer",
   "id_token": "eyJhbGciOiJSUzI1NiIsEXAMPLE......"
}
```

- `GetId()`Richiesta API con l'ID del tuo pool di identità e una `Logins` mappa contenente il token del tuo provider sociale

```
POST GetId
{
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Logins": {
    "accounts.google.com": "eyJhbGciOiJSUzI1NiIsEXAMPLE......"
  }
}
```

  Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetCredentialsForIdentity()`con la `Logins` mappa restituita `IdentityID` e la stessa

```
POST GetCredentialsForIdentity
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "accounts.google.com": "eyJhbGciOiJSUzI1NiIsEXAMPLE......"
  }
}
```

  Risposta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
```

```
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE.....",
    "Expiration": "2025-08-07T00:58:21-07:00"
  },
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Cos'è successo: Amazon Cognito ha convalidato il token rispetto al provider configurato, ha scelto un ruolo IAM in base alla configurazione del provider e ha chiamato per AWS STS conto tuo. Il tuo pool di identità ha quindi restituito credenziali temporanee.

Social identity provider with basic flow

Questa sezione mostra come utilizzare un provider di identità social per accedere a un utente e, utilizzando il flusso di base, scambiare il token del provider in un pool di identità di Amazon Cognito con credenziali temporanee per chiamare i servizi. AWS

Usa l'accesso tramite social con il flusso di base dell'applicazione di esempio

1. Nell'applicazione demo, vai alla sezione Demo interattiva
2. Scegli la scheda Accesso autenticato.
3. Scegli la scheda Basic flow.
4. Scegli un social provider supportato che hai configurato, ad esempio Accedi con Google, Accedi con Facebook o Login with Amazon.
5. Accedi e acconsenti a condividere i dati dell'utente con l'applicazione.
6. Il provider reindirizza nuovamente all'URI di reindirizzamento dell'app
7. L'app invia il token del provider al tuo pool di identità e recupera le credenziali temporanee AWS
8. L'app visualizza il pannello Risultati nell'interfaccia web.

   Una volta completata l'autenticazione, vedrai l'interfaccia web che mostra il pannello Risultati e avrai due opzioni per esplorarli:

   a. Pulsante Visualizza solo le credenziali: scegliete questo pulsante se desiderate visualizzare direttamente AWS le credenziali temporanee generate senza i dettagli del flusso API.

      ```
      {
        "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      ```

```
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. Pulsante Visualizza il flusso API dettagliato: scegli questo pulsante se desideri visualizzare le richieste API dettagliate.

- L'app accede all'utente con un IdP social e ottiene il token del provider. I pool di identità accettano questi artefatti dei provider di social network:

Gestore dell'identità digitale	Chiave del provider Cognito	Scopo
Google	`accounts.google.com`	Token OAuth 2.0 di Google Sign-In
Facebook	`graph.facebook.com`	Accedi ai token da Facebook Login
Amazon	`www.amazon.com`	Token OAuth da Login with Amazon

Dopo aver eseguito correttamente l'autenticazione con il social provider, l'app riceve una risposta OAuth contenente il token di accesso e altri dettagli di autenticazione:

```
{
   "access_token": "ya29.A0AS3H6NEXAMPLE......",
   "expires_in": 3599,
   "scope": "openid https://www.examplesocial....",
   "token_type": "Bearer",
   "id_token": "eyJhbGciOiJSUzI1NiIsEXAMPLE......"
}
```

- `GetId()`Richiesta API con l'ID del tuo pool di identità e una `Logins` mappa contenente il token del tuo provider sociale

```
POST GetId
{
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Logins": {
    "accounts.google.com": "token..."
  }
}
```

Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetOpenIdToken()`con la mappa di accesso restituita `IdentityID` e la stessa

```
POST GetOpenIdToken
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "accounts.google.com": "token..."
  }
}
```

Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Token": "eyJraWQiOiJFWAMPLE......"
}
```

- `AssumeRoleWithWebIdentity()`con il token OpenID

```
POST AssumeRoleWithWebIdentity
{
  "RoleArn": "arn:aws:iam::111122223333:role/Cognito_IdentityPoolAuth_Role",
  "WebIdentityToken": "eyJraWQiOiJFWAMPLE......"
}
```

Risposta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T14:36:17-07:00"
  }
}
```

Cos'è successo: Amazon Cognito ha convalidato il token rispetto al provider configurato e ha emesso un token OpenID. L'applicazione è stata chiamata AWS STS direttamente per assumere un ruolo IAM e ricevere credenziali temporanee.

Comprendi l'accesso ai social

- Gli utenti social ricevono AWS credenziali temporanee tramite i pool di identità di Amazon Cognito dopo l'autenticazione con il loro provider di social network.
- Ogni utente autenticato riceve un ID di identità univoco che persiste tra le sessioni.
- Queste credenziali sono collegate a un ruolo IAM progettato specificamente per l'accesso autenticato, che fornisce autorizzazioni più ampie rispetto all'accesso come ospite.
- I token dei social provider vengono scambiati con AWS credenziali, mantenendo l'identità e le autorizzazioni degli utenti.

Autenticazione dei pool di utenti di Amazon Cognito

Questo passaggio esplora l'autenticazione di Amazon Cognito con l'integrazione dell'accesso gestito da pool di utenti. Quando colleghi un pool di utenti come IdP a un pool di identità, i token del pool di utenti autorizzano il pool di identità a emettere credenziali temporanee.

User pool authentication with enhanced flow

Il flusso avanzato offre un approccio semplificato all'ottenimento di AWS credenziali tramite i pool di identità di Amazon Cognito con una singola richiesta API.

Usa l'autenticazione del pool di utenti di Amazon Cognito con il flusso avanzato del pool di identità

1. Nell'applicazione demo, vai alla sezione Demo interattiva

2. Scegli la scheda Accesso autenticato.
3. Scegli la scheda Flusso avanzato.
4. Scegli Accedi con i pool di utenti di Amazon Cognito
5. Completa l'accesso con nome utente e password nell'accesso gestito.
6. Il pool di utenti reindirizza nuovamente all'URI di reindirizzamento dell'applicazione con un codice di autorizzazione.
7. L'applicazione scambia il codice di autorizzazione con il pool di utenti per i token web JSON.
8. L'applicazione scambia il token ID con il pool di identità dell'utente in cambio di credenziali temporanee AWS
9. L'app visualizza il pannello Risultati nell'interfaccia web

   Una volta completata l'autenticazione, vedrai l'interfaccia web che mostra il pannello Risultati e avrai due opzioni per esplorarli:

   a. Pulsante Visualizza solo le credenziali: scegliete questo pulsante se desiderate visualizzare direttamente AWS le credenziali temporanee generate senza i dettagli del flusso API.

   ```
   {
     "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
     "Credentials": {
       "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
       "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
       "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
       "Expiration": "2025-08-12T13:36:17-07:00"
     }
   }
   ```

   b. Pulsante Visualizza il flusso API dettagliato: scegli questo pulsante se desideri visualizzare le richieste API dettagliate.

      - L'app accede all'utente con Amazon Cognito. Dopo l'autenticazione con il pool di utenti, l'app riceve una risposta OAuth 2.0 contenente il token ID (JWT). I pool di identità accettano i token ID JWT dai pool di utenti utilizzando questo formato di chiave del provider:

Gestore dell'identità digitale	Chiave del provider Cognito	Scopo
Bacino d'utenza di Amazon Cognito	`cognito-idp.{region}.amazonaws.com/{user-pool-id}`	Token ID JWT dai pool di utenti di Amazon Cognito

Dopo l'autenticazione con il pool di utenti, l'app riceve una risposta OAuth 2.0 contenente il token ID (JWT):

```
{
   "id_token": "eyJraWQiOiJFWAMPLE......",
   "token_type": "Bearer",
   "expires_in": 3600
}
```

- `GetId()`Richiesta API con la tua `identityPoolId` e una `Logins` mappa che include la chiave del provider del pool di utenti mappata su. `id_token` Amazon Cognito ha verificato che la firma, l'emittente, la scadenza e il pubblico (`aud`) del token ID del pool di utenti corrispondano a uno degli ID client dell'app che hai registrato per questo pool di utenti IdP nel pool di identità.

```
POST GetId
{
  "AccountId": "111122223333",
  "IdentityPoolId": "us-east-1:1ac4a76d-1fef-48aa-83af-4224799c0b5c",
  "Logins": {
    "cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE123":
 "eyJraWQiOiJFWAMPLE......"
  }
}
```

Se valido, trova o crea e restituisce quello dell'utente. `IdentityID` Un esempio di risposta è simile al seguente:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
```

```
}
```

- `GetCredentialsForIdentity()`con la `Logins` mappa restituita `identityPoolId` e a=la stessa con. `id_token` Amazon Cognito riconvalida la firma, l'emittente, la scadenza e il pubblico (`aud`) del token ID del pool di utenti corrispondenti a uno degli ID client dell'app che hai registrato per questo pool di utenti IdP nel pool di identità.

```
POST GetCredentialsForIdentity
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE123":
 "eyJraWQiOiJFWAMPLE......"
  }
}
```

Se valido, sceglie un ruolo IAM (roles-in-token, rules o default), chiama per tuo conto e restituisce credenziali temporanee. AWS STS AWS Un esempio di risposta è simile al seguente:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "ASIAW7TIP7EJEXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T14:36:17-07:00"
  }
}
```

User pool authentication with basic flow

Il flusso di base fornisce un controllo granulare sul processo di autenticazione con richieste API separate per il recupero dell'identità e la generazione di credenziali.

Usa l'autenticazione del pool di utenti di Amazon Cognito con il flusso di base del pool di identità

1. Nell'applicazione demo, vai alla sezione Demo interattiva
2. Scegli la scheda Accesso autenticato.
3. Scegli la scheda Basic flow.

4. Scegli Accedi con i pool di utenti di Amazon Cognito
5. Completa l'accesso con nome utente e password nell'accesso gestito.
6. Il pool di utenti reindirizza nuovamente all'URI di reindirizzamento dell'applicazione con un codice di autorizzazione.
7. L'applicazione scambia il codice di autorizzazione con il pool di utenti per i token web JSON.
8. L'applicazione scambia il token ID con il pool di identità dell'utente in cambio di credenziali temporanee AWS
9. L'app visualizza il pannello Risultati nell'interfaccia web

   Una volta completata l'autenticazione, vedrai l'interfaccia web che mostra il pannello Risultati e avrai due opzioni per esplorarli:

   a. Pulsante Visualizza solo le credenziali: scegliete questo pulsante se desiderate visualizzare direttamente AWS le credenziali temporanee generate senza i dettagli del flusso API.

   ```
   {
     "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
     "Credentials": {
       "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
       "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
       "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
       "Expiration": "2025-08-12T13:36:17-07:00"
     }
   }
   ```

   b. Pulsante Visualizza il flusso API dettagliato: scegli questo pulsante se desideri visualizzare le richieste API dettagliate.

      - L'app accede all'utente con un pool di utenti Amazon Cognito e ottiene il token ID (JWT) come artefatto. Dopo l'autenticazione con il pool di utenti, l'app riceve una risposta OAuth contenente il token ID (JWT). I pool di identità utilizzano questo token per l'autenticazione:

      ```
      {
         "id_token": "eyJraWQiOiJFWAMPLE......",
         "token_type": "Bearer",
         "expires_in": 3600
      }
      ```

- `GetId()`Richiesta API con l'ID del pool di identità e una `Logins` mappa che include la chiave del provider del pool di utenti e il token ID come valore. Amazon Cognito ha verificato che la firma, la scadenza e il pubblico (aud) del token ID del pool di utenti corrispondano a uno degli ID client dell'app che hai registrato per questo pool di utenti IdP nel pool di identità.

```
POST GetId
{
  "AccountId": "111122223333",
  "IdentityPoolId": "us-east-1:1ac4a76d-1fef-48aa-83af-4224799c0b5c",
  "Logins": {
    "cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE123":
 "eyJraWQiOiJFWAMPLE......"
  }
}
```

Se valido, trova o crea e restituisce quello dell'utente. `IdentityID` Un esempio di risposta è simile al seguente:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetOpenIdToken()`con la stessa `Logins` mappa restituita `IdentityID`

```
POST GetOpenIdToken
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE123":
 "eyJraWQiOiJFWAMPLE......"
  }
}
```

Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Token": "eyJraWQiOiJFWAMPLE......"
}
```

Cosa succede in questa fase: Amazon Cognito emette un token di identità web OpenID Connect di breve durata da cognito-identity.amazonaws.com che lo rappresenta. `IdentityId` Il token include attestazioni OIDC che AWS STS effettuano valutazioni, tra cui aud (ID del pool di identità) e amr (autenticato o non autenticato). La policy di fiducia del tuo ruolo IAM deve richiedere tali attestazioni.

- `AssumeRoleWithWebIdentity()`- La tua app chiama AWS STS direttamente per scambiare il token OpenID di Amazon Cognito con credenziali temporanee AWS

```
POST sts:AssumeRoleWithWebIdentity
{
  "RoleArn": "arn:aws:iam::111122223333:role/Cognito_IdentityPoolAuth_Role",
  "WebIdentityToken": "eyJraWQiOiJFWAMPLE......",
  "RoleSessionName": "CognitoIdentityCredentials"
}
```

Risposta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "FwoGZXIvYXdzEEXAMPLE......",
    "Expiration": "2025-08-12T14:36:17-07:00"
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "AROAW7TIP7EJYEXAMPLE:CognitoIdentityCredentials",
    "Arn": "arn:aws:sts::111122223333:assumed-role/
Cognito_IdentityPoolAuth_Role/CognitoIdentityCredentials"
  }
}
```

Cosa ha fatto la tua applicazione demo: la tua app ha inviato il token OpenID da `GetOpenIdToken()` a AWS STS, richiedendo credenziali temporanee. AWS STS ha eseguito controlli di convalida e ha rilasciato le credenziali:

Comprendi l'accesso al pool di utenti

- Gli utenti del pool di utenti ricevono AWS credenziali temporanee tramite i pool di identità di Amazon Cognito.
- Queste credenziali sono collegate a un ruolo IAM specificato nella configurazione del pool di identità.
- I token ID del pool di utenti vengono scambiati con AWS credenziali tramite il pool di identità.

Autenticazione SAML

Questo passaggio esplora l'autenticazione SAML. Gli utenti possono accedere ai servizi con provider di identità aziendali che supportano SAML. AWS Il flusso di base con SAML non è supportato nell'applicazione di esempio.

SAML authentication with enhanced flow

Questa sezione mostra come utilizzare un provider di identità SAML per accedere a un utente e, utilizzando il flusso avanzato, scambiare l'asserzione SAML in un pool di identità Amazon Cognito con credenziali temporanee AWS per chiamare i servizi. AWS

Usa l'autenticazione SAML con il flusso avanzato del pool di identità

1. Nell'applicazione demo, vai alla sezione Demo interattiva
2. Scegli la scheda Accesso autenticato.
3. Scegli la scheda Flusso avanzato.
4. Scegli Accedi con il provider SAML
5. Completa l'accesso con le tue credenziali aziendali.
6. Il pool di utenti reindirizza nuovamente all'URI di reindirizzamento dell'applicazione con un'asserzione SAML.
7. L'applicazione scambia il codice di autorizzazione con il pool di utenti per i token web JSON.
8. L'applicazione scambia la risposta SAML con il pool di identità dell'utente in cambio di credenziali temporanee AWS
9. L'app visualizza il pannello Risultati nell'interfaccia web

   Una volta completata l'autenticazione, vedrai l'interfaccia web che mostra il pannello Risultati e avrai due opzioni per esplorarli:

a. Pulsante Visualizza solo le credenziali: scegliete questo pulsante se desiderate visualizzare direttamente AWS le credenziali temporanee generate senza i dettagli del flusso API.

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. Pulsante Visualizza il flusso API dettagliato: scegli questo pulsante se desideri visualizzare le richieste API dettagliate.

- L'app accede all'utente con un IdP SAML e ottiene la risposta SAML. I pool di identità accettano le asserzioni SAML dei provider aziendali che utilizzano l'ARN del provider SAML come chiave:

Gestore dell'identità digitale	Chiave del provider Cognito	Scopo
Provider SAML	`arn:aws:iam::111122223333:saml-provider/EXAMPLE`	Asserzioni SAML dell'azienda IdPs

  Dopo una corretta autenticazione con il provider SAML, l'app riceve una risposta SAML tramite HTTP POST all'URL di callback:

```
{
  "saml_response": "PD94bWwgdmVyc2lvbj0iMS4wIiBFWAMPLE...",
  "provider_arn": "arn:aws:iam::111122223333:saml-provider/EXAMPLE",
  "status": "Authentication successful"
}
```

- `GetId()`Richiesta API con l'ID del pool di identità e una `Logins` mappa contenente l'ARN e l'asserzione del provider SAML

```
POST GetId
{
  "AccountId": "111122223333",
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Logins": {
    "arn:aws:iam::111122223333:saml-provider/EXAMPLE":
 "PD94bWwgdmVyc2lvbj0iMS4wIiBFWAMPLE..."
  }
}
```

Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetCredentialsForIdentity()`con la mappa restituita `IdentityID` e la stessa `Logins`

```
POST GetCredentialsForIdentity
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "arn:aws:iam::111122223333:saml-provider/EXAMPLE":
 "PD94bWwgdmVyc2lvbj0iMS4wIiBFWAMPLE..."
  }
}
```

Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......"
  }
}
```

Cos'è successo: Amazon Cognito ha convalidato l'asserzione SAML rispetto al provider configurato, ha scelto un ruolo IAM basato su attributi o regole SAML e ha chiamato per tuo conto. AWS STS

Comprendi l'accesso SAML

- Gli utenti aziendali ricevono AWS credenziali temporanee dai pool di identità di Amazon Cognito dopo l'autenticazione con il proprio provider SAML.
- Ogni utente autenticato riceve un ID di identità univoco che persiste tra le sessioni.
- Queste credenziali sono collegate a un ruolo IAM progettato specificamente per l'accesso autenticato, che fornisce autorizzazioni più ampie rispetto all'accesso come ospite.
- Le asserzioni SAML vengono scambiate con AWS credenziali, mantenendo l'identità dell'utente e gli attributi aziendali.

Autenticazione OpenID Connect (OIDC)

Questo passaggio esplora l'autenticazione OIDC con provider di identità aziendali. Gli utenti possono accedere ai servizi tramite il provider di identità aziendale della propria organizzazione (come Azure AD, Okta o Google Workspace). AWS Continua a leggere se stai cercando informazioni sull'integrazione dell'autenticazione basata su standard con le tue risorse. AWS Dopo aver implementato l'autenticazione OIDC, imparerai come sfruttare le attestazioni OIDC per un controllo granulare degli accessi.

OIDC authentication with enhanced flow

Questa sezione mostra come utilizzare un provider di identità OIDC per accedere a un utente e, utilizzando il flusso avanzato, scambiare il token OIDC in un pool di identità Amazon Cognito con credenziali temporanee per chiamare i servizi. AWS AWS

Usa l'accesso OIDC con il flusso avanzato del pool di identità

1. Nell'applicazione demo, vai alla sezione Demo interattiva
2. Scegli la scheda Accesso autenticato.
3. Scegli la scheda Flusso avanzato.
4. Scegli Accedi con il provider OIDC
5. Completa l'accesso con le tue credenziali aziendali.

6. Il provider OIDC reindirizza nuovamente all'app con un codice di autorizzazione
7. L'applicazione scambia il codice di autorizzazione con il pool di utenti per i token web JSON.
8. L'applicazione invia il token OIDC al pool di identità e recupera le credenziali temporanee. AWS
9. L'app visualizza il pannello Risultati nell'interfaccia web

   Una volta completata l'autenticazione, vedrai l'interfaccia web che mostra il pannello Risultati e avrai due opzioni per esplorarli:

   a. Pulsante Visualizza solo le credenziali: scegliete questo pulsante se desiderate visualizzare direttamente AWS le credenziali temporanee generate senza i dettagli del flusso API.

   ```
   {
     "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
     "Credentials": {
       "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
       "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
       "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
       "Expiration": "2025-08-12T13:36:17-07:00"
     }
   }
   ```

   b. Pulsante Visualizza il flusso API dettagliato: scegli questo pulsante se desideri visualizzare le richieste API dettagliate.
      - L'app accede all'utente con un IdP OIDC e ottiene il token ID. I pool di identità accettano token OIDC di provider aziendali:

Gestore dell'identità digitale	Chiave del provider Cognito	Scopo
Provider OIDC	`example-provider.com/oauth2/default`	Token ID OIDC dell'azienda IdPs

      Dopo una corretta autenticazione con il provider OIDC, l'app riceve una risposta OAuth 2.0 contenente i token:

      ```
      {
      ```

```
  "token_type": "Bearer",
  "expires_in": 3600,
  "access_token": "eyJraWQiOiJFWAMPLE......",
  "scope": "email openid profile",
  "id_token": "eyJraWQiOiJFWAMPLE......"
}
```

- `GetId()`Richiesta API con l'ID del pool di identità e una `Logins` mappa contenente il token del provider OIDC

```
POST GetId
{
  "AccountId": "111122223333",
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Logins": {
    "example-provider.com/oauth2/default": "eyJraWQiOiJFWAMPLE......"
  }
}
```

Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetCredentialsForIdentity()`con la mappa di accesso restituita `IdentityID` e la stessa

```
POST GetCredentialsForIdentity
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "example-provider.com/oauth2/default": "eyJraWQiOiJFWAMPLE......"
  }
}
```

Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
```

```
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......"
  }
}
```

Cos'è successo: Amazon Cognito ha convalidato il token OIDC rispetto al provider configurato, ha scelto un ruolo IAM (predefinito, basato sulle affermazioni o mappato su regole) e ha chiamato per tuo conto. AWS STS

OIDC authentication with basic flow

Questa sezione mostra come utilizzare un provider di identità OIDC per accedere a un utente e, utilizzando il flusso di base, scambiare il token OIDC in un pool di identità Amazon Cognito con credenziali temporanee per chiamare i servizi. AWS AWS

Usa l'accesso OIDC con il flusso di base del pool di identità

1. Nell'applicazione demo, vai alla sezione Demo interattiva
2. Scegli la scheda Accesso autenticato.
3. Scegli la scheda Basic flow.
4. Scegli Accedi con il provider OIDC
5. Completa l'accesso con le tue credenziali aziendali.
6. Il provider OIDC reindirizza nuovamente all'app con un codice di autorizzazione
7. L'applicazione scambia il codice di autorizzazione con il pool di utenti per i token web JSON.
8. L'applicazione invia il token OIDC al pool di identità e recupera le credenziali temporanee. AWS
9. L'app visualizza il pannello Risultati nell'interfaccia web

   Una volta completata l'autenticazione, vedrai l'interfaccia web che mostra il pannello Risultati e avrai due opzioni per esplorarli:

   a. Pulsante Visualizza solo le credenziali: scegliete questo pulsante se desiderate visualizzare direttamente AWS le credenziali temporanee generate senza i dettagli del flusso API.

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. Pulsante Visualizza il flusso API dettagliato: scegli questo pulsante se desideri visualizzare le richieste API dettagliate.

- L'app accede all'utente con un IdP OIDC e ottiene il token ID. I pool di identità accettano token OIDC di provider aziendali:

Gestore dell'identità digitale	Chiave del provider Cognito	Scopo
Provider OIDC	`example-provider.com/oauth2/default`	Token ID OIDC dell'azienda IdPs

Dopo una corretta autenticazione con il provider OIDC, l'app riceve una risposta OAuth 2.0 contenente i token:

```
{
  "token_type": "Bearer",
  "expires_in": 3600,
  "access_token": "eyJraWQiOiJFWAMPLE......",
  "scope": "openid email profile",
  "id_token": "eyJraWQiOiJFWAMPLE......"
}
```

- `GetId()`Richiesta API con l'ID del pool di identità e una `Logins` mappa contenente il token del provider OIDC

```
POST GetId
{
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Logins": {
    "example-provider.com/oauth2/default": "eyJraWQiOiJFWAMPLE......"
  }
```

```
}
```

Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetOpenIdToken()`con l'identityID restituito e la stessa mappa `Logins`

```
POST GetOpenIdToken
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "example-provider.com/oauth2/default": "eyJraWQiOiJFWAMPLE......"
  }
}
```

Risposta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Token": "eyJraWQiOiJFWAMPLE......"
}
```

- `AssumeRoleWithWebIdentity()`con il token OpenID

```
POST AssumeRoleWithWebIdentity
{
  "RoleArn": "arn:aws:iam::111122223333:role/Cognito_IdentityPoolAuth_Role",
  "WebIdentityToken": "eyJraWQiOiJFWAMPLE......"
}
```

Risposta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "FwoGZXIvYXdzEEXAMPLE......",
    "Expiration": "2025-08-12T14:36:17-07:00"
```

```
  }
}
```

Cos'è successo: Amazon Cognito ha convalidato il token OIDC rispetto al provider configurato e ha restituito un token OpenID. L'applicazione è stata chiamata AWS STS direttamente per assumere il ruolo IAM appropriato e ha ricevuto credenziali di breve durata.

Comprendi l'autenticazione OIDC

- Standards-based: OIDC è basato su OAuth 2.0 e fornisce informazioni di identità standardizzate.
- Convalida dei token: i token ID possono essere convalidati per verificarne l'autenticità.
- Claims-based accesso: le attestazioni OIDC possono essere utilizzate per la mappatura dei ruoli e il controllo degli accessi.
- Integrazione aziendale: funziona con i più diffusi provider di identità aziendali.

Fasi successive

Ora che hai configurato ed esplorato l'applicazione demo, puoi:

- Configura provider di identità aggiuntivi che non hai ancora testato
- Sperimenta l'autenticazione avanzata e quella di base per comprenderne le differenze
- Personalizza la demo per il tuo caso d'uso
- Integra Amazon Cognito Identity Pools nelle tue applicazioni.

Sicurezza in Amazon Cognito

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo modello 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. Third-party i revisori 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 ad Amazon Cognito, consulta AWS Services in Scope by Compliance Program Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i tuoi requisiti aziendali e le leggi e le normative applicabili

Questa documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si utilizza Amazon Cognito. Viene illustrato come configurare Amazon Cognito per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amazon Cognito.

Per le protezioni specifiche del pool di utenti contro l'iscrizione, l'accesso e l'acquisizione di account malintenzionati, MFA, autenticazione adattiva e protezione dalle minacce, vedi. Protezione dei pool di utenti con protezione dalle minacce

Indice

- Protezione dei dati in Amazon Cognito
- Identity and Access Management per Amazon Cognito
- Registrazione e monitoraggio in Amazon Cognito
- Accedi ad Amazon Cognito utilizzando un endpoint di interfaccia (AWS PrivateLink)
- Convalida della conformità per Amazon Cognito
- Resilienza in Amazon Cognito

- Sicurezza dell'infrastruttura in Amazon Cognito
- Analisi della configurazione e delle vulnerabilità nei bacini d'utenza di Amazon Cognito
- AWS politiche gestite per Amazon Cognito

Protezione dei dati in Amazon Cognito

Il modello di responsabilità AWS condivisa Modello si applica alla protezione dei dati in Amazon Cognito (Amazon Cognito). Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutto il AWS cloud. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Questo contenuto include le attività di configurazione e gestione della sicurezza per i AWS servizi che utilizzi. Per ulteriori informazioni sulla privacy dei dati, consulta Domande frequenti sulla privacy dei dati.

Ai fini della protezione dei dati, ti consigliamo di proteggere le credenziali degli AWS account e di configurare account utente individuali con AWS Identity and Access Management (IAM). In questo modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere il proprio lavoro. Ti 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.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno AWS dei servizi.
- Utilizza i servizi di sicurezza gestiti avanzati, ad esempio Amazon Macie, che aiutano a individuare e proteggere i dati personali archiviati in Amazon S3.

Ti consigliamo di non inserire mai informazioni identificative sensibili, ad esempio i numeri di account dei clienti, in campi a formato libero, ad esempio un campo Name (Nome). Ciò include quando lavori con Amazon Cognito o altri AWS servizi utilizzando la console, l'API o AWS gli AWS CLI SDK. Gli eventuali dati immessi in Amazon Cognito o altri servizi potrebbero essere prelevati per l'inserimento nei log di diagnostica. Quando fornisci un URL a un server esterno, non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta a tale server.

Crittografia dei dati

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 all'interno dei pool di utenti e dei pool di identità di Amazon Cognito sono crittografati quando sono inattivi in conformità agli standard del settore.

- Amazon Cognito crittografa i dati dei clienti in pool di identità con AWS chiavi di proprietà. Non puoi modificare questo comportamento.
- Per impostazione predefinita, Amazon Cognito crittografa anche i dati dei clienti nei pool di utenti con AWS chiavi di proprietà. Puoi anche configurare i tuoi pool di utenti per crittografare invece le informazioni dei clienti con chiavi gestite dai clienti.

AWS chiave proprietaria

Amazon Cognito crittografa i dati nel tuo pool di utenti o pool di identità con una chiave KMS AWS di proprietà. Le chiavi di questo tipo non sono visibili in. AWS KMS

Chiave gestita dal cliente

Amazon Cognito crittografa i dati del tuo pool di utenti con una chiave gestita dal cliente. Sei responsabile dell'amministrazione delle politiche chiave gestite dal cliente, della rotazione e dell'eliminazione pianificata.

> **Note**
>
> La crittografia con chiavi gestite dal cliente potrebbe non essere disponibile in alcuni pool di utenti. Newly-created i pool di utenti dispongono sempre di questa forma di crittografia.

Cose da sapere sulla crittografia dei pool di utenti con chiavi gestite dal cliente

1. Tutti i dati dei clienti nei pool di utenti sono crittografati quando sono inattivi, anche se non si interviene per configurare le impostazioni di crittografia.
2. Amazon Cognito supporta solo chiavi KMS simmetriche nello stesso pool di utenti per la Regione AWS crittografia del pool di utenti inattivi. Non puoi configurare la crittografia del pool di utenti

inattivi con chiavi asimmetriche. È possibile configurare la crittografia inattiva con chiavi a regione singola e con chiavi multiregionali che si trovano nella stessa regione del pool di utenti.

3. È possibile configurare la crittografia del pool di utenti solo con una chiave KMS ARN, non con un alias.

Crittografia PII

Amazon Cognito supporta la riservatezza, l'integrità e la disponibilità delle informazioni di identificazione personale (PII) nelle ricerche di attributi utente con crittografia ricercabile. Queste funzioni del Hash-based Message Authentication Code (HMAC), ottimizzate per le prestazioni per i set di dati del pool di utenti, mappano tra il testo semplice e i valori crittografati degli attributi utente. Amazon Cognito calcola i valori HMAC con la chiave KMS che crittografa il pool di utenti. Questa protezione si applica ai seguenti attributi:

- `sub`
- `email`
- `phone_number`
- `given_name`
- `family_name`
- `name`
- `username`
- `preferred_username`
- `cognito:user_status`

Le seguenti procedure configurano la crittografia a riposo nel pool di utenti. Per ulteriori informazioni sulle politiche chiave KMS che delegano l'accesso ad applicazioni Servizi AWS come Amazon Cognito, consulta Autorizzazioni per Amazon Cognito nelle politiche chiave.

Set customer managed key policy

> Per utilizzare una chiave gestita dal cliente, la chiave deve affidarsi a un responsabile del servizio Amazon Cognito per eseguire operazioni di crittografia e decrittografia sulla chiave. Configura la politica chiave della tua chiave KMS come mostrato nell'esempio seguente. Il responsabile IAM che scrive questa policy deve avere accesso in scrittura alla tua chiave KMS, con `kms:PutKeyPolicy` autorizzazione.

```
{
    "Id": "cognito-cmk-policy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Amazon Cognito service access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cognito-idp.amazonaws.com",
                    "identitystore.amazonaws.com"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKeyWithoutPlainText"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": [
                        "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-
east-1_EXAMPLE"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:cognito-idp:userpool-arn":
 "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        },
        {
            "Sid": "Allow Amazon Cognito service DescribeKey access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
```

```
                    "cognito-idp.amazonaws.com",
                    "identitystore.amazonaws.com"
                ]
            },
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": [
                        "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-
east-1_EXAMPLE"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                }
            }
        },
        {
            "Sid": "Allow access through Amazon Cognito for all principals in
 account that are authorized to use Amazon Cognito",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "cognito-idp.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "Allow access through Amazon Cognito for all principals in
 account that are authorized to use Amazon Cognito",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKeyWithoutPlainText"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "cognito-idp.us-east-1.amazonaws.com"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:cognito-idp:userpool-arn":
 "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        },
        {
            "Sid": "Allow administrators to manage the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

Set customer managed key policy with permissions for encryption of exported logs

Applica questa politica alla tua chiave KMS quando hai configurato l'esportazione dei log. Questa politica consente ad Amazon Cognito di crittografare i log durante l'esportazione e al servizio intermedio di decrittografarli prima di scriverli nel tuo gruppo di log Logs. CloudWatch Configura la politica chiave della tua chiave KMS come mostrato nell'esempio seguente. Il responsabile IAM che scrive questa policy deve avere accesso in scrittura alla tua chiave KMS, con `kms:PutKeyPolicy` autorizzazione.

```
{
    "Id": "cognito-cmk-policy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Amazon Cognito service access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cognito-idp.amazonaws.com",
                    "identitystore.amazonaws.com"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKeyWithoutPlainText"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": [
                        "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-
east-1_EXAMPLE"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:cognito-idp:userpool-arn":
 "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        },
        {
            "Sid": "Allow Amazon Cognito service DescribeKey access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
```

```
                    "cognito-idp.amazonaws.com",
                    "identitystore.amazonaws.com"
                ]
            },
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": [
                        "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-
east-1_EXAMPLE"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                }
            }
        },
        {
            "Sid": "Allow access through Amazon Cognito for all principals in
 account that are authorized to use Amazon Cognito",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "cognito-idp.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "Allow access through Amazon Cognito for all principals in
 account that are authorized to use Amazon Cognito",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKeyWithoutPlainText"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "cognito-idp.us-east-1.amazonaws.com"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:cognito-idp:userpool-arn":
 "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        },
        {
            "Sid": "Allow Amazon Cognito service log delivery access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cognito-idp.amazonaws.com"
                ]
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Encrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:SourceArn": "arn:aws:logs:us-
east-1:111122223333:*"
                }
            }
        },
        {
```

```
            "Sid": "Allow IngestionHub service log delivery access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "delivery.logs.amazonaws.com"
                ]
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:SourceArn": "arn:aws:logs:us-
east-1:111122223333:*"
                }
            }
        },
        {
            "Sid": "Allow administrators to manage the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

Configure encryption at rest in the console

Per configurare la crittografia inattiva in un pool di utenti

1. Passa alla console Amazon Cognito. Potrebbe comparire una richiesta di inserimento delle credenziali AWS .
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli il menu Impostazioni e vai alla scheda Sicurezza del pool di utenti. Individua Encryption at rest e seleziona Modifica.

5. In Tipo di chiave, seleziona la chiave AWS di proprietà o la chiave gestita dal cliente.

   a. Se hai selezionato la chiave AWS proprietaria, non è richiesta alcuna configurazione aggiuntiva.

   b. Se hai selezionato Chiave gestita dal cliente, inserisci l'ARN di una chiave KMS nell'ARN della chiave gestita dal cliente. Puoi anche scegliere Crea una AWS KMS chiave e aprire una nuova finestra nella AWS KMS console per creare una nuova chiave KMS.

6. Scegli Save changes (Salva modifiche).

Configure encryption at rest with the API

Imposta la configurazione delle chiavi in una richiesta CreateUserPoolo UpdateUserPoolAPI. Il seguente esempio parziale del corpo della richiesta imposta un pool di utenti per utilizzare la chiave gestita dal cliente fornita. Per un esempio completo di richiesta, vedi Esempi.

```
"KeyConfiguration": {
   "KeyType": "CUSTOMER_MANAGED_KEY",
   "KmsKeyArn": "arn:aws:kms:us-east-1:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
},
```

Il seguente corpo di richiesta di esempio parziale imposta un pool di utenti per utilizzare una chiave AWS di proprietà.

```
"KeyConfiguration": {
   "KeyType": "AWS_OWNED_KEY"
},
```

Se la DescribeUserPoolrisposta non include un `KeyConfiguration` parametro, il pool di utenti è configurato per crittografare i dati inattivi con una chiave AWS proprietaria.

Crittografia dei dati in transito

In quanto servizio gestito, Amazon Cognito è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi di AWS 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 ad Amazon Cognito 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 crittografia con Perfect Forward Secrecy (PFS) come DHE (Ephemeral) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

I pool di utenti e i pool di identità di Amazon Cognito dispongono di operazioni IAM-authenticated API non autenticate e autorizzate tramite token. Alcune sono operazioni di tipo piano di controllo per operazioni amministrative come la configurazione di un dominio di pool di utenti, mentre altre sono operazioni di tipo data plane per l'autenticazione. Per ulteriori informazioni, consulta Elenco delle operazioni API raggruppate per modello di autorizzazione. Tutte le classi di operazioni API di Amazon Cognito condividono uno spazio dei nomi (per i pool di utenti, `cognito-idp` `cognito-identity` per i pool di identità) e gli endpoint di servizio.AWS gli endpoint di servizio richiedono la crittografia in transito con una versione TLS minima di 1.2.

I pool di utenti di Amazon Cognito ospitano l'accesso gestito e la classica interfaccia utente ospitata su domini Web serviti da distribuzioni Amazon di proprietà del servizio. CloudFront Amazon Cognito gestisce le impostazioni per la crittografia in transito su tali distribuzioni. Per ulteriori informazioni sulle impostazioni di crittografia per l'accesso gestito, consulta la versione TLS nel capitolo Accesso gestito.

Identity and Access Management per Amazon Cognito

AWS Identity and Access Management (IAM) è un programma Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (chi può effettuare l'accesso) e autorizzato (chi dispone delle autorizzazioni) a utilizzare risorse Amazon Cognito. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell’accesso tramite policy

- Funzionamento di Amazon Cognito con IAM
- Identity-based esempi di policy per Amazon Cognito
- Risoluzione dei problemi relativi all'identità e all'accesso di Amazon Cognito
- Utilizzo di ruoli collegati ai servizi per Amazon Cognito

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta Risoluzione dei problemi relativi all'identità e all'accesso di Amazon Cognito)
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta Funzionamento di Amazon Cognito con IAM)
- Amministratore IAM: scrivi policy per gestire l'accesso (consulta Identity-based esempi di policy per Amazon Cognito)

Autenticazione con identità

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

Puoi accedere come identità federata utilizzando credenziali provenienti 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.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata 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 dell'utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare Centro identità AWS IAM. Per ulteriori informazioni, consulta Che cos'è il Centro identità IAM? nella Guida per l'utente di Centro identità AWS IAM .

Utenti e gruppi 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 Guida per l'utente IAM.

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.

Ruoli IAM

Un ruolo IAM è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo passando da un ruolo utente a un ruolo IAM (console) o chiamando un'operazione AWS CLI o AWS API. 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.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste

politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Identity-based politiche

Identity-based le politiche sono documenti di policy sulle autorizzazioni JSON che alleghi a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Identity-based le politiche possono essere politiche in linea (incorporate direttamente in una singola identità) o politiche gestite (politiche autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scegliere tra policy gestite e policy in linea nella Guida per l'utente di IAM.

Resource-based politiche

Resource-based le politiche sono documenti di policy JSON allegati a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio specificare un'entità principale.

Resource-based le politiche sono politiche in linea che si trovano in quel servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Policy di controllo dei servizi (SCP): specifica il numero massimo di autorizzazioni per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Policy di controllo delle risorse (RCP): imposta le autorizzazioni massime disponibili per le risorse degli account. Per ulteriori informazioni, consulta Policy di controllo delle risorse (RCP) nella Guida per l'utente di AWS Organizations .
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di politiche, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

Funzionamento di Amazon Cognito con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon Cognito, scopri quali funzionalità di IAM sono disponibili per l'uso con Amazon Cognito.

Funzionalità di IAM che puoi utilizzare con Amazon Cognito

Caratteristiche di IAM	Supporto per Amazon Cognito
Identity-based politiche	Sì
Resource-based politiche	No
Operazioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì

Caratteristiche di IAM	Supporto per Amazon Cognito
Liste di controllo degli accessi (ACL)	No
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì
Autorizzazioni del principale	No
Ruoli di servizio	Sì
Service-linked ruoli	Sì

Per avere una visione di alto livello di come Amazon Cognito e AWS altri servizi funzionano con la maggior parte delle funzionalità IAM, AWS consulta i servizi che funzionano con IAM nella IAM User Guide.

Identity-based politiche per Amazon Cognito

Supporta le policy basate sull'identità: sì

Identity-based le policy sono documenti relativi alle policy in materia di autorizzazioni JSON che puoi allegare a un'identità, ad esempio un utente IAM, un gruppo di utenti o un ruolo. Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente IAM.

Identity-based esempi di policy per Amazon Cognito

Per visualizzare esempi di policy basate su identità Amazon Cognito, consulta Identity-based esempi di policy per Amazon Cognito.

Resource-based politiche all'interno di Amazon Cognito

Supporta le policy basate su risorse: no

Resource-based le politiche sono documenti di policy JSON allegati a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy di bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio specificare un'entità principale. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

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. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Operazioni delle policy per Amazon Cognito

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni di eseguire l'operazione associata.

Per visualizzare un elenco di operazioni di Amazon Cognito, consulta Operazioni definite da Amazon Cognito in Riferimento per l'autorizzazione del servizio.

Le operazioni delle policy in Amazon Cognito utilizzano il seguente prefisso prima dell'operazione:

```
cognito-identity
```

Per specificare più operazioni in una sola istruzione, separarle con la virgola.

```
"Action": [
      "cognito-identity:action1",
      "cognito-identity:action2"
         ]
```

API firmate rispetto ad API senza firma

Quando firmi le richieste API di Amazon Cognito con AWS credenziali, puoi limitarle in una policy AWS Identity and Access Management (IAM). Richieste API che è necessario firmare con le credenziali AWS includono l'accesso lato server con `AdminInitiateAuth` e operazioni che creano, visualizzano o modificano le risorse Amazon Cognito come `UpdateUserPool`. Per ulteriori informazioni sulle richieste API firmate, consulta Firmare le richieste AWS API.

Poiché Amazon Cognito è un prodotto di identità dei consumer per le app che si desidera rendere disponibili al pubblico, l'utente ha accesso alle seguenti API non firmate. L'app effettua queste richieste API per gli utenti e i potenziali utenti. Alcune API non richiedono alcuna autorizzazione preventiva, ad esempio, `InitiateAuth` per iniziare una nuova sessione di autenticazione. Alcune API utilizzano token di accesso o chiavi di sessione per l'autorizzazione, come `VerifySoftwareToken` per completare la configurazione MFA per un utente con una sessione autenticata esistente. Un'API dei pool di utenti di Amazon Cognito non firmata e autorizzata supporta una `Session` o il parametro `AccessToken` nella sintassi della richiesta, come mostrato nella Documentazione di riferimento dell'API di Amazon Cognito. Un'API di Amazon Cognito Identity non firmata supporta un parametro `IdentityId` come mostrato nella Documentazione di riferimento dell'API di identità federate di Amazon Cognito.

Per ulteriori informazioni sui modelli di autorizzazione e i ruoli delle operazioni API dei pool di utenti Amazon Cognito, consultare Elenco delle operazioni API raggruppate per modello di autorizzazione.

Operazioni API dei pool di identità di Amazon Cognito

- `GetId`
- `GetOpenIdToken`
- `GetCredentialsForIdentity`
- `UnlinkIdentity`

Operazioni API dei pool di utenti Amazon Cognito

- `AssociateSoftwareToken`
- `ChangePassword`
- `ConfirmDevice`
- `ConfirmForgotPassword`
- `ConfirmSignUp`

- DeleteUser
- DeleteUserAttributes
- ForgetDevice
- ForgotPassword
- GetDevice
- GetUser
- GetUserAttributeVerificationCode
- GlobalSignOut
- InitiateAuth
- ListDevices
- ResendConfirmationCode
- RespondToAuthChallenge
- RevokeToken
- SetUserMFAPreference
- SetUserSettings
- SignUp
- UpdateAuthEventFeedback
- UpdateDeviceStatus
- UpdateUserAttributes
- VerifySoftwareToken
- VerifyUserAttribute

Per visualizzare esempi di policy basate su identità Amazon Cognito, consulta Identity-based esempi di policy per Amazon Cognito.

Risorse di policy per Amazon Cognito

Supporta le risorse relative alle policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Amazon Resource Names (ARN)

ARN per identità federate di Amazon Cognito

Nei pool di identità (identità federate) di Amazon Cognito è possibile limitare l'accesso di un utente IAM a uno specifico pool di identità attraverso il formato Amazon Resource Name (ARN), come nell'esempio riportato di seguito. Per ulteriori informazioni sugli ARN, consultare Identificatori IAM.

```
arn:aws:cognito-identity:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID
```

ARN per Amazon Cognito Sync

In Amazon Cognito Sync, i clienti possono inoltre limitare l'accesso in base all'ID del pool di identità, all'ID identità e al nome del set di dati.

Per le API che operano su un pool di identità, il formato dell'ARN del pool di identità corrisponde a quello delle identità federate di Amazon Cognito, a eccezione del nome del servizio, ovvero `cognito-sync` al posto di `cognito-identity`:

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID
```

Per le API che operano su una identità singola, come `RegisterDevice`, puoi riferirti all'identità individuale con il formato ARN seguente:

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID/
identity/IDENTITY_ID
```

Per le API che operano su set di dati, come `UpdateRecords` e `ListRecords`, puoi riferirti al set di dati individuale utilizzando il formato ARN seguente:

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID/
identity/IDENTITY_ID/dataset/DATASET_NAME
```

ARN per i bacini d'utenza di Amazon Cognito

Per i pool di utenti di Amazon Cognito, è possibile limitare l'accesso di un utente a un pool di utenti specifico, utilizzando il formato ARN seguente:

```
arn:aws:cognito-idp:REGION:ACCOUNT_ID:userpool/USER_POOL_ID
```

Per visualizzare un elenco di tipi di risorse di Amazon Cognito e i relativi ARN, consulta Risorse definite da Amazon Cognito in Riferimento per l'autorizzazione del servizio. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta Operazioni definite da Amazon Cognito.

Per visualizzare esempi di policy basate su identità Amazon Cognito, consulta Identity-based esempi di policy per Amazon Cognito.

Chiavi di condizione delle policy per Amazon Cognito

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di Amazon Cognito, consulta Chiavi di condizione per Amazon Cognito in Riferimento per l'autorizzazione del servizio. Per informazioni su operazioni e risorse con cui è possibile utilizzare una chiave di condizione, consulta Operazioni definite da Amazon Cognito.

Per visualizzare esempi di policy basate su identità Amazon Cognito, consulta Identity-based esempi di policy per Amazon Cognito.

Liste di controllo degli accessi (ACL) in Amazon Cognito

Supporta le ACL: no

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Attribute-based controllo degli accessi (ABAC) con Amazon Cognito

Supporta ABAC (tag nelle policy): parzialmente

Attribute-based il controllo degli accessi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base ad attributi chiamati tag. È possibile allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'elemento 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 Parziale.

Per maggiori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con Amazon Cognito

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono l'accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM e Servizi AWS compatibili con IAM nella Guida per l'utente IAM.

Cross-service autorizzazioni principali per Amazon Cognito

Supporta l'inoltro delle sessioni di accesso (FAS): no

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

Ruoli di servizio per Amazon Cognito

Supporta i ruoli di servizio: sì

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.

Per ulteriori informazioni sui ruoli di servizio in Amazon Cognito, consulta Attivazione della sincronizzazione push e Implementazione della sincronizzazione push.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità di Amazon Cognito. Modifica i ruoli del servizio solo quando Amazon Cognito fornisce le indicazioni per farlo.

Service-linked ruoli per Amazon Cognito

Supporta i ruoli collegati ai servizi: Sì

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 appaiono nel tuo Account AWS account e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Per maggiori dettagli su come creare e gestire i ruoli collegati ai servizi di Amazon Cognito, consulta Utilizzo di ruoli collegati ai servizi per Amazon Cognito.

Identity-based esempi di policy per Amazon Cognito

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse Amazon Cognito. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) nella Guida per l'utente di IAM.

Per informazioni dettagliate sulle operazioni e sui tipi di risorse definiti da Amazon Cognito, incluso il formato degli ARN per ogni tipo di risorse, consulta Operazioni, risorse e chiavi di condizione per Amazon Cognito in Service Authorization Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console Amazon Cognito
- Consentire agli utenti di visualizzare le loro autorizzazioni
- Limitazione dell'accesso alla console ad uno specifico pool di identità
- Consentire l'accesso a un set di dati specifico per tutte le identità di un pool

Best practice per le policy

Identity-based le politiche determinano se qualcuno può creare, accedere o eliminare le risorse di Amazon Cognito nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le politiche AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate

utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.

- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e suggerimenti utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare Protezione dell'accesso API con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

> **Note**
>
> La versione originale e quella nuova della console Amazon Cognito hanno un comportamento di base diverso quando si visualizzano e si modificano le proprie risorse Amazon Cognito. Se hai concesso l'autorizzazione alle operazioni nel prefisso del servizio `cognito-idp` solo quando la condizione `aws:ViaAWSService` è True, è possibile che il principale IAM interessato sia stato effettivo per le risorse Amazon Cognito nella console originale, ma non nella nuova console. Per lavorare nella console di Amazon Cognito, non impostare una condizione `aws:ViaAWSService` nelle autorizzazioni di Amazon Cognito nella policy IAM.

Utilizzo della console Amazon Cognito

Per accedere alla console Amazon Cognito, è necessario disporre di un set di autorizzazioni minimo. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amazon Cognito presenti nel tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso AWS CLI o l'API. AWS Al contrario, è opportuno concedere l'accesso solo alle azioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano continuare a utilizzare la console Amazon Cognito, collega anche Amazon `ConsoleAccess` Cognito `ReadOnly` AWS o la policy gestita alle entità. Per maggiori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente di IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o in modo programmatico. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Limitazione dell'accesso alla console ad uno specifico pool di identità

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cognito-identity:ListIdentityPools"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cognito-identity:*"
            ],
            "Resource": "arn:aws:cognito-identity:us-
east-1:111122223333:identitypool/us-east-1:1a1a1a1a-ffff-1111-9999-12345678"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cognito-sync:*"
            ],
            "Resource": "arn:aws:cognito-sync:us-
east-1:111122223333:identitypool/us-east-1:1a1a1a1a-ffff-1111-9999-12345678"
        }
    ]
}
```

Consentire l'accesso a un set di dati specifico per tutte le identità di un pool

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cognito-sync:ListRecords",
                "cognito-sync:UpdateRecords"
            ],
            "Resource": "arn:aws:cognito-sync:us-
east-1:111122223333:identitypool/us-east-1:1a1a1a1a-ffff-1111-9999-12345678/
identity/*/dataset/UserProfile"
        }
    ]
}
```

Risoluzione dei problemi relativi all'identità e all'accesso di Amazon Cognito

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon Cognito e IAM.

Argomenti

- Non sono autorizzato a eseguire un'operazione in Amazon Cognito
- Non sono autorizzato a eseguire iam: PassRole
- Sono un amministratore e desidero consentire ad altri utenti di accedere a Amazon Cognito
- Voglio consentire l'accesso a persone esterne al mio AWS account per accedere alle mie risorse Amazon Cognito

Non sono autorizzato a eseguire un'operazione in Amazon Cognito

Se ricevi un errore che indica che non sei autorizzato a eseguire un’operazione, le tue policy devono essere aggiornate per poter eseguire l’operazione.

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 `cognito-identity:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cognito-identity:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `cognito-identity:`*`GetWidget`*.

Se hai bisogno di assistenza, contatta il tuo amministratore. AWS L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se si riceve un errore che indica che non si dispone dell'autorizzazione a eseguire l'operazione `iam:PassRole`, per passare un ruolo a Amazon Cognito è necessario aggiornare le policy.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` prova a utilizzare la console per eseguire un'operazione in Amazon Cognito. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per trasmettere il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Sono un amministratore e desidero consentire ad altri utenti di accedere a Amazon Cognito

Per consentire ad altri di accedere ad Amazon Cognito, devi concedere l'autorizzazione alle persone o alle applicazioni che devono accedervi. Se lo utilizzi Centro identità AWS IAM per gestire persone

e applicazioni, assegni set di autorizzazioni a utenti o gruppi per definirne il livello di accesso. I set di autorizzazioni creano e assegnano automaticamente le policy IAM ai ruoli IAM associati alla persona o all'applicazione. Per ulteriori informazioni, consulta Set di autorizzazioni nella Guida per l'Centro identità AWS IAM utente.

Se non utilizzi IAM Identity Center, devi creare entità IAM (utenti o ruoli) per le persone o le applicazioni che necessitano di accesso. Dovrai quindi collegare all'entità una policy che conceda le autorizzazioni corrette in Amazon Cognito. Dopo aver concesso le autorizzazioni, fornisci le credenziali all'utente o allo sviluppatore dell'applicazione. Utilizzeranno tali credenziali per accedere. AWSPer ulteriori informazioni sulla creazione di utenti, gruppi, policy e autorizzazioni IAM, consulta IAM Identities and Policies and permissions in IAM nella IAM User Guide.

Voglio consentire l'accesso a persone esterne al mio AWS account per accedere alle mie risorse Amazon Cognito

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo degli accessi (ACL), utilizzare tali policy per concedere alle persone l'accesso alle proprie risorse.

Per maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon Cognito supporta queste funzionalità, consulta Funzionamento di Amazon Cognito con IAM.
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente di IAM.

Utilizzo di ruoli collegati ai servizi per Amazon Cognito

Amazon Cognito utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo unico di ruolo IAM con una politica di fiducia che consente a un utente di assumere il ruolo. Servizio AWS Service-linked i ruoli sono predefiniti da Amazon Cognito e includono tutte le autorizzazioni richieste dal servizio per chiamare AWS altri servizi per tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di Amazon Cognito perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. Amazon Cognito definisce le autorizzazioni dei ruoli associati ai servizi e, salvo diversamente definito, solo Amazon Cognito può assumere il ruolo. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato al servizio solo dopo avere eliminato le risorse correlate. Questa procedura protegge le risorse di Amazon Cognito perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni su altri servizi che supportano i ruoli collegati ai servizi, consulta AWS Servizi che funzionano con IAM e cerca i servizi con Sì nella colonna Ruolo. Service-Linked Scegliere Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Service-linked autorizzazioni di ruolo per Amazon Cognito

Amazon Cognito utilizza i seguenti ruoli collegati ai servizi:

- AWSServiceRoleForAmazonCognitoIdpEmailService— Consente al servizio di pool di utenti Amazon Cognito di utilizzare le identità Amazon SES per l'invio di e-mail.

- AWSServiceRoleForAmazonCognitoIdp— Consente ai pool di utenti di Amazon Cognito di pubblicare eventi e configurare endpoint per i tuoi progetti Amazon Pinpoint.

AWSServiceRoleForAmazonCognitoIdpEmailService

Ai fini dell'assunzione del ruolo, il ruolo collegato al servizio `AWSServiceRoleForAmazonCognitoIdpEmailService` considera attendibili i seguenti servizi:

- `email.cognito-idp.amazonaws.com`

La policy delle autorizzazioni del ruolo consente ad Amazon Cognito di completare le seguenti operazioni sulle risorse specificate:

Azioni consentite per: AWSServiceRoleForAmazonCognitoIdpEmailService

- Operazione: `ses:SendEmail` e `ses:SendRawEmail`
- Risorsa: *

La policy impedisce a Amazon Cognito di completare le seguenti operazioni sulle risorse specificate:

Operazioni rifiutate

- Operazione: `ses:List*`
- Risorsa: *

Con queste autorizzazioni, Amazon Cognito può utilizzare gli indirizzi e-mail verificati in Amazon SES solo per inviare e-mail ai tuoi utenti. Amazon Cognito invia e-mail agli utenti quando eseguono alcune operazioni nell'app client per un bacino d'utenza, ad esempio quando accedono o reimpostano una password.

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 le autorizzazioni dei Service-linked ruoli nella Guida per l'utente IAM.

AWSServiceRoleForAmazonCognitoIdp

Il ruolo AWSServiceRoleForAmazonCognitoIdp collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `email.cognito-idp.amazonaws.com`

La policy delle autorizzazioni del ruolo consente ad Amazon Cognito di completare le seguenti operazioni sulle risorse specificate:

Azioni consentite per AWSServiceRoleForAmazonCognitoIdp

- Operazione: `cognito-idp:Describe`
- Risorsa: *

Con questa autorizzazione, Amazon Cognito può chiamare le operazioni API `Describe` di Amazon Cognito per tuo conto.

> **Note**
>
> Quando integri Amazon Cognito con Amazon Pinpoint `createUserPoolClient` utilizzando `updateUserPoolClient` e, le autorizzazioni delle risorse vengono aggiunte alla SLR come policy in linea. La policy in linea fornisce e autorizzazioni. `mobiletargeting:UpdateEndpoint` `mobiletargeting:PutEvents` Queste autorizzazioni consentono ad Amazon Cognito di pubblicare eventi e configurare endpoint per i progetti Pinpoint integrati con Cognito.

Creazione di un ruolo collegato ai servizi per Amazon Cognito

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando configuri un pool di utenti per utilizzare la configurazione di Amazon SES per gestire la Console di gestione AWS consegna delle e-mail nell'API Amazon Cognito AWS CLI, Amazon Cognito crea il ruolo collegato al servizio per te.

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando configuri un bacino d'utenza per utilizzare la configurazione di Amazon SES per gestire il recapito e-mail, Amazon Cognito crea il ruolo collegato al servizio per tuo conto.

Prima che Amazon Cognito possa creare questo ruolo, le autorizzazioni IAM utilizzate per configurare il bacino d'utenza devono includere l'operazione `iam:CreateServiceLinkedRole`. Per ulteriori informazioni sull'aggiornamento delle autorizzazioni in IAM, consulta Modifica delle autorizzazioni per un utente IAM nella Guida per l'utente di IAM.

Modifica di un ruolo collegato ai servizi per Amazon Cognito

Non puoi modificare i ruoli AmazonCognitoIdp o AmazonCognitoIdpEmailService quelli collegati al servizio in. AWS Identity and Access Management Dopo aver creato un ruolo collegato al servizio, non è possibile modificarne il nome, perché potrebbero farvi riferimento diverse entità. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Amazon Cognito

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, consigliamo di eliminare il ruolo. Se elimini il ruolo, mantieni solo le entità che Amazon Cognito monitora o mantiene attivamente. Prima di poter eliminare i ruoli AmazonCognitoIdp o AmazonCognitoIdpEmailService quelli collegati al servizio, è necessario eseguire una delle seguenti operazioni per ogni pool di utenti che utilizza il ruolo:

- Eliminare il bacino d'utenza.
- Aggiornare le impostazioni e-mail nel bacino d'utenza in modo da utilizzare la funzionalità e-mail predefinita. L'impostazione di default non utilizza il ruolo collegato al servizio.

Ricordati di eseguire l'azione in ognuno di essi Regione AWS con un pool di utenti che utilizza il ruolo.

> **Note**
>
> Se il servizio Amazon Cognito utilizza tale 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.

Come eliminare un bacino d'utenza Amazon Cognito

1. Accedi Console di gestione AWS e apri la console Amazon Cognito all'indirizzo. https://console.aws.amazon.com/cognito
2. Scegli Manage User Pools (Gestisci pool di utenti).
3. Nella pagina Your User Pools (I tuoi pool di utenti), scegli il bacino d'utenza da eliminare.
4. Scegli Delete pool (Elimina pool).
5. Nella finestra Delete user pool (Elimina pool di utenti), digita **`delete`** e seleziona Elimina pool.

Per aggiornare un bacino d'utenza di Amazon Cognito in modo che utilizzi la funzionalità e-mail di default

1. Accedi Console di gestione AWS e apri la console Amazon Cognito all'indirizzo. https://console.aws.amazon.com/cognito

2. Scegli Manage User Pools (Gestisci pool di utenti).
3. Nella pagina Your User Pools (I tuoi pool di utenti), scegli il bacino d'utenza da aggiornare.
4. Nel menu di navigazione a sinistra, seleziona Message customizations (Personalizzazioni di messaggio).
5. Alla voce Do you want to send emails through your Amazon SES Configuration? (Inviare e-mail tramite la configurazione Amazon SES?), seleziona No - Use Cognito (Default) (No, utilizza Cognito (di default)).
6. Una volta completata la configurazione delle opzioni dell'account e-mail, seleziona Save changes (Salva modifiche).

Per eliminare manualmente il ruolo collegato ai servizi utilizzando IAM

Usa la console IAM AWS CLI, o l' AWS API per eliminare ruoli AmazonCognitoIdp o collegati ai AmazonCognitoIdpEmailService servizi. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di Amazon Cognito

Amazon Cognito supporta ruoli collegati al servizio Regioni AWS ovunque il servizio sia disponibile. Per ulteriori informazioni, consulta Regioni AWS ed endpoint.

Registrazione e monitoraggio in Amazon Cognito

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon Cognito e delle altre AWS soluzioni. Amazon Cognito attualmente supporta i seguenti Servizi AWS in modo da poter monitorare l'organizzazione e le relative attività.

- AWS CloudTrail — Con CloudTrail puoi acquisire chiamate API dalla console Amazon Cognito e da chiamate in codice alle operazioni dell'API Amazon Cognito. Ad esempio, quando un utente si autentica, CloudTrail può registrare dettagli come l'indirizzo IP nella richiesta, chi ha effettuato la richiesta e quando è stata effettuata.
- Amazon CloudWatch Logs: con CloudWatch Logs, puoi inviare log dettagliati delle attività degli utenti a un gruppo di log. Ad esempio, puoi esaminare i log dettagliati delle attività degli utenti per risolvere i problemi relativi alla consegna di e-mail e messaggi SMS agli utenti.
- Amazon CloudWatch Metrics: con le CloudWatch metriche puoi monitorare, segnalare e intraprendere azioni automatiche in caso di evento quasi in tempo reale. Ad esempio, puoi creare CloudWatch dashboard sulle metriche fornite per monitorare i tuoi pool di utenti di Amazon Cognito

oppure puoi CloudWatch creare allarmi sulle metriche fornite per avvisarti in caso di violazione di una soglia prestabilita.

- Amazon CloudWatch Logs Insights: con CloudWatch Logs Insights, puoi configurare l'invio di eventi CloudWatch per CloudTrail il monitoraggio dei file di registro di Amazon CloudTrail Cognito.

Argomenti

- Monitoraggio e gestione dei costi
- Esportazione dei log dai pool di utenti di Amazon Cognito
- Monitoraggio delle quote e dell'utilizzo in CloudWatch e Service Quotas
- Accesso ad Amazon Cognito AWS CloudTrail

Monitoraggio e gestione dei costi

Come per qualsiasi altra cosa Servizio AWS, è importante comprendere l'effetto della configurazione e dell'utilizzo di Amazon Cognito sulla bolletta AWS . Come parte dei preparativi per l'implementazione dei pool di utenti nella produzione, imposta il monitoraggio e le misure di protezione per l'attività e il consumo di risorse. Quando sapete a chi rivolgervi e quali azioni generano costi aggiuntivi, potete inserire nella fattura precauzioni contro le sorprese.

Amazon Cognito addebita i costi per le seguenti dimensioni di utilizzo.

- Pool di utenti attivi mensili (MAU): la tariffa varia in base al piano di funzionalità
- I MAU del pool di utenti hanno effettuato l'accesso con la federazione OIDC o SAML
- Richiedete il volume per l'autorizzazione da macchina a macchina (M2M), sia per quanto riguarda le `client_credentials` concessioni sull'endpoint del token che per il funzionamento dell'API `GetClientToken`
- Utilizzo acquistato superiore alle quote predefinite per alcune categorie di API per pool di utenti

Inoltre, le funzionalità del tuo pool di utenti come messaggi e-mail, messaggi SMS e trigger Lambda possono comportare costi in servizi dipendenti. Per una panoramica completa, consulta i prezzi di Amazon Cognito.

Visualizzazione e previsione dei costi

High-volume eventi come il lancio di prodotti e l'apertura a nuove basi di utenti possono aumentare il numero di MAU e avere un impatto sui costi. Stima in anticipo il numero di nuovi utenti e osserva

l'attività in tempo reale. Potresti scoprire di voler adattare il volume con l'acquisto di una capacità di quota aggiuntiva o controllare il volume con misure di sicurezza aggiuntive.

Puoi visualizzare e generare report sui AWS costi nella Gestione costi e fatturazione AWS console. Puoi trovare gli addebiti più recenti per Amazon Cognito nella sezione Fatturazione e pagamenti. In Fatture, addebiti per servizio, filtra `Cognito` per visualizzare l'utilizzo. Per ulteriori informazioni, consulta Visualizzazione della fattura nella Guida per l'utente di AWS Billing .

Per monitorare i tassi di richiesta API, esamina la metrica di utilizzo nella console Service Quotas. Ad esempio, sia le richieste di `client_credentials` concessione all'endpoint del token che le richieste GetClientTokenAPI vengono visualizzate come Frequenza delle ClientAuthentication richieste, poiché condividono una quota di richieste. Nella fattura, queste richieste sono associate al client dell'app che le ha prodotte. Con queste informazioni, è possibile allocare equamente i costi agli inquilini in un'architettura multi-tenant.

Per ottenere un conteggio delle richieste M2M tra token ed endpoint per un periodo di tempo, puoi anche inviare eventi a Logs per l'analisi.AWS CloudTrail CloudWatch Interroga i tuoi CloudTrail eventi per verificare la presenza di `Token_POST` eventi con una concessione di credenziali client. La seguente query CloudWatch Insights restituisce questo conteggio. Per includere le richieste M2M effettuate con l'operazione `GetClientToken` API, utilizzate la metrica di utilizzo della frequenza delle ClientAuthentication richieste descritta in precedenza in questa sezione.

```
filter eventName = "Token_POST" and @message like '"grant_type":["client_credentials"]'
 | stats count(*)
```

Gestione dei costi

Le fatture di Amazon Cognito si basano sul numero di utenti, sull'utilizzo delle funzionalità e sul volume delle richieste. Di seguito sono riportati alcuni suggerimenti per gestire i costi in Amazon Cognito,

Non attivare gli utenti inattivi

Le operazioni tipiche che rendono attivo un utente sono l'accesso, la registrazione e la reimpostazione della password. Per un elenco più completo, consulta. Utenti mensili attivi Amazon Cognito non conta gli utenti inattivi ai fini della fattura. Evita qualsiasi operazione che renda attivo un utente. Invece dell'operazione AdminGetUserAPI, interroga gli utenti con l'ListUsersoperazione. Non eseguite test amministrativi ad alto volume sulle operazioni del pool di utenti con utenti inattivi.

Collega gli utenti federati

Gli utenti che accedono con un provider di identità SAML 2.0 o OpenID Connect (OIDC) hanno un costo maggiore rispetto agli utenti locali. Puoi collegare questi utenti a un profilo utente locale. Un utente collegato può accedere come utente locale con gli attributi e l'accesso forniti dal proprio utente federato. Gli utenti di SAML o OIDC IdPs che, nel corso di un mese, accedono solo con un account locale collegato vengono fatturati come utenti locali.

Gestisci le tariffe delle richieste

Se il tuo pool di utenti si avvicina al limite massimo della tua quota, potresti prendere in considerazione l'acquisto di capacità aggiuntiva per gestire il volume. Potresti riuscire a ridurre il volume delle richieste nella tua applicazione. Per ulteriori informazioni, consulta Ottimizzare i tassi di richiesta per i limiti di quota.

Richiedi un nuovo token solo quando ne hai bisogno

L'autorizzazione da macchina a macchina (M2M) con concessione di credenziali client può raggiungere un volume elevato di richieste di token. Ogni nuova richiesta di token ha un effetto sulla quota di richiesta e sull'entità della fattura. Per ottimizzare i costi, includi le impostazioni di scadenza dei token e la gestione dei token nella progettazione delle tue applicazioni.

- Memorizza i token di accesso alla cache in modo che, quando l'applicazione richiede un nuovo token, riceva una versione memorizzata nella cache di un token precedentemente emesso. Quando si implementa questo metodo, il proxy di memorizzazione nella cache funge da protezione contro le applicazioni che richiedono token di accesso senza essere consapevoli della scadenza dei token acquisiti in precedenza. I token di caching sono ideali per microservizi di breve durata come le funzioni Lambda e i contenitori Docker.
- Implementa meccanismi di gestione dei token nelle tue applicazioni che tengano conto della scadenza dei token. Non richiedete un nuovo token finché i token precedenti non stanno per scadere. Come best practice, aggiorna i token a circa il 75% della durata del token. Questa pratica massimizza la durata dei token garantendo al contempo la continuità dell'utente nell'applicazione.

  Valuta le esigenze di riservatezza e disponibilità di ciascuna applicazione e configura il client dell'app del pool di utenti per emettere token di accesso con un periodo di validità appropriato. La durata dei token personalizzati funziona meglio con API e server più longevi in grado di gestire in modo persistente la frequenza delle richieste di credenziali.

ListUsers, non AdminGetUser

Per interrogare gli attributi degli utenti nel tuo pool di utenti, utilizza l'operazione ListUsersAPI e i metodi SDK associati, quando possibile. AdminGetUsercontrassegna un utente come attivo per il mese e contribuisce agli utenti attivi mensili (MAU) utilizzati per calcolare la fattura per i pool di utenti.

Gestisci i piani di funzionalità

Quando scegli un piano di funzionalità in un pool di utenti, la tariffa di fatturazione si applica a tutte le MAU del pool di utenti. Se hai utenti che non necessitano di funzionalità fornite con un piano di funzionalità di livello superiore, separali in un altro pool di utenti.

Esportazione dei log dai pool di utenti di Amazon Cognito

Puoi configurare il tuo pool di utenti per inviare registri dettagliati di alcune attività aggiuntive a un altro Servizio AWS, ad esempio un gruppo di CloudWatch log. Questi registri hanno una granularità più precisa di quelli presenti e possono essere utili per risolvere i problemi del pool di utenti e analizzare le attività di accesso degli utenti con protezione dalle minacce. AWS CloudTrail Quando si desidera trasmettere in streaming i registri degli errori di notifica via SMS ed e-mail, il pool di utenti invia i log a livello di un gruppo di log. `ERROR` CloudWatch Quando desideri trasmettere in streaming i log delle attività di accesso degli utenti, il tuo pool di utenti invia i log a `INFO` livello di un gruppo di log, uno stream Amazon Data Firehose o un bucket Amazon S3. Puoi combinare entrambe le opzioni in un pool di utenti.

Argomenti

- Cose da sapere sull'esportazione dei log
- Esportazione degli errori di recapito di messaggi e-mail e SMS
- Esportazione dei log delle attività degli utenti per la protezione dalle minacce

Cose da sapere sull'esportazione dei log

Impatto sui costi

Amazon Data Firehose, Amazon S3 CloudWatch e Logs comportano costi per l'ingestione e il recupero dei dati. La configurazione AWS di registrazione potrebbe influire sulla fattura. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Vended Logs in Amazon CloudWatch Pricing.
- Prezzi di Amazon Data Firehose
- Prezzi di Amazon S3

User-activity le esportazioni di log contengono valutazioni di sicurezza e sono una funzione della protezione dalle minacce del pool di utenti. Amazon Cognito genera questi log solo quando la protezione dalle minacce è Full-functionattiva Audit-onlyo in modalità O e il tuo pool di utenti è incluso nel piano di funzionalità Plus.

I registri delle attività degli utenti sono a livello **`INFO`**

I registri delle attività degli utenti esportati sono solo a livello di `INFO` errore e forniscono informazioni per l'analisi statistica e di sicurezza dell'attività di autenticazione. I messaggi a livello di `ERROR` errore `WARNING` e, ad esempio, gli errori di limitazione, non sono inclusi nei log esportati.

Best-effort consegna

La consegna dei log da Amazon Cognito si basa sul miglior tentativo. Il volume di log fornito dal pool di utenti e le quote di servizio per CloudWatch Logs, Amazon S3 e Firehose possono influire sulla distribuzione dei log.

I log esterni esistenti non sono interessati

Queste opzioni di registrazione non sostituiscono o modificano le seguenti funzioni di registro dei pool di utenti.

1. CloudTrail registri delle attività di routine degli utenti come la registrazione e l'accesso.
2. Analisi dell'attività degli utenti su larga scala con metriche. CloudWatch

Separatamente, puoi anche trovare i log da Visualizzazione dei risultati di importazione del pool di utenti nella CloudWatch console e Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda in CloudWatch Logs. Amazon Cognito e Lambda archiviano questi log in gruppi di log diversi da quelli specificati per i log delle attività degli utenti.

Si applica solo ai pool di utenti

Non esistono funzionalità di esportazione dei log per i pool di identità.

Richiede le autorizzazioni dell'utente e un ruolo collegato al servizio

Il AWS responsabile che configura l'esportazione dei log deve disporre delle autorizzazioni per modificare le risorse di destinazione, come descritto negli argomenti seguenti. Amazon Cognito crea un ruolo collegato al servizio per tuo conto e si assume il ruolo di fornire i log alla risorsa di destinazione.

Per ulteriori informazioni sul modello di autorizzazione per l'invio di log da Amazon Cognito, consulta Enable logging from Servizi AWS nella CloudWatch Amazon Logs User Guide.

Il livello di registro è esclusivo del tipo di registro

Message-delivery i log sono del `userNotification` tipo e del livello di `ERROR` errore. I registri delle attività degli utenti con protezione avanzata sono del `userAuthEvents` tipo e del livello di errore. `INFO` È possibile combinare due membri di`LogConfigurations`, uno per `userNotification` to CloudWatch Logs e uno `userAuthEvents` per Firehose, Amazon S3 o Logs. CloudWatch

Non puoi inviare registri delle attività degli utenti a più destinazioni. Non puoi inviare i registri delle notifiche degli utenti a destinazioni diverse dai registri. CloudWatch

Diverse opzioni di configurazione

Puoi configurare i log delle notifiche degli utenti solo con l'API dei pool di utenti di Amazon Cognito o un SDK. AWS Puoi configurare i log delle attività degli utenti con sicurezza avanzata con l'API o nella console Amazon Cognito. Per impostarli entrambi, utilizza l'API come illustrato nella richiesta di esempio all'indirizzo. SetLogDeliveryConfiguration

È richiesta una configurazione aggiuntiva con politiche basate su risorse di grandi dimensioni

Per inviare registri a gruppi di log con una policy delle risorse di dimensione superiore a 5120 caratteri, configura un gruppo di log con un percorso che inizia con `/aws/vendedlogs`. Per ulteriori informazioni, vedere Abilitazione della registrazione da determinati servizi. AWS

Creazione automatica di una cartella in Amazon S3

Quando configuri l'esportazione dei log di protezione dalle minacce in un bucket Amazon S3, Amazon Cognito potrebbe creare una `AWSLogs` cartella nel tuo bucket. Tale cartella non viene creata in tutti i casi e la configurazione può avere esito positivo anche senza crearla.

Esportazione degli errori di recapito di messaggi e-mail e SMS

In caso di errori di recapito di messaggi e-mail e SMS, puoi inviare registri di notifica utente a livello di errore dal tuo pool di utenti. Quando attivi questa funzionalità, puoi scegliere il gruppo di log a cui Amazon Cognito deve inviare i log. La registrazione delle notifiche degli utenti è utile quando desideri conoscere lo stato dei messaggi e-mail e SMS inviati dal tuo pool di utenti con Amazon SNS e Amazon SES. Questa opzione di esportazione dei log, a differenza dell'esportazione delle attività degli utenti, non richiede il piano di funzionalità Plus.

Puoi configurare log di notifica dettagliati con l'API dei pool di utenti di Amazon Cognito in SetLogDeliveryConfigurationuna richiesta API. Puoi visualizzare la configurazione di registrazione di

un pool di utenti in una GetLogDeliveryConfigurationrichiesta API. Di seguito è riportato un esempio di corpo della richiesta.

```
{
   "LogConfigurations": [
      {
         "CloudWatchLogsConfiguration": {
            "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-group:example-user-
pool-exported"
         },
         "EventSource": "userNotification",
         "LogLevel": "ERROR"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

È necessario autorizzare queste richieste con AWS credenziali che dispongono delle seguenti autorizzazioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
            {
            "Sid": "ManageUserPoolLogs",
            "Action": [
                "cognito-idp:SetLogDeliveryConfiguration",
                "cognito-idp:GetLogDeliveryConfiguration"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "CognitoLog",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
```

```
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "CognitoLoggingCWL",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Di seguito è illustrato un evento di esempio da un pool di utenti. Questo schema di log è soggetto a modifiche. Alcuni campi potrebbero essere registrati con valori null.

```
{
    "eventTimestamp": "1687297330677",
    "eventSource": "USER_NOTIFICATION",
    "logLevel": "ERROR",
    "message": {
        "details": "String"
    },
    "logSourceId": {
        "userPoolId": "String"
    }
}
```

Esportazione dei log delle attività degli utenti per la protezione dalle minacce

I pool di utenti con il piano di funzionalità Plus e la protezione dalle minacce registrano gli eventi delle attività degli utenti: i dettagli e la valutazione della sicurezza delle operazioni di accesso,

disconnessione e altre operazioni di autenticazione degli utenti con il pool di utenti. Potresti voler esaminare i registri delle attività degli utenti nel tuo sistema di gestione dei registri o creare un archivio. Puoi esportare questi dati in un gruppo di log Amazon CloudWatch Logs, uno stream Amazon Data Firehose o un bucket Amazon Simple Storage Service (Amazon S3). Da lì, puoi importare questi dati in altri sistemi che analizzano, normalizzano o elaborano in altro modo i dati in modo da adattarli ai tuoi processi operativi. Per esportare dati di questo tipo, è necessario che il pool di utenti utilizzi il piano di funzionalità Plus e che la protezione dalle minacce sia attiva nel pool di utenti.

Con le informazioni contenute in questi registri delle attività degli utenti, è possibile visualizzare un profilo delle attività di accesso e gestione dell'account degli utenti. Per impostazione predefinita, Amazon Cognito acquisisce questi eventi su uno storage basato nel tuo pool di utenti. L'esempio seguente è un esempio di evento per un utente che ha effettuato l'accesso ed è stato valutato come privo di fattori di rischio. È possibile recuperare queste informazioni con l'operazione `AdminListUserAuthEvents` API. Di seguito è riportato un esempio di output:


```
{
    "AuthEvents": [
        {
            "EventId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "EventType": "SignIn",
            "CreationDate": "2024-06-27T10:49:59.139000-07:00",
            "EventResponse": "Pass",
            "EventRisk": {
                "RiskDecision": "NoRisk",
                "CompromisedCredentialsDetected": false
            },
            "ChallengeResponses": [
                {
                    "ChallengeName": "Password",
                    "ChallengeResponse": "Success"
                }
            ],
            "EventContextData": {
                "IpAddress": "192.0.2.1",
                "DeviceName": "Chrome 126, Windows 10",
                "Timezone": "-07:00",
                "City": "null",
                "Country": "United States"
            }
        }
```

```
    ],
    "NextToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222#2024-06-27T17:49:59.139Z"
}
```

Puoi attivare l'esportazione dei log per l'attività degli utenti nella console Amazon Cognito o con il funzionamento dell'SetLogDeliveryConfigurationAPI.

Console di gestione AWS

1. Se non ne hai già uno che desideri utilizzare, crea un bucket S3, un flusso Firehose o un gruppo di log. CloudWatch
2. Accedi alla console Amazon Cognito.
3. Scegli User Pools (Pool di utenti).
4. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
5. Scegli la scheda Sicurezza avanzata. Individua Esporta i registri delle attività degli utenti e scegli Modifica
6. In Stato di registrazione, seleziona la casella di controllo accanto a Attiva l'esportazione del registro delle attività degli utenti.
7. In Logging destination, scegli il tipo di log con Servizio AWS cui vuoi gestire i log: gruppo di CloudWatch log, flusso Amazon Data Firehose o bucket S3.
8. La selezione inserisce nel selettore di risorse il tipo di risorsa corrispondente. Seleziona un gruppo di log, uno stream o un bucket dall'elenco. Puoi anche selezionare il pulsante Crea Console di gestione AWS per accedere al servizio selezionato e creare una nuova risorsa.
9. Seleziona Salva modifiche.

API

Scegli un tipo di destinazione per i registri delle attività degli utenti.

Di seguito è riportato un esempio di corpo della `SetLogDeliveryConfiguration` richiesta che imposta un flusso Firehose come destinazione del registro.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "FirehoseConfiguration": {
```

```
            "StreamArn": "arn:aws:firehose:us-west-2:123456789012:deliverystream/
example-user-pool-activity-exported"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Di seguito è riportato un esempio di corpo di `SetLogDeliveryConfiguration` richiesta che imposta un bucket Amazon S3 come destinazione del log.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "S3Configuration": {
            "BucketArn": "arn:aws:s3:::amzn-s3-demo-logging-bucket"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Di seguito è riportato un esempio di corpo della `SetLogDeliveryConfiguration` richiesta che imposta un gruppo di CloudWatch log come destinazione del log.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "CloudWatchLogsConfiguration": {
            "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-group:DOC-
EXAMPLE-LOG-GROUP"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

L'utente che configura la consegna dei log deve essere un amministratore del pool di utenti e disporre delle seguenti autorizzazioni aggiuntive:

Amazon S3

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ManageUserPoolLogs",
            "Action": [
                "cognito-idp:SetLogDeliveryConfiguration",
                "cognito-idp:GetLogDeliveryConfiguration"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "ManageLogsS3",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudWatch Logs

JSON

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

```
    "Statement": [
        {
            "Sid": "ManageUserPoolLogs",
            "Action": [
                "cognito-idp:SetLogDeliveryConfiguration",
                "cognito-idp:GetLogDeliveryConfiguration"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "ManageLogsCWL",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Amazon Data Firehose

JSON

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

```
                "cognito-idp:SetLogDeliveryConfiguration",
                "cognito-idp:GetLogDeliveryConfiguration"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "ManageUserPoolLogsFirehose",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "iam:CreateServiceLinkedRole",
                "firehose:TagDeliveryStream"
            ],
            "Resource": "*"
        }
    ]
}
```

Di seguito è illustrato un evento di esempio da un pool di utenti. Questo schema di log è soggetto a modifiche. Alcuni campi potrebbero essere registrati con valori null.

```
{
    "eventTimestamp": "1687297330677",
    "eventSource": "USER_ACTIVITY",
    "logLevel": "INFO",
    "message": {
        "version": "1",
        "eventId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventType": "SignUp",
        "userSub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "userName": "test-user",
        "userPoolId": "us-west-2_EXAMPLE",
        "clientId": "1example23456789",
        "creationDate": "Wed Jul 17 17:25:55 UTC 2024",
        "eventResponse": "InProgress",
        "riskLevel": "",
        "riskDecision": "PASS",
        "challenges": [],
        "deviceName": "Other, Other",
```

```
        "ipAddress": "192.0.2.1",
        "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "idpName": "",
        "compromisedCredentialDetected": "false",
        "city": "Seattle",
        "country": "United States",
        "eventFeedbackValue": "",
        "eventFeedbackDate": "",
        "eventFeedbackProvider": "",
        "hasContextData": "true"
    },
    "logSourceId": {
        "userPoolId": "us-west-2_EXAMPLE"
    }
}
```


Monitoraggio delle quote e dell'utilizzo in CloudWatch e Service Quotas

Puoi monitorare i pool di utenti di Amazon Cognito utilizzando Amazon CloudWatch o Service Quotas. Puoi anche monitorare l'utilizzo dei pool di identità in Service Quotas. CloudWatch raccoglie dati grezzi e li elabora in metriche leggibili e quasi in tempo reale. In CloudWatch, puoi impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando tali soglie vengono raggiunte. Per creare un CloudWatch allarme per una quota di servizio, vedi Creare un avviso. CloudWatch I parametri di Amazon Cognito sono disponibili a intervalli di cinque minuti. Per ulteriori informazioni sui periodi di conservazione in CloudWatch, visita la pagina delle CloudWatch domande frequenti di Amazon.

Puoi utilizzare Service Quotas per visualizzare e gestire l'utilizzo delle quote dei pool di utenti e dei pool di identità di Amazon Cognito. La console Service Quotas dispone di tre funzionalità: visualizzazione delle quote di servizio, richiesta di un aumento delle quote di servizio e visualizzazione dell'utilizzo corrente. Puoi utilizzare la prima funzionalità per visualizzare le quote e verificare se la quota è regolabile. Puoi utilizzare la seconda caratteristica per richiedere un aumento delle Service Quotas Infine, puoi utilizzare l'ultima funzionalità per visualizzare l'utilizzo delle quote. Questa funzione è disponibile solo dopo che l'account è rimasto attivo per un po' di tempo. Per ulteriori informazioni sulla visualizzazione delle quote nella console Service Quotas, consulta Visualizzazione di Service Quotas.

> **Note**
>
> I parametri di Amazon Cognito sono disponibili a intervalli di cinque minuti. Per ulteriori informazioni sui periodi di conservazione in CloudWatch, visita la pagina delle CloudWatch domande frequenti di Amazon.

Se hai effettuato l'accesso a un Account AWS account configurato come account di monitoraggio in modalità osservabilità CloudWatch tra account diversi, puoi utilizzare quell'account di monitoraggio per visualizzare le quote di servizio e impostare allarmi per le metriche negli account di origine collegati a quell'account di monitoraggio. Per maggiori informazioni, consulta la sezione Osservabilità su più account di CloudWatch .

Argomenti

- Metriche del pool di utenti in CloudWatch
- Metriche in Service Quotas

Metriche del pool di utenti in CloudWatch

I pool di utenti segnalano le statistiche sull'attività degli utenti alle metriche as. CloudWatch Da CloudWatch, puoi analizzare il volume delle attività di autenticazione e l'utilizzo delle quote nei tuoi pool di utenti. Con le informazioni contenute in queste metriche, puoi impostare allarmi per eventi importanti e modificare la configurazione del pool di utenti in base alle esigenze. Laddove la registrazione delle attività degli utenti contiene registrazioni dettagliate delle attività degli utenti nei pool di utenti, le CloudWatch metriche contengono statistiche aggregate e indicatori di performance.

Nella tabella seguente sono elencati i parametri e le dimensioni disponibili per i bacini d'utenza di Amazon Cognito. Amazon Cognito pubblica i parametri nei namespace e. `AWS/Cognito` `AWS/Usage` Per ulteriori informazioni, consulta Namespaces in Amazon CloudWatch User Guide.

Per ulteriori informazioni sul monitoraggio delle quote e dell'utilizzo, consulta e. Monitoraggio dell'uso delle quote Tieni traccia degli utenti attivi mensili (MAU)

> **Note**
>
> I parametri che non hanno ricevuto nuovi punti di dati nelle ultime due settimane non vengono visualizzati nella console. Inoltre, non vengono visualizzati quando si immette il nome del parametro o i nomi delle dimensioni nella casella di ricerca nella scheda Tutti i parametri nella

console. Inoltre, non vengono restituiti nei risultati di un comando list-metrics. Il modo migliore per recuperare queste metriche è con i `get-metric-statistics` comandi `get-metric-data` o nella AWS CLI.

Metrica	Description	Namespace
`SignUpSuccesses`	Fornisce il numero totale di richieste di registrazione utente riuscite effettuate al bacino d'utenza di Amazon Cognito. Una richiesta di registrazione utente riuscita produce un valore pari a 1, mentre una richiesta non riuscita produce un valore pari a 0. Anche una richiesta limitata viene considerata una richiesta non riuscita e quindi una richiesta limitata produce anche un conteggio pari a 0. Per trovare la percentuale di richieste di registrazione utente riuscite, utilizza la statistica `Average` di questo parametro. Per calcolare il numero totale di richieste di registrazione utente, utilizza la statistica `Sample Count` su questo parametro. Per calcolare il numero totale di richieste di registrazione utente riuscite, utilizza la statistica `Sum` su questo parametro. Per contare il numero totale di richieste	`AWS/Cognito`

Metrica	Description	Namespace
	di registrazione utente non riuscite, usa l' CloudWatch `Math`espressione e sottrai la statistica dalla statistica. `Sum` `Sample Count` Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client del bacino d'utenza. Nel caso in cui la registrazione dell'utente venga eseguita da un amministratore, il parametro viene pubblicato con il client del bacino d'utenza come `Admin`. Si noti che questo parametro non viene emesso per i casi di importazione di utenti e migrazione di utenti. Dimensione del parametro: `UserPool`, `UserPoolClient` Unità: numero	

Metrica	Description	Namespace
`SignUpThrottles`	Fornisce il numero totale di richieste di registrazione utente limitate effettuate al bacino d'utenza di Amazon Cognito. Ogni volta che una richiesta di registrazione utente viene sottoposta a throttling, viene pubblicato un conteggio di 1. Per calcolare il numero totale di richieste di registrazione utente con throttling, utilizza la statistica `Sum` per questo parametro. Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client. Nel caso in cui la richiesta che è stata sottoposta a limitazione è stata effettuata da un amministratore, il parametro viene pubblicato con il client del bacino d'utenza come `Admin`. Dimensione del parametro: `UserPool`, `UserPoolClient` Unità: numero	`AWS/Cognito`

Metrica	Description	Namespace
`SignInSuccesses`	Fornisce il numero totale di richieste di autenticazione utente riuscite effettuate al bacino d'utenza di Amazon Cognito. Un'autenticazione utente viene considerata riuscita quando il token di autenticazione viene inviato all'utente. Un'autenticazione riuscita produce un valore pari a 1, mentre una richiesta non riuscita produce un valore pari a 0. Anche una richiesta limitata viene considerata una richiesta non riuscita e quindi una richiesta limitata produce anche un conteggio pari a 0. Per trovare la percentuale di richieste di autenticazione utente riuscite, utilizza la statistica `Average` su questo parametro. Per calcolare il numero totale di richieste di autenticazione utente, utilizza la statistica `Sample Count` su questo parametro . Per calcolare il numero totale di richieste di autentica zione utente riuscite, utilizza la statistica `Sum` su questo parametro. Per contare il numero totale di richieste di autenticazione utente non riuscite, usa l' CloudWatch	`AWS/Cognito`

Metrica	Description	Namespace
	`Mathespressione` e sottrai la statistica dalla statistica. `Sum` `Sample Count` Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client. Nel caso in cui con una richiesta venga fornito un client del bacino d'utenza non valido, il valore corrispondente del client del bacino d'utenza nel parametro conterrà un valore `Invalid` fisso anziché il valore non valido effettivo inviato nella richiesta. Si noti che le richieste di aggiornamento del token di Amazon Cognito non sono incluse in questo parametro. Esiste un parametro separato per fornire le statistiche dei token `Refresh`. Dimensione del parametro : `UserPool`, `UserPoolClient` Unità: numero	

Metrica	Description	Namespace
`SignInThrottles`	Fornisce il numero totale di richieste di autenticazione utente limitate effettuate al bacino d'utenza di Amazon Cognito. Ogni volta che una richiesta di autenticazione viene sottoposta a throttling, viene pubblicato un conteggio di 1. Per calcolare il numero totale di richieste di autenticazione utente con throttling, utilizza la statistica `Sum` per questo parametro. Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client. Nel caso in cui con una richiesta venga fornito un client del bacino d'utenza non valido, il valore corrispondente del client del bacino d'utenza nel parametro conterrà un valore `Invalid` fisso anziché il valore non valido effettivo inviato nella richiesta. Le richieste di aggiornam ento del token Amazon Cognito non sono incluse in questo parametro. Esiste un parametro separato per	`AWS/Cognito`

Metrica	Description	Namespace
	fornire le statistiche dei token `Refresh`. Dimensione del parametro : `UserPool`, `UserPoolClient` Unità: numero	

Metrica	Description	Namespace
`TokenRefreshSuccesses`	Fornisce il numero totale di richieste riuscite per aggiornare un token Amazon Cognito che sono state effettuate al bacino d'utenza di Amazon Cognito. Una richiesta riuscita di aggiornamento del token Amazon Cognito produce un valore pari a 1, mentre una richiesta non riuscita produce un valore pari a 0. Anche una richiesta limitata viene considerata una richiesta non riuscita e quindi una richiesta limitata produce anche un conteggio pari a 0. Per trovare la percentuale di richieste riuscite di aggiornamento di un token Amazon Cognito, utilizza la statistica `Average` su questo parametro. Per calcolare il numero totale di richieste di aggiornamento di un token Amazon Cognito, utilizza la statistica `Sample Count` su questo parametro. Per calcolare il numero totale di richieste riuscite di aggiornamento di un token Amazon Cognito, utilizza la statistica `Sum` su questo parametro. Per contare il numero totale di richieste non riuscite di aggiornamento	`AWS/Cognito`

Metrica	Description	Namespace
	di un token Amazon Cognito, usa l'espressione e CloudWatch `Math` sottrai la statistica `Sum` dalla statistica. `SampleCount` Questo parametro viene pubblicato per ogni client del bacino d'utenza. Se un client del bacino d'utenza non valido è riportato in una richiesta, il valore del client conterrà un valore fisso pari a `Invalid`. Dimensione del parametro: `UserPool`, `UserPoolClient` Unità: numero	

Metrica	Description	Namespace
`TokenRefreshThrottles`	Fornisce il numero totale di richieste limitate per aggiornare un token Amazon Cognito che sono state effettuate al bacino d'utenza di Amazon Cognito. Ogni volta che una richiesta di aggiornamento del token Amazon Cognito viene sottoposta a limitazione, viene pubblicato un conteggio di 1 Per calcolare il numero totale di richieste con limitazione di aggiornamento di un token Amazon Cognito, utilizza la statistica `Sum` per questo parametro. Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client. Nel caso in cui con una richiesta venga fornito un client del bacino d'utenza non valido, il valore corrispondente del client del bacino d'utenza nel parametro conterrà un valore `Invalid` fisso anziché il valore non valido effettivo inviato nella richiesta. Dimensione del parametro: `UserPool`, `UserPoolClient`	`AWS/Cognito`

Metrica	Description	Namespace
	Unità: numero	

Metrica	Description	Namespace
`FederationSuccesses`	Fornisce il numero totale di richieste di federazione delle identità riuscite al bacino d'utenza di Amazon Cognito. Una federazione delle identità è considerata riuscita quando Amazon Cognito emette token di autenticazione per l'utente. Una richiesta di federazione delle identità riuscita produce un valore pari a 1, mentre una richiesta non riuscita produce un valore pari a 0. Le richieste limitate e le richieste che generano un codice di autorizzazione senza token producono un valore pari a 0. Per trovare la percentuale di richieste di federazione delle identità riuscite, utilizza la statistica `Average` di questo parametro. Per calcolare il numero totale di richieste di federazione delle identità, utilizza la statistica `Sample Count` di questo parametro. Per calcolare il numero totale di richieste di federazione delle identità riuscite, utilizza la statistica `Sum` di questo parametro. Per contare il numero totale di richieste di federazione delle identità non riuscite, usa l' CloudWatch	`AWS/Cognito`

Metrica	Description	Namespace
	`Mathespressione` e sottrai la statistica dalla statistica. `Sum` `Sample Count` Dimensione del parametro : `UserPool`, `UserPoolClient` , `IdentityProvider` Unità: numero	
`FederationThrottles`	Fornisce il numero totale di richieste di federazione delle identità limitate al bacino d'utenza di Amazon Cognito. Ogni volta che una richiesta di federazione delle identità viene limitata, viene pubblicato un conteggio di 1. Per calcolare il numero totale di richieste di federazione delle identità con throttling, utilizza la statistica `Sum` per questo parametro. Dimensione del parametro : `UserPool`, `UserPoolClient` , `IdentityProvider` Unità: numero	`AWS/Cognito`

Metrica	Description	Namespace
`CallCount`	Fornisce il numero totale di chiamate effettuate dai clienti in relazione a una categoria . Questo parametro include tutte le chiamate, ad esempio chiamate limitate, chiamate non riuscite e chiamate riuscite. La quota di categoria viene applicata per ogni AWS account in tutti i pool di utenti di un account e di una regione. Per contare il numero totale di chiamate in una categoria puoi utilizzare la statistica `Sum` per questo parametro. Dimensione parametro: Servizio, Tipo, Risorsa, Classe Unità: numero	`AWS/Usage`

Metrica	Description	Namespace
`ThrottleCount`	Fornisce il numero totale di chiamate limitate in relazione a una categoria. Questo parametro viene pubblicato a livello di account. Per contare il numero totale di chiamate in una categoria puoi utilizzare la statistica `Sum` per questo parametro. Dimensione parametro: Servizio, Tipo, Risorsa, Classe Unità: numero	`AWS/Usage`

Visualizzazione delle metriche di protezione dalle minacce

Le metriche pubblicate dal pool di utenti contengono informazioni statistiche sull'effetto delle impostazioni di protezione dalle minacce sull'attività di autenticazione degli utenti. Potresti voler sapere quanti utenti stanno tentando di accedere con credenziali compromesse. Puoi anche scoprire quale percentuale di attività di accesso è stata valutata come comportante un certo livello di rischio. Amazon Cognito pubblica i parametri relativi alle funzionalità di protezione dalle minacce per il tuo account in Amazon. CloudWatch Amazon Cognito raggruppa i parametri di protezione dalle minacce per livello di rischio e anche per livello di richiesta.

Per contestualizzare l'analisi del rischio, puoi visualizzare le informazioni sui tentativi di accesso dei singoli utenti, nel tuo pool di utenti o in una fonte di dati esportata.

Per visualizzare le metriche nella console CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri).
3. Scegli Amazon Cognito.
4. Scegli un gruppo di parametri aggregati, ad esempio Classificazione per rischio.

5. Nella scheda All metrics (Tutti i parametri) sono visualizzati tutti i parametri per quella scelta. È possibile effettuare le seguenti operazioni:

   - Per ordinare la tabella, utilizza l'intestazione della colonna.
   - Per creare il grafico di un parametro, seleziona la casella di spunta accanto al parametro. Per selezionare tutte le metriche, seleziona la casella di controllo nella riga del titolo della tabella.
   - Per filtrare in base a una risorsa, scegli l'ID della risorsa e quindi Add to search (Aggiungi alla ricerca).
   - Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).

Metrica	Description	Dimensioni parametro	Namespace
CompromisedCredentialRisk	Richieste in cui Amazon Cognito ha rilevato credenziali compromesse.	Operazione: il tipo di operazione `PasswordChange` , `SignIn` o `SignUp` sono le uniche dimensioni. UserPoolId: l'identificatore del pool di utenti. RiskLevel: alto (impostazione predefinita), medio o basso.	`AWS/Cognito`
AccountTakeoverRisk	Richieste in cui Amazon Cognito ha rilevato il rischio di furto dell'account.	Operazione: il tipo di operazione `PasswordChange` , `SignIn` o `SignUp` sono le uniche dimensioni.	`AWS/Cognito`

Metrica	Description	Dimensioni parametro	Namespace
		UserPoolId: l'identificatore del pool di utenti. RiskLevel: alto, medio o basso.	
OverrideBlock	Richieste che Amazon Cognito ha bloccato a causa della configurazione fornita dallo sviluppatore.	Operazione: il tipo di operazione `PasswordChange`, `SignIn` o `SignUp` sono le uniche dimensioni. UserPoolId: l'identificatore del pool di utenti. RiskLevel: alto, medio o basso.	`AWS/Cognito`
Rischio	Richieste contrassegnate da Amazon Cognito come rischiose.	Operazione: tipo di operazione come `PasswordChange`, `SignIn` o `SignUp`. UserPoolId: l'identificatore del pool di utenti.	`AWS/Cognito`

Metrica	Description	Dimensioni parametro	Namespace
NoRisk	Richieste in cui Amazon Cognito non ha identificato alcun rischio.	Operazione: tipo di operazione come `PasswordChange` , `SignIn` o `SignUp`. UserPoolId: l'identif icatore del pool di utenti.	`AWS/Cognito`

Amazon Cognito ti offre due gruppi predefiniti di metriche per un'analisi pronta. CloudWatch Per classificazione in base al rischio si intende la granularità del livello di rischio per le richieste che Amazon Cognito identifica come rischiose. La classificazione in base alla richiesta riflette le metriche aggregate per livello di richiesta.

Gruppo di parametri aggregati	Description
Classificazione per rischio	Richieste identificate da Amazon Cognito come rischiose.
Classificazione per richiesta	Parametri aggregati per richiesta.

Dimensioni per i bacini d'utenza di Amazon Cognito

Le seguenti dimensioni vengono utilizzate per perfezionare i parametri di utilizzo pubblicati da Amazon Cognito. Le dimensioni si applicano solo ai parametri `CallCount` e `ThrottleCount` .

Dimensione	Descrizione
Servizio	Il nome del AWS servizio che contiene la risorsa. Per i parametri di utilizzo di Amazon Cognito, il valore per questa dimensione è `Cognito user pool`.

Dimensione	Descrizione
Tipo	Il tipo di entità che viene segnalato. Attualmente, l'unico valore valido per i parametri di utilizzo di Amazon Cognito è API.
Risorsa	Il tipo di risorsa in esecuzione. L'unico valore valido è il nome della categoria.
Classe	La classe della risorsa monitorata. Amazon Cognito non utilizza la dimensione della classe.

Usa la CloudWatch console per tenere traccia delle metriche

Puoi tracciare e raccogliere i parametri dei pool di utenti di Amazon Cognito utilizzando. CloudWatch La CloudWatch dashboard mostra le metriche relative a ogni AWS servizio che utilizzi. Puoi usarlo CloudWatch per creare allarmi metrici. Gli allarmi possono essere configurati per inviare notifiche all'utente o apportare una modifica a una risorsa specifica che si sta monitorando. Per visualizzare le metriche delle quote di servizio in CloudWatch, completa i seguenti passaggi.

1. Apri la CloudWatch console.
2. Nel pannello di navigazione, seleziona Metrics (Parametri).
3. In All metrics (Tutti i parametri) seleziona un parametro e una dimensione.
4. Seleziona la casella di controllo accanto a una metrica. Le metriche vengono visualizzate nel grafico.

> **Note**
>
> I parametri che non hanno ricevuto nuovi punti di dati nelle ultime due settimane non vengono visualizzati nella console. Inoltre, non vengono visualizzati quando digiti il nome del parametro o i nomi delle dimensioni nella casella di ricerca della scheda Tutti i parametri della console e non vengono restituiti nei risultati del comando list-metrics. Il modo migliore per recuperare questi parametri è con i comandi `get-metric-data` o `get-metric-statistics` in AWS CLI.

Crea un CloudWatch allarme per una quota

Amazon Cognito fornisce metriche di CloudWatch utilizzo che corrispondono alle quote di AWS servizio e alle API. `CallCount` `ThrottleCount` Per ulteriori informazioni sul monitoraggio dell'utilizzo in, consulta. CloudWatch Monitoraggio dell'uso delle quote

Nella console Service Quotas, puoi configurare gli allarmi che avvisano quando il tuo utilizzo si avvicina a una quota di servizio. Per informazioni su come configurare un CloudWatch allarme utilizzando la console Service Quotas, vedi Service Quotas and alarms. CloudWatch

Metriche in Service Quotas

Puoi visualizzare e gestire le quote dei pool di utenti e dei pool di identità di Amazon Cognito da una posizione centrale con Service Quotas. Puoi utilizzare la console Service Quotas per visualizzare i dettagli su una quota specifica, monitorare l'utilizzo della quota e richiedere un aumento della quota. Per alcuni tipi di quote, è possibile creare un CloudWatch allarme per monitorare l'utilizzo delle quote. Per ulteriori informazioni sulle metriche di Amazon Cognito che puoi monitorare, consultare Monitoraggio dell'uso delle quote.

Per visualizzare l'utilizzo di Service Quotas dei pool di utenti e dei pool di identità di Amazon Cognito, completa i seguenti passaggi.

1. Apri la console Service Quotas.
2. Nel pannello di navigazione, scegli Servizi AWS (servizi AWS).
3. Dall'elenco dei Servizi AWS , cerca e scegli i pool di utenti di Amazon Cognito o le identità federate di Amazon Cognito. Viene visualizzata la pagina Quote di servizio.
4. Seleziona una quota che supporti il CloudWatch monitoraggio. Ad esempio, scegli `Rate of UserAuthentication requests` nei pool di utenti di Amazon Cognito.
5. Scorri verso il basso fino a Monitoring (Monitoraggio). Questa sezione viene visualizzata solo per le quote che supportano il CloudWatch monitoraggio.
6. In Monitoring (Monitoraggio) puoi visualizzare l'utilizzo corrente delle quote di servizio nel grafico.
7. In Monitoring (Monitoraggio) seleziona un'ora, tre ore, dodici ore, un giorno, tre giorni o una settimana.
8. Seleziona una qualsiasi area all'interno del grafico per visualizzare la percentuale di utilizzo delle quote di servizio. Da qui, puoi aggiungere il grafico alla dashboard o utilizzare il menu delle azioni per selezionare Visualizza nelle metriche, che ti porta alle metriche correlate nella console. CloudWatch

Accesso ad Amazon Cognito AWS CloudTrail

Amazon Cognito è integrato con AWS CloudTrail un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in Amazon Cognito. CloudTrail acquisisce un sottoinsieme di chiamate API per Amazon Cognito come eventi, incluse le chiamate dalla console Amazon Cognito e le chiamate di codice alle operazioni dell'API Amazon Cognito. Se crei un trail, puoi scegliere di distribuire CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Amazon Cognito. 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 effettuata ad Amazon Cognito, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e attivarlo, consulta la Guida per l'AWS CloudTrail utente.

Puoi anche creare CloudWatch allarmi Amazon per CloudTrail eventi specifici. Ad esempio, puoi impostare CloudWatch per attivare un allarme se una configurazione di un pool di identità è stato modificata. Per ulteriori informazioni, consulta Creazione di CloudWatch allarmi per CloudTrail eventi: esempi.

Argomenti

- Informazioni inviate da Amazon Cognito a CloudTrail
- Analisi degli eventi di Amazon CloudTrail Cognito con Amazon CloudWatch Logs Insights
- Esempi di eventi Amazon Cognito

Informazioni inviate da Amazon Cognito a CloudTrail

CloudTrail si attiva quando crei il tuo Account AWS. Quando si verifica un'attività di evento supportata in Amazon Cognito, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon Cognito, crea un percorso. Un CloudTrail trail invia i 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 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:

- Panoramica della creazione di un percorso
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche di Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

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 le credenziali dell'utente IAM o root.
- 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.

Dati riservati in AWS CloudTrail

Poiché i pool di utenti e i pool di identità elaborano i dati degli utenti, Amazon Cognito oscura alcuni campi privati dei tuoi CloudTrail eventi con il valore. `HIDDEN_DUE_TO_SECURITY_REASONS` Per esempi di campi che non vengono compilati da Amazon Cognito negli eventi, consulta Esempi di eventi Amazon Cognito. Amazon Cognito oscura solo alcuni campi che contengono di solito informazioni utente, come password e token. Amazon Cognito non rileva o maschera automaticamente le informazioni di identificazione personale inserite in campi non privati nelle richieste API.

Eventi del pool di utenti

Amazon Cognito supporta la registrazione di tutte le azioni elencate nella pagina delle azioni del pool di utenti come eventi nei CloudTrail file di registro. Amazon Cognito registra gli eventi del pool di utenti CloudTrail come eventi di gestione.

Il `eventType` campo in una CloudTrail voce relativa ai pool di utenti di Amazon Cognito indica se l'app ha effettuato la richiesta all'API dei pool di utenti di Amazon Cognito o a un endpoint che fornisce risorse per OpenID Connect, SAML 2.0 o pagine di accesso gestite. Le richieste

API hanno un `eventType` di `AwsApiCall` e le richieste degli endpoint hanno un `eventType` di `AwsServiceEvent`.

Amazon Cognito registra le seguenti richieste nei tuoi servizi di accesso gestito come eventi. CloudTrail

Hosted UI (classic) events

Eventi dell'interfaccia utente (classici) ospitati in CloudTrail

Operation	Description
`Login_GET` , `CognitoAuthentication`	Un utente visualizza o invia le credenziali al tuo Endpoint Login.
`OAuth2_Authorize_GET` , `Beta_Authorize_GET`	Un utente visualizza il tuo Endpoint Authorize.
`OAuth2Response_GET` , `OAuth2Response_POST`	Un utente invia un token IdP al tuo endpoint `/oauth2/idpresponse` .
`SAML2Response_POST` , `Beta_SAML2Response_POST`	Un utente invia un'asserzione SAML IdP al tuo endpoint `/saml2/idpresponse` .
`Login_OIDC_SAML_POST`	Un utente specifica un nome utente nel tuo Endpoint Login e lo abbina con un identificatore IdP.
`Token_POST` , `Beta_Token_POST`	Un utente invia un codice di autorizzazione al tuo Endpoint Token.
`Signup_GET` , `Signup_POST`	Un utente invia le informazioni di registrazione al tuo endpoint `/signup`.
`Confirm_GET` , `Confirm_POST`	Un utente invia un codice di conferma nell'interfaccia utente ospitata.
`ResendCode_POST`	Un utente invia una richiesta per inviare nuovamente un codice di conferma nell'interfaccia utente ospitata.

Operation	Description
`ForgotPassword_GET` , `ForgotPassword_POST`	Un utente invia una richiesta per reimpostare la relativa password sul tuo endpoint `/forgotPassword` .
`ConfirmForgotPassword_GET` , `ConfirmForgotPassword_POST`	Un utente invia un codice al tuo endpoint `/confirmForgotPassword` che conferma la sua richiesta `ForgotPassword` .
`ResetPassword_GET` , `ResetPassword_POST`	Un utente invia una nuova password nell'interfaccia utente ospitata.
`Mfa_GET`, `Mfa_POST`	Un utente invia un codice di autenticazione a più fattori (MFA) nell'interfaccia utente ospitata.
`MfaOption_GET` , `MfaOption_POST`	Un utente sceglie il metodo preferito per l'MFA nell'interfaccia utente ospitata.
`MfaRegister_GET` , `MfaRegister_POST`	Un utente invia un codice di autenticazione a più fattori (MFA) nell'interfaccia utente ospitata durante la registrazione dell'MFA.
`Logout`	Un utente si disconnette dal tuo endpoint `/logout`.
`SAML2Logout_POST`	Un utente si disconnette dal tuo endpoint `/saml2/logout` .
`Error_GET`	Un utente visualizza una pagina di errore nell'interfaccia utente ospitata.
`UserInfo_GET` , `UserInfo_POST`	Un utente o un IdP scambia informazioni con il tuo Endpoint UserInfo.
`Confirm_With_Link_GET`	Un utente invia una conferma in base a un link inviato da Amazon Cognito in un messaggio e-mail.

Operation	Description
`Event_Feedback_GET`	Un utente invia un feedback ad Amazon Cognito in merito a un evento di protezione dalle minacce.

Managed login events

Eventi di accesso gestiti in CloudTrail

Operation	Description
`login_POST`	Un utente invia le credenziali al tuo. Endpoint Login
`login_continue_POST`	Un utente che ha già effettuato l'accesso una volta sceglie di accedere nuovamente.
`forgotPassword_POST`	Un utente reimposta la propria password.
`selectChallenge_POST`	Un utente risponde a una richiesta di autenticazione dopo aver inviato il nome utente o le credenziali.
`confirmUser_GET`	Un utente apre il link in un messaggio e-mail di conferma o di verifica.
`mfa_back_POST`	Un utente sceglie il pulsante Indietro dopo una richiesta MFA.
`mfa_options_POST`	Un utente seleziona un'opzione MFA.
`mfa_phone_register_POST`	Un utente invia un numero di telefono da registrare come fattore MFA. Questa operazione fa sì che Amazon Cognito invii un codice MFA al loro numero di telefono.
`mfa_phone_verify_POST`	Un utente invia un codice MFA inviato al proprio numero di telefono.

Operation	Description
`mfa_phone_resendCode_POST`	Un utente invia una richiesta per inviare nuovamente un codice MFA al proprio numero di telefono.
`mfa_totp_POST`	Un utente invia un codice MFA TOTP.
`signup_POST`	Un utente invia informazioni alla tua `/signup` pagina di accesso gestita.
`signup_confirm_POST`	Un utente invia un codice di conferma tramite e-mail o SMS.
`verifyCode_POST`	Un utente invia una password monouso (OTP) per l'autenticazione senza password.
`passkeys_add_POST`	Un utente invia una richiesta per registrare una nuova credenziale di accesso.
`passkeys_add_GET`	Un utente accede alla pagina in cui può registrare una passkey.
`login_passkey_POST`	Un utente accede con una passkey.

> **Note**
>
> Amazon Cognito registra`UserSub`, ma non `UserName` nei CloudTrail log, le richieste specifiche di un utente. È possibile trovare un utente per un dato `UserSub`richiamando l'API `ListUsers` e utilizzando un filtro per sub.

Eventi relativi ai pool di identità

Eventi di dati

Amazon Cognito registra i seguenti eventi di Amazon Cognito Identity come eventi di dati. CloudTrail Gli eventi relativi ai dati sono operazioni API data-plane ad alto volume che CloudTrail non vengono registrate per impostazione predefinita. Per gli eventi di dati sono previsti costi aggiuntivi.

- GetCredentialsForIdentity
- GetId
- GetOpenIdToken
- GetOpenIdTokenForDeveloperIdentity
- UnlinkIdentity

Per generare CloudTrail log per queste operazioni API, devi attivare gli eventi relativi ai dati nel tuo percorso e scegliere i selettori di eventi per i pool di identità di Cognito. Per ulteriori informazioni, consulta Registrazione di eventi di dati per i percorsi nella Guida per l'utente di AWS CloudTrail .

Puoi anche aggiungere i selettori di eventi dei pool di identità al tuo trail con il seguente comando CLI.

```
aws cloudtrail put-event-selectors --trail-name <trail name> --advanced-event-selectors
 \
"{\
   \"Name\": \"Cognito Selector\",\
   \"FieldSelectors\": [\
      {\
         \"Field\": \"eventCategory\",\
         \"Equals\": [\
            \"Data\"\
         ]\
      },\
      {\
         \"Field\": \"resources.type\",\
         \"Equals\": [\
            \"AWS::Cognito::IdentityPool\"\
         ]\
      }\
   ]\
}"
```

Eventi di gestione

Amazon Cognito registra il resto delle operazioni API dei pool di identità di Amazon Cognito come eventi di gestione. CloudTrail registra le operazioni API degli eventi di gestione per impostazione predefinita.

Per un elenco delle operazioni delle API dei pool di identità di Amazon Cognito a cui Amazon Cognito accede, consulta il riferimento CloudTrail all'API dei pool di identità di Amazon Cognito.

Amazon Cognito Sync

Amazon Cognito registra tutte le operazioni API di Amazon Cognito Sync come eventi di gestione. Per un elenco delle operazioni dell'API Amazon Cognito Sync a cui Amazon Cognito accede, consulta il riferimento CloudTrail all'API Amazon Cognito Sync.

Analisi degli eventi di Amazon CloudTrail Cognito con Amazon CloudWatch Logs Insights

Puoi cercare e analizzare i tuoi CloudTrail eventi di Amazon Cognito con Amazon CloudWatch Logs Insights. Quando configuri il percorso per inviare eventi a CloudWatch Logs, CloudTrail invia solo gli eventi che corrispondono alle impostazioni del percorso.

Per interrogare o ricercare CloudTrail gli eventi di Amazon Cognito, nella CloudTrail console, assicurati di selezionare l'opzione Management events nelle impostazioni del trail in modo da poter monitorare le operazioni di gestione eseguite sulle tue AWS risorse. Se lo desideri, puoi selezionare l'opzione Eventi Insights nelle impostazioni del percorso per identificare errori, attività insolite o comportamenti insoliti dell'utente nell'account.

Query di Amazon Cognito di esempio

Puoi utilizzare le seguenti query nella CloudWatch console Amazon.

Query generali

Trova i 25 log eventi aggiunti più di recente.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com"
```

Ottieni un elenco dei 25 eventi di log aggiunti più di recente che includono eccezioni.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com" and @message like /Exception/
```

Query di eccezioni ed errori

Trova i 25 eventi di log aggiunti più di recente con codice di errore `NotAuthorizedException` insieme al bacino d'utenza di Amazon Cognito `sub`.

```
fields @timestamp, additionalEventData.sub as user | sort @timestamp desc | limit 25
```

```
| filter eventSource = "cognito-idp.amazonaws.com" and errorCode=
 "NotAuthorizedException"
```

Trova il numero di record con `sourceIPAddress` e `eventName` corrispondente.

```
filter eventSource = "cognito-idp.amazonaws.com"
| stats count(*) by sourceIPAddress, eventName
```

Trova i primi 25 indirizzi IP che hanno attivato un errore `NotAuthorizedException`.

```
filter eventSource = "cognito-idp.amazonaws.com" and errorCode=
 "NotAuthorizedException"
| stats count(*) as count by sourceIPAddress, eventName
| sort count desc | limit 25
```

Trova i primi 25 indirizzi IP che hanno chiamato l'API `ForgotPassword`.

```
filter eventSource = "cognito-idp.amazonaws.com" and eventName = 'ForgotPassword'
| stats count(*) as count by sourceIPAddress
| sort count desc | limit 25
```

Esempi di eventi Amazon Cognito

Amazon Cognito registra le informazioni sull'attività di autenticazione degli utenti e AWS CloudTrail sull'attività di gestione amministrativa. Questo vale sia per i pool di utenti che per i pool di identità. Ad esempio, è possibile visualizzare `UpdateIdentityPool` tutti `GetId` gli eventi nello stesso percorso oppure `UpdateAuthEventFeedback` e `SetRiskConfiguration` gli eventi. Verranno inoltre visualizzati i registri del pool di utenti per le attività dell'interfaccia utente ospitata che non corrispondono alle operazioni nell'API dei pool di utenti. Questa sezione contiene alcuni esempi di log che potresti vedere. Per comprendere lo schema CloudTrail degli eventi di qualsiasi operazione, genera una richiesta per quell'operazione ed esamina gli eventi che crea nel tuo percorso.

Un trail può fornire eventi come file di log a un bucket Amazon S3 da te specificato. 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.

Argomenti

- CloudTrail Eventi di esempio per una registrazione all'interfaccia utente ospitata
- CloudTrail Evento di esempio per una richiesta SAML
- CloudTrail Eventi di esempio per le richieste all'endpoint del token
- CloudTrail Evento di esempio per CreateIdentityPool
- CloudTrail Evento di esempio per GetCredentialsForIdentity
- CloudTrail Evento di esempio per GetId
- CloudTrail Evento di esempio per GetOpenIdToken
- CloudTrail Evento di esempio per GetOpenIdTokenForDeveloperIdentity
- CloudTrail Evento di esempio per UnlinkIdentity

CloudTrail Eventi di esempio per una registrazione all'interfaccia utente ospitata

CloudTrail Gli eventi di esempio seguenti mostrano le informazioni che Amazon Cognito registra quando un utente si registra tramite l'interfaccia utente ospitata.

Amazon Cognito registra il seguente evento quando un nuovo utente accede alla pagina di accesso della tua app.


```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-04-06T05:38:12Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Login_GET",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "errorCode": "",
    "errorMessage": "",
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 200.0
        },
        "requestParameters":
```

```
        {
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "response_type":
            [
                "token"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        }
    },
    "eventID": "382ae09a-151d-4116-8f2b-6ac0a804a38c",
    "readOnly": true,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra il seguente evento quando un nuovo utente sceglie Sign up (Registrazione) dalla pagina di accesso della tua app.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-05T23:21:43Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Signup_GET",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
```

```
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 200
        },
        "requestParameters":
        {
            "response_type":
            [
                "code"
            ],
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        },
        "userPoolDomain": "mydomain.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "requestID": "7a63e7c2-b057-4f3d-a171-9d9113264fff",
    "eventID": "5e7b27a0-6870-4226-adb4-f86cd51ac5d8",
    "readOnly": true,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra il seguente evento quando un nuovo utente sceglie un nome utente, immette un indirizzo e-mail e sceglie una password dalla pagina di accesso dell'app. Amazon Cognito non registra le informazioni di identificazione sull'identità dell'utente. CloudTrail

```
{
```

```
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-05T23:22:05Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Signup_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 302
        },
        "requestParameters":
        {
            "password":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "requiredAttributes[email]":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "response_type":
            [
                "code"
            ],
            "_csrf":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
```

```
            ],
            "username":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ]
        },
        "userPoolDomain": "mydomain.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "requestID": "9ad58dd8-3517-4aa8-96a5-d17a01df9eb4",
    "eventID": "c75eb7a5-eb8c-43d1-8331-f4412e756e69",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra il seguente evento quando un nuovo utente accede alla pagina di conferma dell'utente nell'interfaccia utente ospitata dopo la registrazione.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-05T23:22:06Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Confirm_GET",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
```

```
            "status": 200
        },
        "requestParameters":
        {
            "response_type":
            [
                "code"
            ],
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        },
        "userPoolDomain": "mydomain.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "requestID": "58a5b170-3127-45bb-88cc-3e652d779e0b",
    "eventID": "7f87291a-6d50-409a-822f-e3a5ec7e60da",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra il seguente evento quando, nella pagina di conferma dell'utente nell'interfaccia utente ospitata, un utente immette un codice inviato da Amazon Cognito in un messaggio e-mail.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
```

```
    "eventTime": "2022-05-05T23:23:32Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Confirm_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 302
        },
        "requestParameters":
        {
            "confirm":
            [
                ""
            ],
            "deliveryMedium":
            [
                "EMAIL"
            ],
            "sub":
            [
                "704b1e47-34fe-40e9-8c41-504997494531"
            ],
            "code":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "destination":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "response_type":
            [
                "code"
            ],
            "_csrf":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
```

```
            "cognitoAsfData":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
            ],
            "username":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ]
        },
        "userPoolDomain": "mydomain.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "requestID": "9764300a-ed35-4f87-8a0f-b18b3fe2b11e",
    "eventID": "e24ac6e5-2f70-4c6e-ad4e-2f08a547bb36",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

CloudTrail Evento di esempio per una richiesta SAML

Amazon Cognito registra il seguente evento quando un utente che si è autenticato con il tuo IdP SAML invia l'asserzione SAML al tuo endpoint `/saml2/idpresponse`.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
```

```
    },
    "eventTime": "2022-05-06T00:50:57Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "SAML2Response_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 302
        },
        "requestParameters":
        {
            "RelayState":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "SAMLResponse":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ]
        },
        "userPoolDomain": "mydomain.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "requestID": "4f6f15d1-c370-4a57-87f0-aac4817803f7",
    "eventID": "9824b50f-d9d1-4fb8-a2c1-6aa78ca5902a",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "625647942648",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

CloudTrail Eventi di esempio per le richieste all'endpoint del token

Di seguito sono riportati eventi di esempio dalle richieste all'Endpoint Token.

Amazon Cognito registra il seguente evento quando un utente che ha autenticato e ricevuto un codice di autorizzazione invia il codice al tuo endpoint `/oauth2/token`.


```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-12T22:12:30Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Token_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 200
        },
        "requestParameters":
        {
            "code":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "grant_type":
            [
                "authorization_code"
            ],
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
```

```
            ]
        },
        "userPoolDomain": "mydomain.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "requestID": "f257f752-cc14-4c52-ad5b-152a46915238",
    "eventID": "0bd1586d-cd3e-4d7a-abaf-fd8bfc3912fd",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra il seguente evento quando il tuo sistema di back-end invia una richiesta `client_credentials` per un token di accesso al tuo endpoint `/oauth2/token`.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-12T21:07:05Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Token_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 200
        },
        "requestParameters":
        {
```

```
            "grant_type":
            [
                "client_credentials"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        },
        "userPoolDomain": "mydomain.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "requestID": "4f871256-6825-488a-871b-c2d9f55caff2",
    "eventID": "473e5cbc-a5b3-4578-9ad6-3dfdcb8a6d34",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra il seguente evento quando la tua app scambia un token di aggiornamento con un nuovo ID e un token di accesso con il tuo endpoint `/oauth2/token`.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-12T22:16:40Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Token_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
```

```
    {
        "responseParameters":
        {
            "status": 200
        },
        "requestParameters":
        {
            "refresh_token":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "grant_type":
            [
                "refresh_token"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        },
        "userPoolDomain": "mydomain.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "requestID": "2829f0c6-a3a9-4584-b046-11756dfe8a81",
    "eventID": "12bd3464-59c7-44fa-b8ff-67e1cf092018",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

CloudTrail Evento di esempio per CreateIdentityPool

Il seguente esempio è una voce di log per una richiesta dell'operazione `CreateIdentityPool`. La richiesta è stata effettuata da un utente IAM denominato Alice.

```
{
    "eventVersion": "1.03",
```

```
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "['EXAMPLE_KEY_ID']",
        "userName": "Alice"
    },
    "eventTime": "2016-01-07T02:04:30Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "CreateIdentityPool",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "USER_AGENT",
    "requestParameters": {
        "identityPoolName": "TestPool",
        "allowUnauthenticatedIdentities": true,
        "supportedLoginProviders": {
            "graph.facebook.com": "000000000000000"
        }
    },
    "responseElements": {
        "identityPoolName": "TestPool",
        "identityPoolId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
        "allowUnauthenticatedIdentities": true,
        "supportedLoginProviders": {
            "graph.facebook.com": "000000000000000"
        }
    },
    "requestID": "15cc73a1-0780-460c-91e8-e12ef034e116",
    "eventID": "f1d47f93-c708-495b-bff1-cb935a6064b2",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

CloudTrail Evento di esempio per GetCredentialsForIdentity

Il seguente esempio è una voce di log per una richiesta dell'operazione `GetCredentialsForIdentity`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "Unknown"
    },
    "eventTime": "2023-01-19T16:55:08Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "GetCredentialsForIdentity",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.4",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.get-credentials-for-identity",
    "requestParameters": {
        "logins": {
            "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
 "HIDDEN_DUE_TO_SECURITY_REASONS"
        },
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": "aAaAaAaAaAaAab1111111111EXAMPLE",
            "expiration": "Jan 19, 2023 5:55:08 PM"
        },
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "659dfc23-7c4e-4e7c-858a-1abce884d645",
    "eventID": "6ad1c766-5a41-4b28-b5ca-e223ccb00f0d",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
}
```

CloudTrail Evento di esempio per GetId

Il seguente esempio è una voce di log per una richiesta dell'operazione `GetId`.

```
{
```

```
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown"
    },
    "eventTime": "2023-01-19T16:55:05Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "GetId",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.4",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.get-id",
    "requestParameters": {
        "identityPoolId": "us-east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE",
        "logins": {
            "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
 "HIDDEN_DUE_TO_SECURITY_REASONS"
        }
    },
    "responseElements": {
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "dc28def9-07c8-460a-a8f3-3816229e6664",
    "eventID": "c5c459d9-40ec-41fd-8f6b-57865d5a9975",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
}
```

CloudTrail Evento di esempio per GetOpenIdToken

Il seguente esempio è una voce di log per una richiesta dell'operazione `GetOpenIdToken`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown"
```

```
    },
    "eventTime": "2023-01-19T16:55:08Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "GetOpenIdToken",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.4",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.get-open-id-token",
    "requestParameters": {
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
        "logins": {
            "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
 "HIDDEN_DUE_TO_SECURITY_REASONS"
        }
    },
    "responseElements": {
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "a506ba18-10d7-4fdb-9548-a8187b2e38bb",
    "eventID": "19ffc1a6-6ed8-4580-a4e1-3062c5ce6457",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
}
```

CloudTrail Evento di esempio per GetOpenIdTokenForDeveloperIdentity

Il seguente esempio è una voce di log per una richiesta dell'operazione `GetOpenIdTokenForDeveloperIdentity`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA1EXAMPLE:johns-AssumedRoleSession",
```

```
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/johns-AssumedRoleSession",
        "accountId": "111122223333",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA1EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2023-01-19T16:53:14Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-01-19T16:55:08Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "GetOpenIdTokenForDeveloperIdentity",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "27.0.3.154",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.get-open-id-token-for-developer-identity",
    "requestParameters": {
        "tokenDuration": 900,
        "identityPoolId": "us-east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE",
        "logins": {
            "JohnsDeveloperProvider": "HIDDEN_DUE_TO_SECURITY_REASONS"
        }
    },
    "responseElements": {
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "b807df87-57e7-4dd6-b90c-b06f46a61c21",
    "eventID": "f26fed91-3340-4d70-91ae-cdf555547b76",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
```

```
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
}
```

CloudTrail Evento di esempio per UnlinkIdentity

Il seguente esempio è una voce di log per una richiesta dell'operazione `UnlinkIdentity`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown"
    },
    "eventTime": "2023-01-19T16:55:08Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "UnlinkIdentity",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.4",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.unlink-identity",
    "requestParameters": {
        "logins": {
            "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
 "HIDDEN_DUE_TO_SECURITY_REASONS"
        },
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
        "loginsToRemove": ["cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa"]
    },
    "responseElements": null,
    "requestID": "99c2c8e2-9c29-416f-bb17-b650a5cbada9",
    "eventID": "d8e26126-202a-43c2-b458-3f225efaedc7",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
```

```
}
```

Accedi ad Amazon Cognito utilizzando un endpoint di interfaccia (AWS PrivateLink)

Puoi usarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e Amazon Cognito. Puoi accedere ad Amazon Cognito come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. Direct Connect Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per accedere ad Amazon Cognito.

Stabilisci questa connessione privata creando un endpoint di interfaccia attivato da AWS PrivateLink. In ciascuna sottorete viene creata un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Si tratta di interfacce di rete gestite dai richiedenti che fungono da punto di ingresso per il traffico destinato ad Amazon Cognito.

Per ulteriori informazioni, consulta la sezione Accesso a Servizi AWS tramite AWS PrivateLink nella Guida di AWS PrivateLink .

> **Important**
>
> I seguenti tipi di autenticazione non sono attualmente supportati tramite: AWS PrivateLink
>
> 1. L'autorizzazione da macchina a macchina (M2M) con le credenziali del client OAuth 2.0 proviene dall'endpoint del token. L'autorizzazione M2M con l'operazione `GetClientToken` API è supportata, poiché non richiede un dominio del pool di utenti.
> 2. Sign-in con accesso gestito e la classica interfaccia utente ospitata.

Argomenti

- Flussi di autenticazione per AWS PrivateLink integrazione
- Modalità operative per AWS PrivateLink
- Considerazioni per Amazon Cognito
- Controllo dell'accesso con politiche di controllo delle risorse
- Crea un endpoint di interfaccia per Amazon Cognito
- Creazione di una policy dell' endpoint per l'endpoint dell'interfaccia

- Crea una politica basata sull'identità per AWS PrivateLink operazioni

Flussi di autenticazione per AWS PrivateLink integrazione

La tabella seguente descrive i flussi di autenticazione disponibili per i client nei VPC e le politiche IAM che è possibile applicare per gestirli. Le policy che puoi valutare nelle richieste ai pool di utenti sono le policy di controllo delle risorse (RCP), le policy degli endpoint VPC e le policy basate sull'identità.

Risorsa	Flusso di autenticazione	Policy valutate quando il client transita su un endpoint VPC	Politiche valutate quando l'origine del client è pubblica
Bacino d'utenza	Accesso gestito e accesso classico all'interfaccia utente ospitata	Nessuno (nessun accesso) [1]	Nessuno [2]
Bacino d'utenza	Machine-to-machine autorizzazione dall'endpoint del token (l'autorizzazione M2M con il funzionamento dell'`GetClientToken` API è coperta dalla riga delle richieste non autenticate delle API SDK e REST)	Nessuna (nessun accesso) [1]	Nessuno [2]
Bacino d'utenza	Richieste non autenticate per SDK e API REST	RCP, politiche degli endpoint VPC [3]	RCP
Bacino d'utenza	Richieste autenticate SDK e REST API SigV4	RCP, policy degli endpoint VPC, policy basate sull'identità [3]	RCP, politiche basate sull'identità
Pool di identità	Richieste non autenticate tramite SDK e API REST (flussi di base e avanzati)	RCP, politiche degli endpoint VPC	RCP

Risorsa	Flusso di autenticazione	Policy valutate quando il client transita su un endpoint VPC	Politiche valutate quando l'origine del client è pubblica
Pool di identità	Richieste autenticate SDK e REST API SigV4 (flusso autenticato dallo sviluppatore)	RCP, politiche basate sull'identità	RCP, politiche basate sull'identità

[1] Gli endpoint VPC non accettano richieste per i domini del pool di utenti. Se il client dispone di un percorso verso Internet, viene applicato il NAT, rendendo pubblica l'origine.

[2] L'esistenza di un dominio del pool di utenti impedisce il completamento di qualsiasi richiesta del pool di utenti che transita su un endpoint VPC. Qualsiasi client può utilizzare percorsi di transito pubblico solo verso il dominio del pool di utenti e gli endpoint del servizio API, rendendo l'endpoint VPC inutilizzabile per il pool di utenti. I pool di utenti con domini assegnati sono incompatibili con. AWS PrivateLink

[3 Al] pool di utenti non deve essere assegnato un dominio.

Modalità operative per AWS PrivateLink

I seguenti modelli di implementazione di esempio sono supportati con AWS PrivateLink Amazon Cognito.

Risorsa	Implementazione	Azioni
Bacino d'utenza	Applicazione SDK o API REST completamente privata	1. Eliminare il dominio 2. Creare un endpoint VPC 3. Configura RCP per `Deny` tutte le azioni cognito-idp ad eccezione di VPC
Bacino d'utenza	IP pubblici e privati	1. Eliminare il dominio 2. Creare un endpoint VPC
Bacino d'utenza	Server di autorizzazione OAuth 2.0 privato o pubblico	1. Non disponibile per VPC

Risorsa	Implementazione	Azioni
Pool di identità	Completamente privato	1. Creare un endpoint VPC 2. Configura RCP per `Deny` tutte le azioni di cognito-identità ad eccezione di VPC
Pool di identità	IP pubblici e privati	1. Creare un endpoint VPC

Considerazioni per Amazon Cognito

Prima di configurare un endpoint di interfaccia per Amazon Cognito, consulta le considerazioni nella Guida.AWS PrivateLink Amazon Cognito supporta le chiamate a tutte le azioni dell'API Amazon Cognito tramite l'endpoint dell'interfaccia. Per ulteriori informazioni su queste operazioni, consulta Amazon Cognito user pools API Reference e Amazon Cognito Federated Identities API Reference.

Argomenti

- Pool di utenti e AWS PrivateLink
- Pool di identità e AWS PrivateLink

Pool di utenti e AWS PrivateLink

È possibile effettuare richieste a tutte le operazioni API dei pool di utenti tramite l'endpoint dell'interfaccia, ma non alle operazioni richieste dall'applicazione al server di autorizzazione OAuth 2.0 del pool di utenti, ad esempio la concessione delle credenziali dei client e l'accesso gestito.

L'API dei pool di `cognito-idp` utenti presenta operazioni API non autenticate, autenticate e autorizzate tramite token. Puoi concedere le autorizzazioni per le operazioni autenticate nelle politiche di controllo degli endpoint e delle risorse VPC. È inoltre possibile concedere autorizzazioni per operazioni non autenticate e autorizzate tramite token, a differenza delle politiche basate sull'identità. I tipi di policy di controllo degli endpoint e delle risorse VPC sono in grado di valutare e rifiutare o consentire richieste per operazioni altrimenti pubbliche.

Anche le richieste agli endpoint di dominio sono pubbliche, ma non è possibile valutarle nelle policy. Il DNS privato VPC non indirizza le richieste per i domini del pool di utenti all'endpoint VPC. È possibile effettuare richieste di servizi di dominio solo tramite percorsi Internet pubblici. Per ulteriori informazioni, consulta Effetti delle politiche sulle operazioni del pool di utenti.

Argomenti

- Operazioni supportate
- Effetti delle politiche sulle operazioni del pool di utenti

Operazioni supportate

I sistemi in un VPC possono inviare richieste alle azioni API del pool di utenti ma non agli endpoint del dominio del pool di utenti. I flussi di lavoro OpenID Connect (OIDC) e OAuth 2.0 che utilizzano endpoint di dominio, ad esempio machine-to-machine (M2M), accesso federato e concessioni di codici di autorizzazione, sono inaccessibili tramite gli endpoint VPC. Le policy degli endpoint VPC non hanno alcun effetto su questi flussi di lavoro HTTP e non possono elaborarli. Le richieste agli endpoint del dominio dall'interno di un VPC hanno sempre esito negativo sull'endpoint dell'interfaccia, ma continuano a essere disponibili tramite DNS pubblico e routing quando configuri gli endpoint VPC per i tuoi pool di utenti.

Per impedire l'assegnazione di domini dai sistemi in un VPC, Amazon Cognito blocca `CreateUserPoolDomain` le richieste sull'endpoint dell'interfaccia. Ciò impedisce l'aggiunta di domini ai pool di utenti da sistemi che si trovano in un VPC. Per impedire l'aggiunta di un dominio da tutti i sistemi, applica una politica di controllo delle risorse (RCP) come nell'esempio seguente al tuo. Account AWS Questa politica blocca l'`CreateUserPoolDomain`azione contro il pool di utenti specificato.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Principal": "*",
            "Effect": "Deny",
            "Action": [
                "cognito-idp:CreateUserPoolDomain"
            ],
            "Resource": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_EXAMPLE"
        }
    ]
}
```

Il tuo pool di utenti potrebbe avere un dominio e in tutti i casi tale dominio non è disponibile tramite AWS PrivateLink. Tutte le richieste API del pool di SDK-based utenti agli endpoint di `cognito-`

idp servizio accettano le richieste AWS PrivateLink, ad eccezione di`CreateUserPoolDomain`. Gli endpoint del servizio API del pool di utenti e gli endpoint del dominio rimangono sempre accessibili tramite percorsi Internet pubblici. Per gestire l'accesso da fonti pubbliche, implementa gli ACL Web.AWS WAF

Effetti delle politiche sulle operazioni del pool di utenti

Tutte le operazioni API del pool di utenti, anche quelle tipicamente pubbliche e non autenticate, possono essere controllate nelle policy degli endpoint VPC e nelle politiche di controllo delle risorse (RCP). Puoi anche applicare restrizioni all'accesso al pool di utenti in policy basate sull'identità con chiavi di condizione VPC. Solo le richieste che includono informazioni di autenticazione in formato SigV4 possono essere controllate in policy basate sull'identità. L'accesso gestito e le operazioni classiche dell'interfaccia utente ospitata sono una categoria separata e non sono idonee al transito VPC o all'applicazione di alcun tipo di policy alle loro azioni.

Operazioni non autenticate

Le operazioni di Amazon Cognito per le applicazioni lato client non sono autenticate con SigV4. Le operazioni di esempio sono riportate nella policy di esempio all'indirizzo. Creazione di una policy dell' endpoint per l'endpoint dell'interfaccia Altri esempi di operazioni non autenticate sono `GetUser` e. `AssociateSoftwareToken` Quando si aggiungono queste operazioni alle politiche basate sull'identità, non hanno alcun effetto. Tuttavia, puoi consentire o limitare l'accesso a queste azioni nelle policy e negli RCP degli endpoint VPC.

Le operazioni non autenticate non sono associate a un principal IAM. La tua policy sugli endpoint VPC o RCP deve consentire a tutti i principali di eseguire queste azioni.

Operazioni autenticate

Le operazioni API per l'amministrazione del pool di utenti e l'autenticazione lato server sono autenticate con SigV4. Per le operazioni autenticate, puoi limitare i principali con le policy degli endpoint che applichi all'endpoint VPC, le politiche di controllo delle risorse nell'organizzazione e le politiche basate sull'identità che applichi ai principali. Identity-based e le politiche di controllo delle risorse prevedono chiavi di condizione basate sulla rete come e. VPC-aware `aws:SourceVpc` `aws:SourceVpce`

Per ulteriori informazioni sulle classi di operazioni API lato server, lato client e amministrative per pool di utenti, consulta. Modelli di autorizzazione per l'autenticazione tramite API e SDK

Pool di identità e AWS PrivateLink

I pool di identità di Amazon Cognito supportano tutte le operazioni API tramite. AWS PrivateLink

Argomenti

- Operazioni supportate
- Limitazioni del contesto di rete con AWS STS integrazione
- Service-specific chiavi di contesto

Operazioni supportate

Tutte le operazioni delle API dei pool di identità sono supportate tramite l'endpoint dell'interfaccia. I pool di identità non hanno endpoint di dominio e non sono soggetti alle stesse limitazioni. Tuttavia, i pool di identità hanno considerazioni specifiche per i controlli degli accessi basati sulla rete grazie alla loro integrazione con. AWS STS

Limitazioni del contesto di rete con AWS STS integrazione

I pool di identità utilizzano AWS STS `AssumeRoleWithWebIdentity` le operazioni per fornire AWS credenziali temporanee. Quando i pool di identità AWS STS richiamano `aws:SourceIp` il flusso di autenticazione avanzato, le chiavi del contesto di rete `aws:SourceVpc` utilizzano e `aws:SourceVpce` contengono valori provenienti dall'infrastruttura del servizio di pool di identità, non dal contesto di rete dell'applicazione. AWS PrivateLink

Se le politiche di trust dei ruoli IAM o le politiche di controllo delle risorse (RCP) utilizzano chiavi di condizione basate sulla rete per limitare l'accesso, le operazioni dei pool di identità potrebbero essere negate in modo imprevisto. Per ovviare a questa limitazione, puoi utilizzare uno dei seguenti approcci:

Tag principali per l'identificazione del servizio

Etichetta i ruoli IAM utilizzati con i pool di identità e modifica le tue politiche per consentire le operazioni quando il principale ha il tag appropriato. Innanzitutto, aggiungi un tag al ruolo del tuo pool di identità:

```
aws iam tag-role \
    --role-name MyIdentityPoolRole \
    --tags Key=CognitoServiceCall,Value=true
```

Quindi modifica le politiche basate sulla rete per consentire i principali con tag. Ad esempio, in un RCP:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Resource": "*",
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": ["allowed-ip-ranges"]
                },
                "StringNotEqualsIfExists": {
                    "aws:ResourceTag/CognitoServiceCall": "true"
                }
            }
        }
    ]
}
```

Service-specific chiavi di contesto

I pool di identità forniscono chiavi di contesto specifiche del servizio per l'autorizzazione a livello di risorsa nelle policy degli endpoint VPC e negli RCP. Con queste chiavi contestuali, puoi abilitare un controllo granulare degli accessi e distinguere tra utenti autenticati e non autenticati nelle policy.

Sono disponibili chiavi contestuali specifiche del servizio per operazioni non SIGv4 come GetId, GetCredentialsForIdentity, GetOpenIdToken, UnlinkIdentity

- `cognito-identity-unauth:IdentityPoolArn`- Filtra l'accesso tramite l'ARN del pool di identità per gli utenti non autenticati
- `cognito-identity-unauth:AccountId`- Filtra l'accesso in base all' Account AWS ID per gli utenti non autenticati
- `cognito-identity-auth:IdentityPoolArn`- Filtra l'accesso tramite l'ARN del pool di identità per gli utenti autenticati
- `cognito-identity-auth:AccountId`- Filtra l'accesso in base all' Account AWS ID per gli utenti autenticati

Sono disponibili chiavi contestuali specifiche del servizio per operazioni SIGv4 come DeleteIdentities and DescribeIdentity

- `cognito-identity:IdentityPoolArn`- Filtra l'accesso tramite il pool di identità ARN

Puoi utilizzare queste chiavi di contesto nelle policy degli endpoint VPC per limitare l'accesso in base allo stato di autenticazione, come dimostrato nell'esempio seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "cognito-identity:GetId",
                "cognito-identity:GetCredentialsForIdentity"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cognito-identity-unauth:IdentityPoolArn": "arn:aws:cognito-
identity:us-east-1:123456789012:identitypool/us-east-1:12345678-ffff-ffff-ffff-123456"
                }
            }
        }
    ]
}
```

Controllo dell'accesso con politiche di controllo delle risorse

Amazon Cognito supporta il controllo dell'accesso alle risorse con politiche di controllo delle risorse (RCP). Con le chiavi di condizione basate sulla rete, gli RCP possono definire le reti e le azioni consentite per l' AWS PrivateLink accesso ai pool di utenti e ai pool di identità. Le `Action` istruzioni contenute negli RCP possono controllare l'accesso alle operazioni API del pool di utenti autenticate e non autenticate.

Ad esempio, la seguente politica di esempio impedisce l'accesso a tutti i pool di utenti da un VPC specifico.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyCognitoAccessOutsideVPC",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "cognito-idp:*",
      "Resource": "*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceVpc": "vpc-02d6770f46ef1653b"
        }
      }
    }
  ]
}
```

Crea un endpoint di interfaccia per Amazon Cognito

Puoi creare un endpoint di interfaccia per Amazon Cognito utilizzando la console Amazon VPC o il (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consulta la sezione Creazione di un endpoint di interfaccia nella Guida per l'utente di AWS PrivateLink .

Crea un endpoint di interfaccia per i pool di utenti di Amazon Cognito utilizzando il seguente nome di servizio:

```
com.amazonaws.region.cognito-idp
```

Crea un endpoint di interfaccia per i pool di identità di Amazon Cognito utilizzando il seguente nome di servizio:

```
com.amazonaws.region.cognito-identity
```

Se abiliti il DNS privato per l'endpoint dell'interfaccia, puoi effettuare richieste API ad Amazon Cognito utilizzando il nome DNS regionale predefinito. Ad esempio, per i pool di utenti e `cognito-idp.us-east-1.amazonaws.com` `cognito-identity.us-east-1.amazonaws.com` per i pool di identità.

Creazione di una policy dell' endpoint per l'endpoint dell'interfaccia

Una policy dell'endpoint è una risorsa IAM che è possibile allegare all'endpoint dell'interfaccia. La policy predefinita per gli endpoint consente l'accesso completo ad Amazon Cognito tramite l'endpoint dell'interfaccia. Per controllare l'accesso consentito ad Amazon Cognito dal tuo VPC, collega una policy personalizzata per gli endpoint all'endpoint dell'interfaccia.

Una policy di endpoint specifica le informazioni riportate di seguito:

- I principali che possono eseguire azioni (Account AWS, utenti IAM e ruoli IAM).
- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.
- Le condizioni che devono essere soddisfatte prima che la richiesta venga consentita o rifiutata.

Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai servizi con policy di endpoint nella Guida di AWS PrivateLink .

Esempio: policy degli endpoint VPC per le azioni del pool di utenti

Di seguito è riportato un esempio di policy endpoint personalizzata per pool di utenti. Quando si allega questa policy all'endpoint di interfaccia, si concede l'accesso alle azioni del pool di utenti elencate per tutti i principali utenti su tutte le risorse.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:assumed-role/MyWebAppRole/
MyWebAppSession"
         },
         "Effect": "Allow",
         "Action": [
            "cognito-idp:AdminInitiateAuth",
            "cognito-idp:AdminRespondToAuthChallenge",
            "cognito-idp:AdminSetUserPassword"
         ],
         "Resource":"arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_EXAMPLE"
      },
```

```
      {
         "Effect": "Allow",
         "Action": [
            "cognito-idp:InitiateAuth",
            "cognito-idp:RespondToAuthChallenge",
            "cognito-idp:ForgotPassword",
            "cognito-idp:ConfirmForgotPassword"
         ],
         "Resource":"arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_EXAMPLE"
      }
   ]
}
```

Esempio: policy degli endpoint VPC per le azioni del pool di identità

Di seguito è riportato un esempio di policy endpoint personalizzata per pool di identità. Questa policy utilizza chiavi contestuali specifiche del servizio per limitare l'accesso agli utenti autenticati da uno specifico pool di identità.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": [
            "cognito-identity:GetId",
            "cognito-identity:GetCredentialsForIdentity",
            "cognito-identity:GetOpenIdToken"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "cognito-identity-auth:IdentityPoolArn": "arn:aws:cognito-identity:us-
east-1:123456789012:identitypool/us-east-1:12345678-ffff-ffff-ffff-123456"
            }
         }
      }
   ]
}
```

Crea una politica basata sull'identità per AWS PrivateLink operazioni

Identity-basedle policy sono risorse IAM che puoi collegare ai principali. AWS Puoi controllare l'accesso ad Amazon Cognito tramite endpoint VPC con policy operative basate sull'identità. IAM-authenticated A differenza delle policy sugli endpoint, non è possibile configurare le autorizzazioni per operazioni non autenticate nelle politiche basate sull'identità. Le operazioni autenticate o amministrative richiedono l'autorizzazione Signature Version 4. Per i pool di utenti, le operazioni autenticate includono richieste di autenticazione lato server AdminInitiateAuthe richieste amministrative come. UpdateUserPool Per i pool di identità, le operazioni autenticate includono richieste amministrative come e. DeleteIdentitiesDescribeIdentity

Una politica basata sull'identità specifica le seguenti informazioni:

- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.
- Le condizioni che devono essere soddisfatte prima che la richiesta venga consentita o rifiutata.

Esempio: politica basata sull'identità per l'autenticazione lato server del pool di utenti

La seguente politica di esempio concede l'accesso alle azioni del pool di utenti elencate nel pool di utenti specificato, dall'endpoint specificato. Applica questa policy al ruolo IAM presunto per la tua applicazione web.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cognito-idp:AdminInitiateAuth",
                "cognito-idp:AdminRespondToAuthChallenge",
                "cognito-idp:AdminSetUserPassword"
            ],
            "Resource": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_EXAMPLE",
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpce": "vpce-1a2b3c4d"
                }
            }
```

```
        }
    ]
}
```

Esempio: politica basata sull'identità per le operazioni amministrative del pool di identità

La seguente politica di esempio concede l'accesso alle azioni amministrative del pool di identità dall'endpoint VPC specificato. Applica questa policy al responsabile IAM che deve eseguire l'amministrazione del pool di identità.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cognito-identity:DeleteIdentities",
                "cognito-identity:DescribeIdentity"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpce": "vpce-1a2b3c4d",
                    "cognito-identity:IdentityPoolArn": "arn:aws:cognito-identity:us-
east-1:123456789012:identitypool/us-east-1:12345678-ffff-ffff-ffff-123456"
                }
            }
        }
    ]
}
```

Convalida della conformità per Amazon Cognito

Third-party i revisori valutano la sicurezza e la conformità di Amazon Cognito nell'ambito di AWS diversi programmi di conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, consulta la sezione AWS Servizi rientranti nell'ambito del programma di conformità (AWS servizi inclusi). Per informazioni generali, consulta Programmi per la conformità di AWS.

È possibile scaricare i report di controllo di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La responsabilità di conformità durante l'utilizzo di Amazon Cognito è determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'azienda e dalle normative vigenti. Per semplificare il rispetto della conformità, AWS mette a disposizione le seguenti risorse:

- Guide Quick Start per la sicurezza e la conformità. Queste guide all'implementazione illustrano considerazioni relative all'architettura e forniscono i passaggi per l'implementazione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Whitepaper sulla sicurezza e la conformità HIPAA: questo white paper descrive come le aziende possono utilizzare per creare applicazioni. AWS HIPAA-compliant
- AWS risorse per la conformità: risorse per AWS : questa raccolta di cartelle di lavoro e guide potrebbe riguardare il settore e la località in cui operate.
- Valutazione delle risorse in base alle regole contenute nella Guida per gli AWS Config sviluppatori: AWS Config valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub CSPM— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente di verificare la conformità agli standard e alle best practice del settore della sicurezza. AWS

Resilienza in Amazon Cognito

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, connesse tramite reti altamente ridondanti, a bassa latenza e throughput elevato. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta infrastruttura AWS globale.

Argomenti

- Considerazioni sui dati regionali

Considerazioni sui dati regionali

I pool di utenti di Amazon Cognito vengono creati ciascuno in una AWS regione e archiviano i dati del profilo utente solo in quella regione. I pool di utenti possono inviare dati utente a una AWS regione diversa, a seconda di come sono configurate le funzionalità opzionali.

- Se l'impostazione di default `no-reply@verificationemail.com` di indirizzo e-mail viene utilizzata per la verifica di instradamento degli indirizzi e-mail dei bacini d'utenza di Amazon Cognito, le e-mail vengono instradate nella stessa regione del bacino d'utenza associato.
- Se viene utilizzato un indirizzo e-mail diverso per configurare Amazon Simple Email Service (Amazon SES) con i pool di utenti di Amazon Cognito, tale indirizzo e-mail viene instradato AWS attraverso la regione associata all'indirizzo e-mail in Amazon SES.
- I messaggi SMS provenienti dai bacini d'utenza di Amazon Cognito vengono instradati attraverso la stessa regione Amazon SNS, a meno che non sia diversamente indicato in Configurazione della verifica di e-mail o telefono.
- Se si utilizza l'analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito, i dati di eventi vengono instradati alla regione Stati Uniti orientali (Virginia settentrionale).

> **Note**
>
> Amazon Pinpoint è disponibile in diverse AWS regioni in Nord America, Europa, Asia e Oceania. Le regioni di Amazon Pinpoint includono l'API Amazon Pinpoint. Se una regione Amazon Pinpoint è supportata da Amazon Cognito, Amazon Cognito invia eventi ai progetti Amazon Pinpoint all'interno della stessa regione Amazon Pinpoint. Se una regione non è supportata da Amazon Pinpoint, Amazon Cognito supporta solo l'invio di eventi in us-east-1. Per informazioni dettagliate sulla regione di Amazon Pinpoint, consultare endpoint e quote di Amazon Pinpoint e Utilizzo delle analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito.

Sicurezza dell'infrastruttura in Amazon Cognito

In quanto servizio gestito, Amazon Cognito è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi di AWS 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 ad Amazon Cognito 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 crittografia con Perfect Forward Secrecy (PFS) come DHE (Ephemeral) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Analisi della configurazione e delle vulnerabilità nei bacini d'utenza di Amazon Cognito

AWS gestisce le attività di sicurezza di base come l'applicazione di patch al sistema operativo guest (OS) e 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 :

- Convalida della conformità per Amazon Cognito
- Modello di responsabilità condivisa

AWS politiche gestite per Amazon Cognito

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le politiche AWS gestite che scrivere le politiche da soli. 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 rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste politiche coprono casi d'uso comuni e sono disponibili nel tuo AWS account. Per ulteriori informazioni sulle policy AWS gestite, consulta le policy AWS gestite nella IAM User Guide.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi occasionalmente aggiungono altre autorizzazioni a una policy gestita da AWS per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una policy gestita da AWS quando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy ReadOnlyAccess AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consultare la sezione Policy gestite da AWS per funzioni di processi nella Guida per l'utente di IAM.

AWS policy IAM gestite che garantiscono l'accesso ad Amazon Cognito

- `AmazonCognitoPowerUser`: autorizzazioni per accedere e gestire tutti gli aspetti dei tuoi pool di identità e bacini d'utenza. Per visualizzare le autorizzazioni relative a questa politica, consulta. AmazonCognitoPowerUser
- `AmazonCognitoReadOnly`: autorizzazioni per accesso in sola lettura ai tuoi pool di identità e ai tuoi bacini d'utenza. Per visualizzare le autorizzazioni relative a questa politica, vedere. AmazonCognitoReadOnly
- `AmazonCognitoDeveloperAuthenticatedIdentities`: autorizzazioni per il tuo sistema di autenticazione a integrarsi con Amazon Cognito. Per visualizzare le autorizzazioni relative a questa politica, vedere. AmazonCognitoDeveloperAuthenticatedIdentities

Queste policy sono gestite dal team di Amazon Cognito, per cui gli utenti continueranno ad avere lo stesso livello di accesso anche se verranno aggiunte nuove API.

> **Note**
>
> Quando crei un nuovo pool di identità, puoi creare automaticamente nuovi ruoli per l'accesso utente utenti autenticato e guest. L'amministratore che crea il pool di identità con nuovi ruoli IAM deve anche disporre delle autorizzazioni IAM per creare i ruoli.

I pool di identità con accesso guest non autenticato applicano una politica AWS gestita aggiuntiva come politica di sessione agli utenti non autenticati. Questa politica AWS gestita non ha alcun uso amministrativo previsto. Limita invece l'ambito delle autorizzazioni che è possibile applicare agli utenti guest nel flusso di autenticazione migliorato del pool di identità. Per ulteriori informazioni, consulta Ruoli IAM.

AWS politiche IAM gestite che Amazon Cognito concede agli utenti ospiti

- `AmazonCognitoUnAuthedIdentitiesSessionPolicy`- In combinazione con una policy di sessione in linea, limita le autorizzazioni che gli amministratori IAM possono concedere agli utenti ospiti del pool di identità. Amazon Cognito applica automaticamente questa politica alle sessioni guest. Per ulteriori informazioni, consulta Il AWS politica di sessione gestita per gli ospiti.

Amazon Cognito si aggiorna a AWS policy gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon Cognito da quando questo servizio ha iniziato a tracciare queste modifiche. Per gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivere il feed RSS nella pagina Document history (Cronologia dei documenti) di Amazon Cognito.

Modifica	Descrizione	Data
`AmazonCognitoPowerUser` —Modifica	Amazon Cognito ha aggiunto nuove azioni per consentire l'uso del funzionamento dell' AWS End User Messaging SMS API DescribeAccountAttributesper gli utenti con potere amministrativo dei pool di utenti di Amazon Cognito.	27 febbraio 2025
`AmazonCognitoUnAuthedIdentitiesSessionPolicy` —Cambia	Amazon Cognito ha aggiunto nuove azioni per consentire l'uso di AWS Key Management Service utenti non autenticati (ospiti) nei pool di identità.	30 ottobre 2024

Modifica	Descrizione	Data
`AmazonCognitoUnAuthedIdentitiesSessionPolicy` —Modifica	Amazon Cognito ha aggiunto nuove azioni per consentire l'uso di Amazon Location Service per utenti non autenticati (ospiti) nei pool di identità.	9 agosto 2024
`AmazonCognitoUnAuthedIdentitiesSessionPolicy` —Nuova politica	È stata aggiunta una policy AWS gestita per la riduzione dei privilegi degli utenti ospiti nei pool di identità.	14 luglio 2023
`AmazonCognitoPowerUser` e —Cambia `AmazonCognitoReadOnly`	Sono state aggiunte nuove autorizzazioni per consentire agli utenti esperti di visualizzare e gestire le associazioni di ACL AWS WAF Web ai pool di utenti di Amazon Cognito. Sono state aggiunte nuove autorizzazioni per consentire agli utenti di sola lettura di visualizzare le associazioni degli ACL AWS WAF Web ai pool di utenti di Amazon Cognito.	19 luglio 2022

Modifica	Descrizione	Data
`AmazonCognitoPowerUser` —Modifica	Aggiunta una nuova autorizzazione per consentire ad Amazon Cognito di chiamare le operazioni `PutIdentityPolicy` e `ListConfigurationSets` di Amazon Simple Notification Service. Questa modifica consente ai bacini d'utenza di Amazon Cognito di aggiornare le policy di autorizzazione di invio di Amazon SES e di applicare i set di configurazione Amazon SES quando configuri l'invio di e-mail nel bacino d'utenza.	17 novembre 2021
`AmazonCognitoPowerUser` —Cambia	Aggiunta una nuova autorizzazione per consentire ad Amazon Cognito di chiamare l'operazione `GetSMSSandboxAccountStatus` di Amazon Simple Notification Service. Questa modifica consente ai bacini d'utenza di Amazon Cognito di decidere se è necessario uscire dalla sandbox di Amazon Simple Notification Service per inviare messaggi a tutti gli utenti finali tramite i bacini d'utenza.	1° giugno 2021

Modifica	Descrizione	Data
Amazon Cognito ha iniziato a monitorare le modifiche	Amazon Cognito ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	1° marzo 2021

Legacy

> **Important**
>
> Amazon Cognito Sync è chiuso ai nuovi clienti. Se state avviando una nuova applicazione, utilizzate invece un'alternativa corrente come AWS AppSyncDynamoDB. Il contenuto di questa sezione è riservato ai clienti con implementazioni Amazon Cognito Sync esistenti.

Questa sezione raccoglie le funzionalità di Amazon Cognito di generazione precedente. Amazon Cognito Sync è chiuso ai nuovi clienti, come indicato sopra. L'interfaccia utente ospitata classica è una versione di branding precedente che puoi ancora selezionare per client dell'app, ma l'accesso gestito è la versione di branding corrente per i nuovi lavori; per spostare un client di app dalla classica interfaccia utente ospitata all'accesso gestito, consulta. Migrazione dalla classica interfaccia utente ospitata all'accesso gestito

Argomenti

- Amazon Cognito Sync
- Utilizzo di Amazon Cognito Sync con pool di identità
- Personalizzazione del marchio dell'interfaccia utente ospitata (classica)

Amazon Cognito Sync

> **Note**
>
> Amazon Cognito Sync non è più aperto a nuovi clienti. Per le alternative ad Amazon Cognito Sync, esplora e AWS AppSyncDynamoDB. Ulteriori informazioni.

> **Note**
>
> Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio per la sincronizzazione dei dati delle applicazioni tra dispositivi.

Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito Sync è una Servizio AWS libreria client che consente di sincronizzare i dati utente relativi alle applicazioni tra dispositivi. Amazon Cognito Sync può sincronizzare i dati del profilo utente su dispositivi mobili e sul Web senza utilizzare il tuo backend. Le librerie client archiviano localmente i dati nella cache, in modo che la tua app sia in grado di leggere e scrivere i dati indipendentemente dallo stato di connettività del dispositivo. Quando il dispositivo è online, puoi sincronizzare i dati. Se configuri la sincronizzazione push, puoi avvisare immediatamente altri dispositivi della disponibilità di un aggiornamento.

Per informazioni sulla disponibilità regionale delle identità di Amazon Cognito, consulta Disponibilità delle regioni del servizio AWS.

Per ulteriori informazioni su Amazon Cognito Sync consulta i seguenti argomenti.

Argomenti

- Modifica della disponibilità di Amazon Cognito Sync
- Nozioni di base su Amazon Cognito Sync
- Sincronizzazione dei dati tra client
- Gestione dei callback degli eventi
- Implementazione della sincronizzazione push
- Implementazione dei flussi di Amazon Cognito Sync
- Personalizzazione dei flussi di lavoro con Amazon Cognito Events

Modifica della disponibilità di Amazon Cognito Sync

Amazon Cognito Sync non è più aperto a nuovi clienti. Per le alternative ad Amazon Cognito Sync, esplora e AWS AppSyncDynamoDB.

Questa pagina fornisce informazioni sulle modifiche ad Amazon Cognito Sync e include alternative per i clienti Amazon Cognito Sync.

AWS AppSyncè un servizio serverless che consente agli sviluppatori di creare e gestire le API GraphQL. DynamoDB è un archivio chiave-valore per dati per utente come profili, preferenze e

impostazioni delle applicazioni. La migrazione da Amazon Cognito Sync richiede il trasferimento di dati e l'aggiornamento delle applicazioni alle diverse API e modelli di query tramite il nuovo servizio.

Se hai altre domande, contatta l'AWS assistenza.

Domande frequenti

Cosa significa questo per il servizio?

Amazon Cognito Sync non accetterà più nuovi clienti. Il servizio continuerà a funzionare per i clienti esistenti, ma non ci sarà lo sviluppo di nuove funzionalità.

In che modo verranno influenzati i clienti esistenti?

I clienti esistenti non subiranno alcuna interruzione dei loro carichi di lavoro. Puoi continuare a utilizzare Amazon Cognito Sync normalmente.

Quali sono le alternative consigliate ad Amazon Cognito Sync?

- AWS AppSync— Per la sincronizzazione dei dati in tempo reale tra dispositivi con API GraphQL.
- DynamoDB — Per una semplice memorizzazione dei dati utente con valori chiave (profili, preferenze, impostazioni).

Come faccio a migrare i miei dati?

La migrazione richiede l'aggiornamento dello schema dei dati, la migrazione dei dati dallo store Amazon Cognito Sync all'alternativa prescelta e l'aggiornamento delle applicazioni client per utilizzare i nuovi servizi.

Come posso ricevere assistenza in caso di problemi?

Se hai domande in merito, contatta l'AWS assistenza.

Nozioni di base su Amazon Cognito Sync

> **Note**
>
> Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio per la sincronizzazione dei dati delle applicazioni tra dispositivi.

Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

> **Note**
>
> Gli esempi di codice in questa sezione utilizzano il client Amazon Cognito Sync nell'SDK AWS Mobile archiviato (per Android, iOS, Unity e Xamarin/.net). Tale Mobile SDK non è più gestito e i relativi pacchetti non sono disponibili negli attuali repository di pacchetti. Gli esempi rimangono validi per i clienti esistenti che continuano a effettuare l'integrazione con quell'SDK. Per un nuovo sviluppo, usa AWS AppSync.

Amazon Cognito Sync è una libreria di AWS servizi e client che consente la sincronizzazione tra dispositivi dei dati utente relativi alle applicazioni. Puoi usarlo per sincronizzare i dati del profilo utente tra dispositivi mobili e i Web. Le librerie client memorizzano localmente i dati nella cache, in modo che la tua app sia in grado di leggere e scrivere i dati indipendentemente dallo stato di connettività del dispositivo. Quando il dispositivo è online, puoi sincronizzare i dati e, se configuri la sincronizzazione push, puoi avvisare immediatamente altri dispositivi della disponibilità di un aggiornamento.

Configurazione di un pool di identità in Amazon Cognito

Amazon Cognito Sync richiede un pool di identità Amazon Cognito per fornire le identità degli utenti. Prima di utilizzare Amazon Cognito Sync, dovrai prima configurare un pool di identità. Per creare un pool di identità e installare l'SDK, consulta Guida introduttiva ai pool di identità di Amazon Cognito.

Archiviazione e sincronizzazione dei dati

Dopo aver configurato il tuo pool di identità e installato l'SDK, puoi iniziare ad archiviare e a sincronizzare dati tra i dispositivi. Per ulteriori informazioni, consulta Sincronizzazione dei dati tra client.

Sincronizzazione dei dati tra client

> **Note**
>
> Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio per la sincronizzazione dei dati delle applicazioni tra dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito ti consente di salvare i dati dell'utente finale in set di dati contenenti coppie chiave-valore. Questi dati sono associati a un'identità di Amazon Cognito nel tuo pool di identità, in modo che sia possibile accedervi attraverso gli accessi e i dispositivi. Per sincronizzare i dati tra il servizio Amazon Cognito e i dispositivi di un utente finale, richiama il metodo di sincronizzazione. Ogni set di dati può avere una dimensione massima di 1 MB. Puoi associare fino a un massimo di 20 set di dati con un'identità.

Il client Amazon Cognito Sync crea una cache locale per i dati di identità. Quando legge e scrive chiavi, la tua app comunica con questa cache locale. Questo garantisce che tutte le modifiche che effettui sul dispositivo siano immediatamente disponibili sullo stesso, anche quando sei offline. Quando viene chiamato il metodo di sincronizzazione, le modifiche del servizio vengono trasferite al dispositivo e eventuali modifiche locali vengono trasferite al servizio. A questo punto, le modifiche sono disponibili per la sincronizzazione con altri dispositivi.

Inizializzazione del client di Amazon Cognito Sync

Per inizializzare il client di Amazon Cognito Sync, è necessario creare prima un fornitore di credenziali. Il fornitore di credenziali acquisisce AWS credenziali temporanee per consentire all'app di accedere alle risorse. AWS Devi anche importare i file di intestazione necessari. Utilizza la procedura seguente per inizializzare il client di Amazon Cognito Sync.

Android

1. Crea un fornitore di credenziali, seguendo le istruzioni in Ottenere le credenziali.
2. Importa il pacchetto Amazon Cognito come segue: `import com.amazonaws.mobileconnectors.cognito.*;`

3. Inizializza Amazon Cognito Sync. Passa il contesto dell'app Android, l'ID del pool di identità, una Regione AWS e un fornitore di credenziali Amazon Cognito inizializzato come segue:

```
CognitoSyncManager client = new CognitoSyncManager(
    getApplicationContext(),
    Regions.YOUR_REGION,
    credentialsProvider);
```

iOS - Objective-C

1. Crea un fornitore di credenziali, seguendo le istruzioni in Ottenere le credenziali.
2. Importa `AWSCore` e `Cognito` e inizializza `AWSCognito` come segue:

```
#import <AWSiOSSDKv2/AWSCore.h>
#import <AWSCognitoSync/Cognito.h>

AWSCognito *syncClient = [AWSCognito defaultCognito];
```

3. Se lo stai usando CocoaPods, sostituiscilo `<AWSiOSSDKv2/AWSCore.h>` con`AWSCore.h`. Segui la stessa sintassi per l'importazione di Amazon Cognito.

iOS - Swift

1. Crea un fornitore di credenziali, seguendo le istruzioni in Ottenere le credenziali.
2. Importa e inizializza `AWSCognito` come segue:

```
import AWSCognito
let syncClient = AWSCognito.default()!
```

JavaScript

1. Scarica Amazon Cognito Sync Manager per. JavaScript
2. Includi la libreria del manager della sincronizzazione nel tuo progetto;
3. Crea un fornitore di credenziali, seguendo le istruzioni in Ottenere le credenziali.
4. Inizializza il manager della sincronizzazione come segue:

```
var syncManager = new AWS.CognitoSyncManager();
```

Unità

1. Avrai bisogno inizialmente di creare un'istanza delle `CognitoAWSCredentials`, seguendo le istruzioni in Ottenere le credenziali.
2. Creare un'istanza di `CognitoSyncManager`. Passa l'oggetto delle `CognitoAwsCredentials` e un `AmazonCognitoSyncConfig` e includi almeno il set Regione, come segue:

```
AmazonCognitoSyncConfig clientConfig = new AmazonCognitoSyncConfig { RegionEndpoint =
 REGION };
CognitoSyncManager syncManager = new CognitoSyncManager(credentials, clientConfig);
```

Xamarin

1. Crea un'istanza delle `CognitoAWSCredentials`, seguendo le istruzioni in Ottenere le credenziali.
2. Creare un'istanza di `CognitoSyncManager`. Passa l'oggetto delle `CognitoAwsCredentials` e un `AmazonCognitoSyncConfig` e includi almeno il set Regione, come segue:

```
AmazonCognitoSyncConfig clientConfig = new AmazonCognitoSyncConfig { RegionEndpoint =
 REGION };
CognitoSyncManager syncManager = new CognitoSyncManager(credentials, clientConfig);
```

Comprendere i set di dati

Amazon Cognito organizza i dati del profilo dell'utente finale in set di dati. Ogni set di dati può contenere fino a 1 MB di dati sotto forma di coppie chiave-valore. Un set di dati è l'entità più granulare che puoi sincronizzare. Le operazioni di lettura e scrittura eseguite su un set di dati riguardano solo l'archivio locale fino a quando il metodo di sincronizzazione viene invocato. Amazon Cognito identifica un set di dati attraverso una stringa univoca. Puoi creare un nuovo set di dati oppure aprirne uno esistente nel modo seguente.

Android

```
Dataset dataset = client.openOrCreateDataset("datasetname");
```

Per eliminare un set di dati, per prima cosa chiama il metodo per rimuoverlo dallo spazio di archiviazione locale, quindi richiama il metodo `synchronize` per eliminare il set di dati da Amazon Cognito nel modo seguente:

```
dataset.delete();
dataset.synchronize(syncCallback);
```

iOS - Objective-C

```
AWSCognitoDataset *dataset = [syncClient openOrCreateDataset:@"myDataSet"];
```

Per eliminare un set di dati, per prima cosa chiama il metodo per rimuoverlo dallo spazio di archiviazione locale, quindi richiama il metodo `synchronize` per eliminare il set di dati da Amazon Cognito nel modo seguente:

```
[dataset clear];
[dataset synchronize];
```

iOS - Swift

```
let dataset = syncClient.openOrCreateDataset("myDataSet")!
```

Per eliminare un set di dati, per prima cosa chiama il metodo per rimuoverlo dallo spazio di archiviazione locale, quindi richiama il metodo `synchronize` per eliminare il set di dati da Amazon Cognito:

```
dataset.clear()
dataset.synchronize()
```

JavaScript

```
syncManager.openOrCreateDataset('myDatasetName', function(err, dataset) {
   // ...
});
```

Unità

```
string myValue = dataset.Get("myKey");
dataset.Put("myKey", "newValue");
```

Per eliminare una chiave dal set di dati, utilizza `Remove` come segue:

```
dataset.Remove("myKey");
```

Xamarin

```
Dataset dataset = syncManager.OpenOrCreateDataset("myDatasetName");
```

Per eliminare un set di dati, per prima cosa chiama il metodo per rimuoverlo dallo spazio di archiviazione locale, quindi richiama il metodo `synchronize` per eliminare il set di dati da Amazon Cognito nel modo seguente:

```
dataset.Delete();
dataset.SynchronizeAsync();
```

Lettura e scrittura dei dati nei set di dati

Funzione di set di dati di Amazon Cognito come dizionari, con i valori accessibili tramite chiave. Le chiavi e valori di un set di dati possono essere letti, aggiunti o modificati esattamente come se il set di dati fosse un dizionario, come mostrato negli esempi seguenti.

Nota che i valori che scrivi su un set di dati locali riguardano solo la copia memorizzata nella cache locale finché non chiami il metodo di sincronizzazione.

Android

```
String value = dataset.get("myKey");
dataset.put("myKey", "my value");
```

iOS - Objective-C

```
[dataset setString:@"my value" forKey:@"myKey"];
NSString *value = [dataset stringForKey:@"myKey"];
```

iOS - Swift

```
dataset.setString("my value", forKey:"myKey")
let value = dataset.stringForKey("myKey")
```

JavaScript

```
dataset.get('myKey', function(err, value) {
  console.log('myRecord: ' + value);
});

dataset.put('newKey', 'newValue', function(err, record) {
  console.log(record);
});

dataset.remove('oldKey', function(err, record) {
  console.log(success);
});
```

Unità

```
string myValue = dataset.Get("myKey");
dataset.Put("myKey", "newValue");
```

Xamarin

```
//obtain a value
string myValue = dataset.Get("myKey");

// Create a record in a dataset and synchronize with the server
dataset.OnSyncSuccess += SyncSuccessCallback;
dataset.Put("myKey", "myValue");
dataset.SynchronizeAsync();

void SyncSuccessCallback(object sender, SyncSuccessEventArgs e) {
  // Your handler code here
}
```

Android

Puoi utilizzare il metodo `remove` per rimuovere le chiavi da un set di dati come segue:

```
dataset.remove("myKey");
```

iOS - Objective-C

Per eliminare una chiave dal set di dati, utilizza `removeObjectForKey` come segue:

```
[dataset removeObjectForKey:@"myKey"];
```

iOS - Swift

Per eliminare una chiave dal set di dati, utilizza `removeObjectForKey` come segue:

```
dataset.removeObjectForKey("myKey")
```

Unità

Per eliminare una chiave dal set di dati, utilizza `Remove` come segue:

```
dataset.Remove("myKey");
```

Xamarin

Usa il comando `Remove` per eliminare una chiave dal set di dati:

```
dataset.Remove("myKey");
```

Sincronizzazione dei dati locali con lo store di sincronizzazione

Android

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

```
dataset.synchronize(syncCallback);
```

Il metodo `synchronize` riceve un'implementazione dell' interfaccia `SyncCallback`, illustrato di seguito.

Il metodo `synchronizeOnConnectivity()` tenta di sincronizzare quando è disponibile la connettività. Se la connettività è immediatamente disponibile, `synchronizeOnConnectivity()` si comporta come `synchronize()`. In caso contrario, monitora i cambiamenti di connettività ed esegue una sincronizzazione una volta che la connettività è disponibile. Se `synchronizeOnConnectivity()` viene chiamato più volte, viene mantenuta solo l'ultima richiesta di sincronizzazione e viene attivata solo l'ultima callback. Se il set di dati o la callback vengono sottoposti al processo di garbage collection, questo metodo non eseguirà una sincronizzazione e la callback non verrà attivata.

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback degli eventi.

iOS - Objective-C

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

Il metodo `synchronize` è asincrono e restituisce un oggetto `AWSTask` per gestire la risposta:

```
[[dataset synchronize] continueWithBlock:^id(AWSTask *task) {
    if (task.isCancelled) {
        // Task cancelled.
    } else if (task.error) {
        // Error while executing task.
    } else {
        // Task succeeded. The data was saved in the sync store.
    }
    return nil;
}];
```

Il metodo `synchronizeOnConnectivity` tenta di sincronizzare quando il dispositivo ha la connettività. In primo luogo, `synchronizeOnConnectivity` verifica la connettività e se il dispositivo è online immediatamente invoca la sincronizzazione e restituisce l'oggetto `AWSTask` associato al tentativo.

Se il dispositivo è offline, `synchronizeOnConnectivity` 1) pianifica un sincronizzare per la prossima volta che il dispositivo sarà online e 2) restituisce un `AWSTask` con risultato nullo. La sincronizzazione programmata è valida solo per il ciclo di vita dell'oggetto del set di dati. I dati non verranno sincronizzati se l'app viene chiusa prima che la connettività sia riconquistata. Se desideri essere avvisato ogni qualvolta hanno luogo eventi durante la sincronizzazione programmata, è necessario aggiungere osservatori delle notifiche trovati in `AWSCognito`.

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback degli eventi.

iOS - Swift

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

Il metodo `synchronize` è asincrono e restituisce un oggetto `AWSTask` per gestire la risposta:

```
dataset.synchronize().continueWith(block: { (task) -> AnyObject? in

            if task.isCancelled {
                // Task cancelled.
            } else if task.error != nil {
                // Error while executing task
            } else {
                // Task succeeded. The data was saved in the sync store.
            }
            return task
})
```

Il metodo `synchronizeOnConnectivity` tenta di sincronizzare quando il dispositivo ha la connettività. In primo luogo, `synchronizeOnConnectivity` verifica la connettività e se il dispositivo è online immediatamente invoca la `synchronize` e restituisce l'oggetto `AWSTask` associato al tentativo.

Se il dispositivo è offline, `synchronizeOnConnectivity` 1) pianifica un sincronizzare per la prossima volta che il dispositivo sarà online e 2) restituisce un oggetto `AWSTask` con un risultato nullo. La sincronizzazione programmata è valida solo per il ciclo di vita dell'oggetto del set di dati. I dati non verranno sincronizzati se l'app viene chiusa prima che la connettività sia riconquistata. Se

desideri essere avvisato ogni qualvolta hanno luogo eventi durante la sincronizzazione programmata, è necessario aggiungere osservatori delle notifiche trovati in `AWSCognito`.

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback degli eventi.

JavaScript

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

```
dataset.synchronize();
```

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback degli eventi.

Unità

Il metodo di sincronizzazione confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

```
dataset.Synchronize();
```

Synchronize viene eseguito in modo asincrono e finisce per chiamare uno dei numerosi callback che è possibile specificare nel Dataset.

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback degli eventi.

Xamarin

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati

sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

```
dataset.SynchronizeAsync();
```

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback degli eventi.

Gestione dei callback degli eventi

> **Note**
>
> Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio per la sincronizzazione dei dati delle applicazioni tra dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

In qualità di sviluppatore di Amazon Cognito Sync, puoi implementare diverse callback per gestire diversi eventi e scenari di sincronizzazione. L'`SyncCallback`interfaccia di Android SDK configura le notifiche sulla sincronizzazione dei set di dati, ad esempio `onSuccess()` quando un set di dati viene scaricato correttamente, quando si verifica un'eccezione e per `onFailure()` risolvere i conflitti tra dati locali e `onConflict()` remoti.

Nell'SDK iOS, puoi registrarti per notifiche simili `AWSCognitoDidStartSynchronizeNotification` e impostare gestori come quello `AWSCognitoRecordConflictHandler` per la risoluzione dei conflitti. Le JavaScript piattaforme Unity e Xamarin hanno meccanismi di callback analoghi. Quando implementi questi callback, l'applicazione può gestire in modo corretto i vari eventi e scenari di sincronizzazione che possono verificarsi quando si utilizza Amazon Cognito Sync.

Android

SyncCallback Interfaccia

Se implementi l'interfaccia `SyncCallback`, puoi ricevere notifiche sulla tua app riguardanti la sincronizzazione del set di dati. La tua app può quindi prendere decisioni attive sull'eliminazione

di dati locali, sull'unione di profili autenticati e non autenticati e la risoluzione di conflitti di sincronizzazione. È consigliabile implementare i seguenti metodi, che sono richiesti dall'interfaccia:

- `onSuccess()`
- `onFailure()`
- `onConflict()`
- `onDatasetDeleted()`
- `onDatasetsMerged()`

Nota che, se non desideri specificare tutti le callback, puoi anche utilizzare la classe `DefaultSyncCallback`, che fornisce implementazioni predefinite e vuote per ognuna di esse.

onSuccess

La callback `onSuccess()` viene attivata quando un set di dati è stato scaricato con successo dall'archivio di sincronizzazione.

```
@Override
public void onSuccess(Dataset dataset, List<Record> newRecords) {
}
```

onFailure

onFailure() viene chiamata se si verifica un'eccezione durante la sincronizzazione.

```
@Override
public void onFailure(DataStorageException dse) {
}
```

onConflict

Possono verificarsi dei conflitti se la stessa chiave è stata modificata nello store locale e in quello di sincronizzazione. Il metodo `onConflict()` gestisce la risoluzione dei conflitti. Se non implementi questo metodo, il client Amazon Cognito Sync utilizza per impostazione predefinita la modifica più recente.

```
@Override
public boolean onConflict(Dataset dataset, final List<SyncConflict> conflicts) {
    List<Record> resolvedRecords = new ArrayList<Record>();
```

```
    for (SyncConflict conflict : conflicts) {
        /* resolved by taking remote records */
        resolvedRecords.add(conflict.resolveWithRemoteRecord());

        /* alternately take the local records */
        // resolvedRecords.add(conflict.resolveWithLocalRecord());

        /* or customer logic, say concatenate strings */
        // String newValue = conflict.getRemoteRecord().getValue()
        //     + conflict.getLocalRecord().getValue();
        // resolvedRecords.add(conflict.resolveWithValue(newValue);
    }
    dataset.resolve(resolvedRecords);

    // return true so that synchronize() is retried after conflicts are resolved
    return true;
}
```

su DatasetDeleted

Quando un set di dati viene eliminato, il client Amazon Cognito usa l'interfaccia `SyncCallback` per confermare se la copia del set di dati memorizzata nella cache locale deve anch'essa essere eliminata. Implementa il metodo on`DatasetDeleted()` per comunicare all'SDK del client che cosa fare con i dati locali.

```
@Override
public boolean onDatasetDeleted(Dataset dataset, String datasetName) {
    // return true to delete the local copy of the dataset
    return true;
}
```

su DatasetMerged

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le applicazioni vengono avvisate dell'unione tramite il metodo on`DatasetsMerged()`:

```
@Override
public boolean onDatasetsMerged(Dataset dataset, List<String> datasetNames) {
    // return false to handle Dataset merge outside the synchronization callback
    return false;
}
```

iOS - Objective-C

Notifiche di sincronizzazione

Il client Amazon Cognito emette una serie di `NSNotification` eventi durante una chiamata di sincronizzazione. Puoi registrarti per monitorare queste notifiche attraverso lo standard `NSNotificationCenter`:

```
[NSNotificationCenter defaultCenter]
  addObserver:self
  selector:@selector(myNotificationHandler:)
  name:NOTIFICATION_TYPE
  object:nil];
```

Amazon Cognito supporta cinque tipi di notifica, elencati di seguito.

AWSCognitoDidStartSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione è in fase di avvio. `userInfo`Contiene il set di dati chiave, che è il nome del set di dati da sincronizzare.

AWSCognitoDidEndSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione viene completata (correttamente o meno). `userInfo`Contiene il set di dati chiave che è il nome del set di dati da sincronizzare.

AWSCognitoDidFailToSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione non va a buon fine. `userInfo`Contiene il set di dati chiave che è il nome del set di dati da sincronizzare e l'errore chiave che contiene l'errore che ha causato l'errore.

AWSCognitoDidChangeRemoteValueNotification

Chiamato quando le modifiche locali vengono inviate correttamente ad Amazon Cognito. `userInfo`Contiene il set di dati chiave che è il nome del set di dati da sincronizzare e le chiavi chiave che contengono un NSArray di chiavi di registrazione che sono state inserite.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Viene chiamato quando un valore locale si modifica a causa di un'operazione di sincronizzazione. `userInfo`Contiene il set di dati chiave che è il nome del set di dati da sincronizzare e le chiavi chiave che contengono un NSArray di chiavi di registrazione modificate.

Gestore della risoluzione del conflitto

Durante un'operazione di sincronizzazione, possono sorgere dei conflitti se la stessa chiave è stata modificata nell'archivio locale e in quello di sincronizzazione. Se non hai impostato un gestore della risoluzione dei conflitti, Amazon Cognito sceglierà di default l'aggiornamento più recente.

Implementando e assegnando un è AWSCognitoRecordConflictHandler possibile modificare la risoluzione predefinita dei conflitti. Il conflitto dei parametri AWSCognitoConflict di input contiene un AWSCognitoRecord oggetto sia per i dati memorizzati nella cache locale che per il record in conflitto nell'archivio di sincronizzazione. Utilizzando il AWSCognitoConflict è possibile risolvere il conflitto con il record locale: [risoluzione dei conflittiWithLocalRecord], il record remoto: [risoluzione dei conflittiWithRemoteRecord] o un valore nuovo di zecca: [risoluzione WithValue:value dei conflitti]. La restituzione di nil da questo metodo impedisce il proseguimento della sincronizzazione e i conflitti vengono ripresentati al successivo avvio del processo di sincronizzazione.

Puoi impostare il gestore della risoluzione del conflitto al livello del client:

```
client.conflictHandler = ^AWSCognitoResolvedConflict* (NSString *datasetName,
 AWSCognitoConflict *conflict) {
    // always choose local changes
    return [conflict resolveWithLocalRecord];
};
```

Oppure a livello del set di dati:

```
dataset.conflictHandler = ^AWSCognitoResolvedConflict* (NSString *datasetName,
 AWSCognitoConflict *conflict) {
    // override and always choose remote changes
    return [conflict resolveWithRemoteRecord];
};
```

Gestore dell'eliminazione del set di dati

Quando un set di dati viene eliminato, il client Amazon Cognito usa `AWSCognitoDatasetDeletedHandler` per confermare se la copia del set di dati memorizzata nella cache locale deve anch'essa essere eliminata. Se `AWSCognitoDatasetDeletedHandler` è implementato no, i dati locali vengono eliminati automaticamente. Implementa un `AWSCognitoDatasetDeletedHandler` se desideri mantenere una copia dei dati locali prima della cancellazione, o mantenere i dati locali.

Puoi impostare il gestore dell'eliminazione del set di dati a livello del client:

```
client.datasetDeletedHandler = ^BOOL (NSString *datasetName) {
    // make a backup of the data if you choose
    ...
    // delete the local data (default behavior)
    return YES;
};
```

Oppure a livello del set di dati:

```
dataset.datasetDeletedHandler = ^BOOL (NSString *datasetName) {
    // override default and keep the local data
    return NO;
};
```

Gestore dell'unione del set di dati

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le i vengono avvisate dell'unione tramite `DatasetMergeHandler`. Il gestore riceve il nome del set di dati radice e una matrice di nomi di set di dati contrassegnati come fusioni del set di dati radice.

Se non `DatasetMergeHandler` viene implementato, questi set di dati vengono ignorati, ma continuano a utilizzare lo spazio nei 20 set di dati totali massimi dell'identità.

Puoi impostare il gestore dell'unione del set di dati a livello del client:

```
client.datasetMergedHandler = ^(NSString *datasetName, NSArray *datasets) {
    // Blindly delete the datasets
    for (NSString *name in datasets) {
        AWSCognitoDataset *merged = [[AWSCognito defaultCognito]
 openOrCreateDataset:name];
       [merged clear];
       [merged synchronize];
    }
};
```

Oppure a livello del set di dati:

```
dataset.datasetMergedHandler = ^(NSString *datasetName, NSArray *datasets) {
    // Blindly delete the datasets
    for (NSString *name in datasets) {
```

```
        AWSCognitoDataset *merged = [[AWSCognito defaultCognito]
 openOrCreateDataset:name];
        // do something with the data if it differs from existing dataset
        ...
        // now delete it
        [merged clear];
        [merged synchronize];
    }
};
```

iOS - Swift

Notifiche di sincronizzazione

Il client Amazon Cognito emette una serie di `NSNotification` eventi durante una chiamata di sincronizzazione. Puoi registrarti per monitorare queste notifiche attraverso lo standard `NSNotificationCenter`:

```
NSNotificationCenter.defaultCenter().addObserver(observer: self,
   selector: "myNotificationHandler",
   name:NOTIFICATION_TYPE,
   object:nil)
```

Amazon Cognito supporta cinque tipi di notifica, elencati di seguito.

AWSCognitoDidStartSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione è in fase di avvio. `userInfo`Contiene il set di dati chiave, che è il nome del set di dati da sincronizzare.

AWSCognitoDidEndSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione viene completata (correttamente o meno). `userInfo`Contiene il set di dati chiave che è il nome del set di dati da sincronizzare.

AWSCognitoDidFailToSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione non va a buon fine. `userInfo`Contiene il set di dati chiave che è il nome del set di dati da sincronizzare e l'errore chiave che contiene l'errore che ha causato l'errore.

AWSCognitoDidChangeRemoteValueNotification

Chiamato quando le modifiche locali vengono inviate correttamente ad Amazon Cognito. `userInfo`Contiene il set di dati chiave che è il nome del set di dati da sincronizzare e le chiavi chiave che contengono un NSArray di chiavi di registrazione che sono state inserite.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Viene chiamato quando un valore locale si modifica a causa di un'operazione di sincronizzazione. `userInfo`Contiene il set di dati chiave che è il nome del set di dati da sincronizzare e le chiavi chiave che contengono un NSArray di chiavi di registrazione modificate.

Gestore della risoluzione del conflitto

Durante un'operazione di sincronizzazione, possono sorgere dei conflitti se la stessa chiave è stata modificata nell'archivio locale e in quello di sincronizzazione. Se non hai impostato un gestore della risoluzione dei conflitti, Amazon Cognito sceglierà per impostazione predefinita l'aggiornamento più recente.

Con l'implementazione e l'assegnazione di un `AWSCognitoRecordConflictHandler` puoi modificare la risoluzione dei conflitti predefiniti. Il conflitto del parametro di input `AWSCognitoConflict` contiene un oggetto `AWSCognitoRecord` sia per i dati memorizzati nella cache locale, sia per i record di conflitto nello store di sincronizzazione. Utilizzando `AWSCognitoConflict` è possibile risolvere il conflitto con il record locale: [conflict resolveWithLocalRecord], il record remoto: [conflict resolveWithRemoteRecord] o un nuovo valore: [conflict resolve]. WithValue:value La restituzione di nil da questo metodo impedisce il proseguimento della sincronizzazione e i conflitti vengono ripresentati al successivo avvio del processo di sincronizzazione.

Puoi impostare il gestore della risoluzione del conflitto al livello del client:

```
client.conflictHandler = {
    (datasetName: String?, conflict: AWSCognitoConflict?) ->
 AWSCognitoResolvedConflict? in
    return conflict.resolveWithLocalRecord()
}
```

Oppure a livello del set di dati:

```
dataset.conflictHandler = {
    (datasetName: String?, conflict: AWSCognitoConflict?) ->
 AWSCognitoResolvedConflict? in
    return conflict.resolveWithLocalRecord()
```

```
}
```

Gestore dell'eliminazione del set di dati

Quando un set di dati viene eliminato, il client Amazon Cognito usa `AWSCognitoDatasetDeletedHandler` per confermare se la copia del set di dati memorizzata nella cache locale deve anch'essa essere eliminata. Se `AWSCognitoDatasetDeletedHandler` è implementato no, i dati locali vengono eliminati automaticamente. Implementa un `AWSCognitoDatasetDeletedHandler` se desideri mantenere una copia dei dati locali prima della cancellazione, o mantenere i dati locali.

Puoi impostare il gestore dell'eliminazione del set di dati a livello del client:

```
client.datasetDeletedHandler = {
      (datasetName: String!) -> Bool in
      // make a backup of the data if you choose
      ...
      // delete the local data (default behaviour)
      return true
}
```

Oppure a livello del set di dati:

```
dataset.datasetDeletedHandler = {
      (datasetName: String!) -> Bool in
      // make a backup of the data if you choose
      ...
      // delete the local data (default behaviour)
      return true
}
```

Handler dell'unione dei set di dati

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le i vengono avvisate dell'unione tramite `DatasetMergeHandler`. Il gestore riceve il nome del set di dati radice e una matrice di nomi di set di dati contrassegnati come fusioni del set di dati radice.

Se non `DatasetMergeHandler` viene implementato, questi set di dati vengono ignorati, ma continuano a utilizzare lo spazio nei 20 set di dati totali massimi dell'identità.

Puoi impostare il gestore dell'unione del set di dati a livello del client:

```
client.datasetMergedHandler = {
    (datasetName: String!, datasets: [AnyObject]!) -> Void in
    for nameObject in datasets {
        if let name = nameObject as? String {
            let merged = AWSCognito.defaultCognito().openOrCreateDataset(name)
            merged.clear()
            merged.synchronize()
        }
    }
}
```

Oppure a livello del set di dati:

```
dataset.datasetMergedHandler = {
    (datasetName: String!, datasets: [AnyObject]!) -> Void in
    for nameObject in datasets {
        if let name = nameObject as? String {
            let merged = AWSCognito.defaultCognito().openOrCreateDataset(name)
            // do something with the data if it differs from existing dataset
              ...
              // now delete it
            merged.clear()
            merged.synchronize()
        }
    }
}
```

JavaScript

Callback di sincronizzazione

Quando esegui una sincronizzazione() su un set di dati, puoi, a tua discrezione, specificare le callback per gestire ciascuno dei seguenti stati:

```
dataset.synchronize({

   onSuccess: function(dataset, newRecords) {
      //...
   },

   onFailure: function(err) {
      //...
```

```
   },

   onConflict: function(dataset, conflicts, callback) {
      //...
   },

   onDatasetDeleted: function(dataset, datasetName, callback) {
      //...
   },

  onDatasetMerged: function(dataset, datasetNames, callback) {
      //...
  }

});
```

onSuccess()

La callback `onSuccess()` viene attivata quando un set di dati è stato aggiornato con successo dallo store di sincronizzazione. Se definisci una callback, la sincronizzazione riuscirà automaticamente.

```
onSuccess: function(dataset, newRecords) {
   console.log('Successfully synchronized ' + newRecords.length + ' new records.');
}
```

onFailure()

`onFailure()` viene chiamata se si verifica un'eccezione durante la sincronizzazione. Se non definisci una callback, la sincronizzazione fallirà automaticamente.

```
onFailure: function(err) {
   console.log('Synchronization failed.');
   console.log(err);
}
```

onConflict()

Possono verificarsi dei conflitti se la stessa chiave è stata modificata nello store locale e in quello di sincronizzazione. Il metodo `onConflict()` gestisce la risoluzione dei conflitti. Se non implementi questo metodo, la sincronizzazione verrà interrotta ogni volta che si verifica un conflitto.

```
onConflict: function(dataset, conflicts, callback) {
```

```
   var resolved = [];

   for (var i=0; i<conflicts.length; i++) {

      // Take remote version.
      resolved.push(conflicts[i].resolveWithRemoteRecord());

      // Or... take local version.
      // resolved.push(conflicts[i].resolveWithLocalRecord());

      // Or... use custom logic.
      // var newValue = conflicts[i].getRemoteRecord().getValue() +
 conflicts[i].getLocalRecord().getValue();
      // resolved.push(conflicts[i].resovleWithValue(newValue);

   }

   dataset.resolve(resolved, function() {
      return callback(true);
   });

   // Or... callback false to stop the synchronization process.
   // return callback(false);

}
```

su () DatasetDeleted

Quando un set di dati viene eliminato, il client Amazon Cognito usa la richiamata `onDatasetDeleted()` per decidere se anche la copia del set di dati memorizzato nella cache locale deve essere eliminata. Per impostazione predefinita, il set di dati non viene eliminato.

```
onDatasetDeleted: function(dataset, datasetName, callback) {

   // Return true to delete the local copy of the dataset.
   // Return false to handle deleted datasets outside the synchronization callback.

   return callback(true);

}
```

su DatasetMerged ()

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le applicazioni vengono avvisate dell'unione tramite la chiamata `onDatasetsMerged()`.

```
onDatasetMerged: function(dataset, datasetNames, callback) {

   // Return true to continue the synchronization process.
   // Return false to handle dataset merges outside the synchronization callback.

   return callback(false);

}
```

Unità

Dopo aver aperto o creato un set di dati, potete impostarne diversi callback che vengono attivati quando utilizzate il metodo Synchronize. Questo è il modo per registrarvi le tue callback:

```
dataset.OnSyncSuccess += this.HandleSyncSuccess;
dataset.OnSyncFailure += this.HandleSyncFailure;
dataset.OnSyncConflict = this.HandleSyncConflict;
dataset.OnDatasetMerged = this.HandleDatasetMerged;
dataset.OnDatasetDeleted = this.HandleDatasetDeleted;
```

Nota che `SyncSuccess` e `SyncFailure` usano += invece di = pertanto puoi sottoscrivervi più di una callback.

OnSyncSuccess

La callback `OnSyncSuccess` viene attivata quando un set di dati è stato aggiornato correttamente dal cloud. Se definisci una callback, la sincronizzazione riuscirà automaticamente.

```
private void HandleSyncSuccess(object sender, SyncSuccessEvent e)
{
    // Continue with your game flow, display the loaded data, etc.
}
```

OnSyncFailure

`OnSyncFailure` viene chiamata se si verifica un'eccezione durante la sincronizzazione. Se non definisci una callback, la sincronizzazione fallirà automaticamente.

```
private void HandleSyncFailure(object sender, SyncFailureEvent e)
{
    Dataset dataset = sender as Dataset;
    if (dataset.Metadata != null) {
        Debug.Log("Sync failed for dataset : " + dataset.Metadata.DatasetName);
    } else {
        Debug.Log("Sync failed");
    }
    // Handle the error
    Debug.LogException(e.Exception);
}
```

OnSyncConflict

Possono verificarsi dei conflitti se la stessa chiave è stata modificata nello store locale e in quello di sincronizzazione. La callback `OnSyncConflict` gestisce la risoluzione dei conflitti. Se non implementi questo metodo, la sincronizzazione verrà interrotta ogni volta che si verifica un conflitto.

```
private bool HandleSyncConflict(Dataset dataset, List < SyncConflict > conflicts)
{
  if (dataset.Metadata != null) {
    Debug.LogWarning("Sync conflict " + dataset.Metadata.DatasetName);
  } else {
    Debug.LogWarning("Sync conflict");
  }
  List < Amazon.CognitoSync.SyncManager.Record > resolvedRecords = new List <
 Amazon.CognitoSync.SyncManager.Record > ();
  foreach(SyncConflict conflictRecord in conflicts) {
    // SyncManager provides the following default conflict resolution methods:
    //      ResolveWithRemoteRecord - overwrites the local with remote records
    //      ResolveWithLocalRecord - overwrites the remote with local records
    //      ResolveWithValue - to implement your own logic
    resolvedRecords.Add(conflictRecord.ResolveWithRemoteRecord());
  }
  // resolves the conflicts in local storage
  dataset.Resolve(resolvedRecords);
  // on return true the synchronize operation continues where it left,
  //      returning false cancels the synchronize operation
  return true;
}
```

OnDatasetDeleted

Quando un set di dati viene eliminato, il client Amazon Cognito usa la richiamata `OnDatasetDeleted` per decidere se anche la copia del set di dati memorizzato nella cache locale deve essere eliminata. Per impostazione predefinita, il set di dati non viene eliminato.

```
private bool HandleDatasetDeleted(Dataset dataset)
  {
      Debug.Log(dataset.Metadata.DatasetName + " Dataset has been deleted");
      // Do clean up if necessary
      // returning true informs the corresponding dataset can be purged in the local
 storage and return false retains the local dataset
      return true;
  }
```

OnDatasetMerged

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le applicazioni vengono avvisate dell'unione tramite la chiamata `OnDatasetsMerged`.

```
public bool HandleDatasetMerged(Dataset localDataset, List<string> mergedDatasetNames)
{
    foreach (string name in mergedDatasetNames)
    {
        Dataset mergedDataset = syncManager.OpenOrCreateDataset(name);
        //Lambda function to delete the dataset after fetching it
        EventHandler<SyncSuccessEvent> lambda;
        lambda = (object sender, SyncSuccessEvent e) => {
            ICollection<string> existingValues = localDataset.GetAll().Values;
            ICollection<string> newValues = mergedDataset.GetAll().Values;

            //Implement your merge logic here

            mergedDataset.Delete(); //Delete the dataset locally
            mergedDataset.OnSyncSuccess -= lambda; //We don't want this callback to be
 fired again
            mergedDataset.OnSyncSuccess += (object s2, SyncSuccessEvent e2) => {
                localDataset.Synchronize(); //Continue the sync operation that was
 interrupted by the merge
            };
            mergedDataset.Synchronize(); //Synchronize it as deleted, failing to do so
 will leave us in an inconsistent state
        };
        mergedDataset.OnSyncSuccess += lambda;
        mergedDataset.Synchronize(); //Asnchronously fetch the dataset
```

```
    }

    // returning true allows the Synchronize to continue and false stops it
    return false;
}
```

Xamarin

Dopo aver aperto o creato un set di dati, è possibile impostarne diversi callback che vengono attivati quando si utilizza il metodo Synchronize. Questo è il modo per registrarvi le tue callback:

```
dataset.OnSyncSuccess += this.HandleSyncSuccess;
dataset.OnSyncFailure += this.HandleSyncFailure;
dataset.OnSyncConflict = this.HandleSyncConflict;
dataset.OnDatasetMerged = this.HandleDatasetMerged;
dataset.OnDatasetDeleted = this.HandleDatasetDeleted;
```

Nota che `SyncSuccess` e `SyncFailure` usano += invece di = pertanto puoi sottoscrivervi più di una callback.

OnSyncSuccess

La callback `OnSyncSuccess` viene attivata quando un set di dati è stato aggiornato correttamente dal cloud. Se definisci una callback, la sincronizzazione riuscirà automaticamente.

```
private void HandleSyncSuccess(object sender, SyncSuccessEventArgs e)
{
    // Continue with your game flow, display the loaded data, etc.
}
```

OnSyncFailure

`OnSyncFailure` viene chiamata se si verifica un'eccezione durante la sincronizzazione. Se non definisci una callback, la sincronizzazione fallirà automaticamente.

```
private void HandleSyncFailure(object sender, SyncFailureEventArgs e)
{
    Dataset dataset = sender as Dataset;
    if (dataset.Metadata != null) {
        Console.WriteLine("Sync failed for dataset : " + dataset.Metadata.DatasetName);
    } else {
        Console.WriteLine("Sync failed");
```

```
    }
}
```

OnSyncConflict

Possono verificarsi dei conflitti se la stessa chiave è stata modificata nello store locale e in quello di sincronizzazione. La callback `OnSyncConflict` gestisce la risoluzione dei conflitti. Se non implementi questo metodo, la sincronizzazione verrà interrotta ogni volta che si verifica un conflitto.

```
private bool HandleSyncConflict(Dataset dataset, List < SyncConflict > conflicts)
{
  if (dataset.Metadata != null) {
    Console.WriteLine("Sync conflict " + dataset.Metadata.DatasetName);
  } else {
    Console.WriteLine("Sync conflict");
  }
  List < Amazon.CognitoSync.SyncManager.Record > resolvedRecords = new List <
 Amazon.CognitoSync.SyncManager.Record > ();
  foreach(SyncConflict conflictRecord in conflicts) {
    // SyncManager provides the following default conflict resolution methods:
    //      ResolveWithRemoteRecord - overwrites the local with remote records
    //      ResolveWithLocalRecord - overwrites the remote with local records
    //      ResolveWithValue - to implement your own logic
    resolvedRecords.Add(conflictRecord.ResolveWithRemoteRecord());
  }
  // resolves the conflicts in local storage
  dataset.Resolve(resolvedRecords);
  // on return true the synchronize operation continues where it left,
  //      returning false cancels the synchronize operation
  return true;
}
```

OnDatasetDeleted

Quando un set di dati viene eliminato, il client Amazon Cognito usa la richiamata `OnDatasetDeleted` per decidere se anche la copia del set di dati memorizzato nella cache locale deve essere eliminata. Per impostazione predefinita, il set di dati non viene eliminato.

```
private bool HandleDatasetDeleted(Dataset dataset)
{
    Console.WriteLine(dataset.Metadata.DatasetName + " Dataset has been deleted");
    // Do clean up if necessary
```

```
    // returning true informs the corresponding dataset can be purged in the local
 storage and return false retains the local dataset
    return true;
}
```

OnDatasetMerged

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le applicazioni vengono avvisate dell'unione tramite la chiamata `OnDatasetsMerged`.

```
public bool HandleDatasetMerged(Dataset localDataset, List<string> mergedDatasetNames)
{
    foreach (string name in mergedDatasetNames)
    {
        Dataset mergedDataset = syncManager.OpenOrCreateDataset(name);

            //Implement your merge logic here

        mergedDataset.OnSyncSuccess += lambda;
        mergedDataset.SynchronizeAsync(); //Asnchronously fetch the dataset
    }

    // returning true allows the Synchronize to continue and false stops it
    return false;
}
```

Implementazione della sincronizzazione push

> **Note**
>
> Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio per la sincronizzazione dei dati delle applicazioni tra dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito monitora automaticamente l'associazione tra identità e dispositivi. L'uso della funzione della sincronizzazione push ti permette di assicurare che tutte le istanze di una determinata

identità siano avvisate della modifica di alcuni dati di identità. La sincronizzazione push garantisce che, ogni volta che i dati dell'archivio di sincronizzazione cambiano per una determinata identità, tutti i dispositivi associati a tale identità ricevono una notifica push silenziosa che li informa delle modifiche.

> **Note**
>
> La sincronizzazione push non è supportata per JavaScript Unity o Xamarin.

Prima di poter utilizzare la sincronizzazione push, devi abilitarla nella console Amazon Cognito e configurare l'account per la sincronizzazione.

Creazione di un'app Amazon Simple Notification Service (Amazon SNS)

Crea e configura un'app Amazon SNS per la tua piattaforma supportata, come descritto nella Guida per gli sviluppatori di SNS.

Abilitazione della sincronizzazione push nella console di Amazon Cognito

Puoi abilitare la sincronizzazione push tramite la console Amazon Cognito. Dalla home page della console:

1. Fai clic sul nome del pool di identità per cui desideri attivare la sincronizzazione push. Viene visualizzata la pagina Dashboard (Pannello di controllo) per il tuo pool di identità.
2. Nell'angolo in alto a destra della pagina Dashboard (Pannello di controllo), fai clic su Manage Identity Pools (Gestisci pool di identità). Viene visualizzata la pagina Federated Identities (Identità federate).
3. Scorri verso il basso e fai clic su Push synchronization (Sincronizzazione push) per ingrandire questa opzione.
4. Nel menu a discesa Service role (Ruolo di servizio) seleziona il ruolo IAM che garantisce a Cognito l'autorizzazione per inviare una notifica SNS. Fai clic su Create role (Crea ruolo) per creare o modificare i ruoli associati al tuo pool di identità nella Console AWS IAM.
5. Seleziona un'applicazione della piattaforma, quindi fai clic su Save Changes (Salva le modifiche).
6. Concedi l'accesso SNS alla tua applicazione

Nella AWS Identity and Access Management console, configura i tuoi ruoli IAM per avere accesso completo ad Amazon SNS o crea un nuovo ruolo con accesso completo ad Amazon SNS. L'esempio

seguente di policy di attendibilità dei ruoli garantisce ad Amazon Cognito Sync una capacità limitata di assumere un ruolo IAM. Amazon Cognito Sync può assumere il ruolo solo quando lo fa per conto sia del pool di identità nella condizione `aws:SourceArn` che dell'account nella condizione `aws:SourceAccount`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cognito-sync.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "AWS:SourceArn": "arn:aws:cognito-identity:us-
east-1:123456789012:identitypool/us-east-1:177a950c-2c08-43f0-9983-28727EXAMPLE"
                }
            }
        }
    ]
}
```

Per ulteriori informazioni sui ruoli IAM, consulta i Ruoli (Delega e Federazione).

Uso della sincronizzazione push nella tua app: Android

La tua applicazione deve importare i servizi di Google Play. Puoi scaricare la versione più recente di SDK di Google Play tramite il gestore SDK di Android. Segui la documentazione di Android suAndroid Implementation per registrare la tua app e ricevere un ID di registrazione da GCM. Una volta che hai ottenuto l'ID di registrazione, hai bisogno di registrare il dispositivo con Amazon Cognito come illustrato nel frammento che segue:

```
String registrationId = "MY_GCM_REGISTRATION_ID";
```

```
try {
    client.registerDevice("GCM", registrationId);
} catch (RegistrationFailedException rfe) {
    Log.e(TAG, "Failed to register device for silent sync", rfe);
} catch (AmazonClientException ace) {
    Log.e(TAG, "An unknown error caused registration for silent sync to fail", ace);
}
```

Ora puoi sottoscrivere un dispositivo per ricevere gli aggiornamenti provenienti da un determinato set di dati:

```
Dataset trackedDataset = client.openOrCreateDataset("myDataset");
if (client.isDeviceRegistered()) {
    try {
        trackedDataset.subscribe();
    } catch (SubscribeFailedException sfe) {
        Log.e(TAG, "Failed to subscribe to datasets", sfe);
    } catch (AmazonClientException ace) {
        Log.e(TAG, "An unknown error caused the subscription to fail", ace);
    }
}
```

Per interrompere la ricezione di notifiche push da un set di dati, è sufficiente chiamare il metodo di annullamento. Per effettuare la sotttoscrizione a tutti i set di dati (o un sottoinsieme specifico) nell'oggetto `CognitoSyncManager`, usa `subscribeAll()`:

```
if (client.isDeviceRegistered()) {
    try {
        client.subscribeAll();
    } catch (SubscribeFailedException sfe) {
        Log.e(TAG, "Failed to subscribe to datasets", sfe);
    } catch (AmazonClientException ace) {
        Log.e(TAG, "An unknown error caused the subscription to fail", ace);
    }
}
```

Nell'implementazione dell' BroadcastReceiveroggetto Android, puoi controllare la versione più recente del set di dati modificato e decidere se l'app deve essere nuovamente sincronizzata:

```
@Override
public void onReceive(Context context, Intent intent) {
```

```
    PushSyncUpdate update = client.getPushSyncUpdate(intent);

    // The update has the source (cognito-sync here), identityId of the
    // user, identityPoolId in question, the non-local sync count of the
    // data set and the name of the dataset. All are accessible through
    // relevant getters.

    String source = update.getSource();
    String identityPoolId = update.getIdentityPoolId();
    String identityId = update.getIdentityId();
    String datasetName = update.getDatasetName();
    long syncCount = update.getSyncCount();

    Dataset dataset = client.openOrCreateDataset(datasetName);

    // need to access last sync count. If sync count is less or equal to
    // last sync count of the dataset, no sync is required.

    long lastSyncCount = dataset.getLastSyncCount();
    if (lastSyncCount < syncCount) {
        dataset.synchronize(new SyncCallback() {
            // ...
        });
    }

}
```

Le seguenti chiavi sono disponibili nel payload delle notifiche push:

- `source`: cognito-sync. Questo può servire come un fattore di differenziazione tra le notifiche.
- `identityPoolId`: ID pool di identità. Questo può essere utilizzato per la convalida o per informazioni aggiuntive, ma non è integrante dal punto di vista del ricevitore.
- `identityId`: ID identità all'interno del pool.
- `datasetName`: Nome del set di dati che è stato aggiornato. Questo è disponibile ai fini del bando apertoOrCreateDataset .
- `syncCount`: Numero di sincronizzazione per il set di dati remoto. Puoi usare questo come un modo per assicurarti che il set di dati locali sia obsoleto e che la sincronizzazione in entrata sia nuova.

Usa la sincronizzazione push nella tua app: iOS - Objective-C

Per ottenere un token di dispositivo per la tua app, segui la documentazione Apple sulla Registrazione per notifiche remote. Una volta che hai ricevuto il token di dispositivo come un oggetto NSData dagli APN, è necessario registrare il dispositivo con Amazon Cognito utilizzando il metodo `registerDevice:` del client di sincronizzazione, come illustrato di seguito:

```
AWSCognito *syncClient = [AWSCognito defaultCognito];
    [[syncClient registerDevice: devToken] continueWithBlock:^id(AWSTask *task) {
        if(task.error){
            NSLog(@"Unable to registerDevice: %@", task.error);
        } else {
            NSLog(@"Successfully registered device with id: %@", task.result);
        }
        return nil;
      }
    ];
```

In modalità debug, il dispositivo si registra nella sandbox APNs; in modalità release, si registra con gli APN. Per ricevere aggiornamenti da un particolare set di dati, usa il metodo `subscribe`:

```
[[[syncClient openOrCreateDataset:@"MyDataset"] subscribe]
 continueWithBlock:^id(AWSTask *task) {
        if(task.error){
            NSLog(@"Unable to subscribe to dataset: %@", task.error);
        } else {
            NSLog(@"Successfully subscribed to dataset: %@", task.result);
        }
        return nil;
      }
    ];
```

Per interrompere la ricezione di notifiche push da un set di dati, è sufficiente chiamare il metodo `unsubscribe`:

```
[[[syncClient openOrCreateDataset:@”MyDataset”] unsubscribe]
 continueWithBlock:^id(AWSTask *task) {
        if(task.error){
            NSLog(@"Unable to unsubscribe from dataset: %@", task.error);
        } else {
            NSLog(@"Successfully unsubscribed from dataset: %@", task.result);
        }
```

```
        return nil;
      }
    ];
```

Per effettuare la sottoscrizione a tutti i set di dati nell'oggetto `AWSCognito`, chiama `subscribeAll`:

```
[[syncClient subscribeAll] continueWithBlock:^id(AWSTask *task) {
        if(task.error){
            NSLog(@"Unable to subscribe to all datasets: %@", task.error);
        } else {
            NSLog(@"Successfully subscribed to all datasets: %@", task.result);
        }
        return nil;
      }
    ];
```

Prima di chiamare `subscribeAll`, assicurati di sincronizzare tutti i set di dati almeno una volta, in modo tale che gli stessi set di dati esistano nel server.

Per rispondere alle notifiche push, hai bisogno di implementare il metodo `didReceiveRemoteNotification` nel tuo delegato dell'app:

```
- (void)application:(UIApplication *)application didReceiveRemoteNotification:
(NSDictionary *)userInfo
    {
        [[NSNotificationCenter defaultCenter]
 postNotificationName:@"CognitoPushNotification" object:userInfo];
    }
```

Se pubblichi una notizia tramite il gestore delle notifiche, puoi quindi rispondere alla notifica in qualsiasi punto dell'applicazione in cui disponi di un gestore per il set di dati. Se effettui la sottoscrizione alla notifica in questo modo ...

```
[[NSNotificationCenter defaultCenter] addObserver:self
 selector:@selector(didReceivePushSync:)
    name: :@"CognitoPushNotification" object:nil];
```

... puoi intervenire sulla notifica in questo modo:

```
- (void)didReceivePushSync:(NSNotification*)notification
    {
```

```
        NSDictionary * data = [(NSDictionary *)[notification object]
 objectForKey:@"data"];
        NSString * identityId = [data objectForKey:@"identityId"];
        NSString * datasetName = [data objectForKey:@"datasetName"];
        if([self.dataset.name isEqualToString:datasetName] && [self.identityId
 isEqualToString:identityId]){
            [[self.dataset synchronize] continueWithBlock:^id(AWSTask *task) {
                if(!task.error){
                    NSLog(@"Successfully synced dataset");
                }
                return nil;
            }];
        }
    }
```

Le seguenti chiavi sono disponibili nel payload delle notifiche push:

- `source`: cognito-sync. Questo può servire come un fattore di differenziazione tra le notifiche.
- `identityPoolId`: ID pool di identità. Questo può essere utilizzato per la convalida o per informazioni aggiuntive, ma non è integrante dal punto di vista del ricevitore.
- `identityId`: ID identità all'interno del pool.
- `datasetName`: Nome del set di dati che è stato aggiornato. Questo è disponibile per la sicurezza della chiamata `openOrCreateDataset`.
- `syncCount`: Numero di sincronizzazione per il set di dati remoto. Puoi usare questo come un modo per assicurarti che il set di dati locali sia obsoleto e che la sincronizzazione in entrata sia nuova.

Uso della sincronizzazione push nella tua app: iOS - Swift

Per ottenere un token di dispositivo per la tua app, segui la documentazione Apple sulla Registrazione per notifiche remote. Una volta che hai ricevuto il token di dispositivo come un oggetto NSData dagli APN, è necessario registrare il dispositivo con Amazon Cognito utilizzando il registerDevice: del client di sincronizzazione, come illustrato di seguito:

```
let syncClient = AWSCognito.default()
syncClient.registerDevice(devToken).continueWith(block: { (task: AWSTask!) ->
 AnyObject! in
    if (task.error != nil) {
        print("Unable to register device: " + task.error.localizedDescription)
```

```
    } else {
        print("Successfully registered device with id: \(task.result)")
    }
    return task
})
```

In modalità di debug, il dispositivo si registra nella sandbox APNs; in modalità di rilascio, si registra con gli APN. Per ricevere aggiornamenti da un particolare set di dati, usa il metodo `subscribe`:

```
syncClient.openOrCreateDataset("MyDataset").subscribe().continueWith(block: { (task:
 AWSTask!) -> AnyObject! in
  if (task.error != nil) {
      print("Unable to subscribe to dataset: " + task.error.localizedDescription)

  } else {
      print("Successfully subscribed to dataset: \(task.result)")
  }
  return task
})
```

Per interrompere la ricezione di notifiche push da un set di dati, chiama il metodo `unsubscribe`:

```
syncClient.openOrCreateDataset("MyDataset").unsubscribe().continueWith(block: { (task:
 AWSTask!) -> AnyObject! in
  if (task.error != nil) {
      print("Unable to unsubscribe to dataset: " + task.error.localizedDescription)

  } else {
      print("Successfully unsubscribed to dataset: \(task.result)")
  }
  return task
})
```

Per effettuare la sottoscrizione a tutti i set di dati nell'oggetto `AWSCognito`, chiama `subscribeAll`:

```
syncClient.openOrCreateDataset("MyDataset").subscribeAll().continueWith(block: { (task:
 AWSTask!) -> AnyObject! in
  if (task.error != nil) {
      print("Unable to subscribe to all datasets: " + task.error.localizedDescription)

  } else {
```

```
      print("Successfully subscribed to all datasets: \(task.result)")
  }
  return task
})
```

Prima di chiamare `subscribeAll`, assicurati di sincronizzare tutti i set di dati almeno una volta, in modo tale che gli stessi set di dati esistano nel server.

Per rispondere alle notifiche push, hai bisogno di implementare il metodo `didReceiveRemoteNotification` nel tuo delegato dell'app:

```
func application(application: UIApplication, didReceiveRemoteNotification userInfo:
 [NSObject : AnyObject],
  fetchCompletionHandler completionHandler: (UIBackgroundFetchResult) -> Void) {

 NSNotificationCenter.defaultCenter().postNotificationName("CognitoPushNotification",
 object: userInfo)
})
```

Se pubblichi una notizia tramite il gestore delle notifiche, puoi quindi rispondere alla notifica in qualsiasi punto dell'applicazione in cui disponi di un gestore per il set di dati. Se effettui la sottoscrizione alla notifica in questo modo ...

```
NSNotificationCenter.defaultCenter().addObserver(observer:self,
   selector:"didReceivePushSync:",
   name:"CognitoPushNotification",
   object:nil)
```

... puoi intervenire sulla notifica in questo modo:

```
func didReceivePushSync(notification: NSNotification) {
    if let data = (notification.object as! [String: AnyObject])["data"] as? [String:
 AnyObject] {
       let identityId = data["identityId"] as! String
       let datasetName = data["datasetName"] as! String

       if self.dataset.name == datasetName && self.identityId == identityId {
         dataset.synchronize().continueWithBlock {(task) -> AnyObject! in
             if task.error == nil {
               print("Successfully synced dataset")
             }
             return nil
```

```
            }
        }
    }
}
```

Le seguenti chiavi sono disponibili nel payload delle notifiche push:

- `source`: cognito-sync. Questo può servire come un fattore di differenziazione tra le notifiche.
- `identityPoolId`: ID pool di identità. Questo può essere utilizzato per la convalida o per informazioni aggiuntive, ma non è integrante dal punto di vista del ricevitore.
- `identityId`: ID identità all'interno del pool.
- `datasetName`: Nome del set di dati che è stato aggiornato. Questo è disponibile per la sicurezza della chiamata `openOrCreateDataset`.
- `syncCount`: Numero di sincronizzazione per il set di dati remoto. Puoi usare questo come un modo per assicurarti che il set di dati locali sia obsoleto e che la sincronizzazione in entrata sia nuova.

Implementazione dei flussi di Amazon Cognito Sync

> **Note**
>
> Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio per la sincronizzazione dei dati delle applicazioni tra dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito Streams offre agli sviluppatori il controllo e l'analisi dei loro dati archiviati in Amazon Cognito. Gli sviluppatori possono ora configurare un flusso Kinesis per ricevere eventi non appena i dati sono aggiornati e sincronizzati. Amazon Cognito può eseguire il push di ogni modifica del set di dati in un flusso Kinesis in tempo reale.

Utilizzando Amazon Cognito Streams, puoi spostare tutti i tuoi dati di Sync in Kinesis, che possono quindi essere distribuiti in uno strumento data warehouse come Amazon Redshift per un'ulteriore analisi. Per ulteriori informazioni su Kinesis, consulta Nozioni di base per l'uso di Amazon Kinesis.

Configurazione dei flussi

Puoi configurare Amazon Cognito Streams nella console Amazon Cognito. Per abilitare Amazon Cognito Streams nella console Amazon Cognito, devi selezionare il flusso Kinesis in cui pubblicare e un ruolo IAM, che concede l'autorizzazione di Amazon Cognito per inserire gli eventi nel flusso selezionato.

Dalla home page della console:

1. Fai clic sul nome del pool di identità per cui desideri configurare Amazon Cognito Streams. Viene visualizzata la pagina Dashboard (Pannello di controllo) per il tuo pool di identità.
2. Nell'angolo in alto a destra della pagina Dashboard (Pannello di controllo), fai clic su Manage Identity Pools (Gestisci pool di identità). Viene visualizzata la pagina di gestione delle identità federate.
3. Scorri verso il basso e fai clic su Cognito Streams (Flussi di Cognito) per espandere l'operazione.
4. Nel menu a discesa Stream name (Nome del flusso), seleziona il nome di un flusso Kinesis esistente. In alternativa, fai clic su Create stream (Crea flusso) per crearne uno, immettendo un nome di flusso e il numero di shard. Per ulteriori informazioni sugli shard e per valutare il numero di shard di cui hai bisogno per il tuo flusso, consulta la Guida per gli sviluppatori di Kinesis.
5. Nel menu a discesa Publish role (Pubblica ruolo), seleziona il ruolo IAM che garantisce l'autorizzazione di Amazon Cognito per pubblicare il tuo flusso. Fai clic su Create role (Crea ruolo) per creare o modificare i ruoli associati al tuo pool di identità nella Console AWS IAM.
6. Nel menu a discesa Stream status (Stato del flusso) seleziona Enabled (Abilitato) per abilitare gli aggiornamenti del flusso. Fai clic su Save Changes (Salva modifiche).

Dopo aver configurato correttamente gli stream di Amazon Cognito, tutti gli aggiornamenti successivi ai set di dati in questo pool di identità vengono inviati allo stream.

Contenuti del flusso

Ogni record inviato al flusso rappresenta una singola sincronizzazione. Ecco l'esempio di un record inviato al flusso:

```
{
    "identityPoolId": "Pool Id",
    "identityId": "Identity Id",
    "dataSetName": "Dataset Name",
```

```
    "operation": "(replace|remove)",
    "kinesisSyncRecords": [
        {
            "key": "Key",
            "value": "Value",
            "syncCount": 1,
            "lastModifiedDate": 1424801824343,
            "deviceLastModifiedDate": 1424801824343,
            "op": "(replace|remove)"
        },
        ...
    ],
    "lastModifiedDate": 1424801824343,
    "kinesisSyncRecordsURL": "S3Url",
    "payloadType": "(S3Url|Inline)",
    "syncCount": 1
}
```

Per gli aggiornamenti di dimensioni superiori alla dimensione massima di payload di Kinesis, ovvero 1 MB, Amazon Cognito include un URL Amazon S3 prefirmato con il contenuto completo dell'aggiornamento.

Dopo aver configurato i flussi di Amazon Cognito, se elimini il flusso Kinesis o modifichi l'autorizzazione di attendibilità del ruolo in modo che non possa più essere assunto da Amazon Cognito Sync, disabiliterai i flussi di Amazon Cognito. Devi creare nuovamente il flusso Kinesis o correggere il ruolo e quindi riattivare il flusso.

Pubblicazione in blocco

Una volta che hai configurato i flussi di Amazon Cognito potrai eseguire un'operazione di pubblicazione in blocco per i dati esistenti nel tuo pool di identità. Dopo aver avviato un'operazione di pubblicazione in blocco, tramite la console o direttamente tramite l'API, Amazon Cognito inizia a pubblicare questi dati nello stesso flusso che riceve gli aggiornamenti.

Amazon Cognito non garantisce l'univocità dei dati inviati al flusso quando si utilizza l'operazione di pubblicazione in blocco. Potresti ricevere lo stesso aggiornamento sia come un aggiornamento, sia come parte di una pubblicazione in blocco. Tieni presente questa possibilità durante l'elaborazione del record proveniente dal tuo flusso.

Per pubblicare in blocco tutti i tuoi flussi, segui i passaggi 1-6 della sezione di configurazione dei flussi e quindi fai clic su Start bulk publish (Inizia la pubblicazione in blocco). Hai un limite per

un'operazione di pubblicazione in blocco in corso in qualsiasi momento e per una richiesta di pubblicazione in blocco riuscita ogni 24 ore.

Personalizzazione dei flussi di lavoro con Amazon Cognito Events

> **Note**
>
> Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio per la sincronizzazione dei dati delle applicazioni tra dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito Events ti consente di eseguire una AWS Lambda funzione in risposta a eventi importanti in Amazon Cognito. Amazon Cognito lancia l'evento trigger di sincronizzazione quando viene sincronizzato un set di dati. Puoi utilizzare l'evento del trigger di sincronizzazione per eseguire un'azione quando un utente aggiorna i dati. La funzione è in grado di valutare e, facoltativamente, modificare i dati prima che siano archiviati nel cloud e sincronizzati negli altri dispositivi dell'utente. Questa funzione è utile per convalidare i dati provenienti dal dispositivo prima che siano sincronizzati negli altri dispositivi dell'utente o per aggiornare altri valori nel set di dati in base ai dati in entrata, come l'attribuzione di un riconoscimento quando un giocatore raggiunge un nuovo livello.

I passaggi seguenti ti guidano nella configurazione di una funzione Lambda che viene eseguita ogni volta che un set di dati Amazon Cognito viene sincronizzato.

> **Note**
>
> Quando utilizzi gli eventi di Amazon Cognito, puoi usare solo le credenziali ottenute da identità Amazon Cognito. Se hai una funzione Lambda associata, ma chiami `UpdateRecords` con le credenziali AWS dell'account (credenziali dello sviluppatore), la tua funzione Lambda non verrà richiamata.

Creazione di una funzione in AWS Lambda

Per integrare Lambda con Amazon Cognito, per prima cosa hai bisogno di creare una funzione in Lambda. A tale scopo:

Selezione della funzione Lambda in Amazon Cognito

1. Apri la console Lambda.
2. Fai clic su Create a Lambda function (Crea una funzione Lambda).
3. Nella schermata Select blueprint (Seleziona piano), cerca e seleziona "cognito-sync-trigger" ("trigger di sincronizzazione di Cognito");
4. Nella schermata Configure event sources (Configura origini eventi), lascia il tipo di origine impostato su "Cognito Sync Triggering" ("Attivazione della Sincronizzazione di Cognito") e seleziona il tuo pool di identità. Fare clic su Avanti.

> **Note**
>
> Quando si configura un trigger Amazon Cognito Sync all'esterno della console, è necessario aggiungere autorizzazioni basate sulle risorse Lambda per consentire ad Amazon Cognito di richiamare la funzione. Puoi aggiungere questa autorizzazione dalla console Lambda (vedi Utilizzo delle politiche basate sulle risorse per AWS Lambda) o utilizzando l'operazione Lambda. AddPermission
> Esempio di politica Lambda Resource-Based
> La seguente policy AWS Lambda basata su risorse garantisce ad Amazon Cognito una capacità limitata di richiamare una funzione Lambda. Amazon Cognito può richiamare la funzione solo per conto del pool di identità nella `aws:SourceArn` condizione e dell'account nella condizione `aws:SourceAccount`.
>
> JSON

```
{
    "Version":"2012-10-17",
    "Id": "default",
    "Statement": [
        {
            "Sid": "lambda-allow-cognito-my-function",
            "Effect": "Allow",
            "Principal": {
                "Service": "cognito-sync.amazonaws.com"
            },
            "Action": "lambda:InvokeFunction",
            "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:MyFunction",
            "Condition": {
                "StringEquals": {
```

```
                    "AWS:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "AWS:SourceArn": "arn:aws:cognito-
identity:us-east-1:111122223333:identitypool/us-
east-1:abcdefg-1234-5678-910a-0e8443553f95"
                }
            }
        }
    ]
}
```

5. Nella schermata Configure function (Configura funzione), inserisci un nome e una descrizione per la funzione. Lascia Runtime impostato su "Node.js.» Lascia il codice invariato per il nostro esempio. L'esempio predefinito modifica i dati in fase di sincronizzazione. Registra solo il fatto che l'evento trigger di Amazon Cognito Sync si è verificato. Lascia il nome del gestore impostato su "index.handler." Per Ruolo, seleziona un ruolo IAM che concede al tuo codice l'autorizzazione per accedere a AWS Lambda. Per modificare i ruoli, consulta la console IAM. Lascia invariate le impostazioni avanzate. Fare clic su Avanti.
6. Nella schermata Review (Revisione) rivedi i dettagli e fai clic su Create function (Crea funzione). La pagina successiva mostra la tua nuova funzione Lambda.

Ora che disponi di una funzione appropriata scritta in Lambda, devi scegliere quella funzione come gestore dell'evento del trigger di Amazon Cognito Sync. La seguente procedura ti guiderà attraverso il processo.

Dalla home page della console:

1. Fai clic sul nome del pool di identità per cui desideri configurare Amazon Cognito Events. Viene visualizzata la pagina Dashboard (Pannello di controllo) per il tuo pool di identità.
2. Nell'angolo in alto a destra della pagina Dashboard (Pannello di controllo), fai clic su Edit identity pool (Modifica pool di identità). Viene visualizzata la pagina di gestione delle identità federate.
3. Scorri verso il basso e fai clic su Cognito Events (Eventi di Cognito) per espandere l'operazione;
4. Nel menu a discesa del trigger di sincronizzazione, seleziona la funzione Lambda che desideri attivare quando si verifica un evento di sincronizzazione.
5. Fai clic su Salva modifiche.

Ora, la funzione Lambda viene eseguita ogni volta che un set di dati viene sincronizzato. La sezione seguente spiega in che modo puoi leggere e modificare i dati nella tua funzione durante la loro fase di sincronizzazione.

Scrittura di una funzione Lambda per trigger di sincronizzazione

I trigger di sincronizzazione seguono il modello di programmazione usato dall'interfaccia del fornitore di servizi. Amazon Cognito fornisce l'input alla tua funzione Lambda nel formato JSON seguente.

```
{
  "version": 2,
  "eventType": "SyncTrigger",
  "region": "us-east-1",
  "identityPoolId": "identityPoolId",
  "identityId": "identityId",
  "datasetName": "datasetName",
  "datasetRecords": {
    "SampleKey1": {
      "oldValue": "oldValue1",
      "newValue": "newValue1",
      "op": "replace"
    },
    "SampleKey2": {
      "oldValue": "oldValue2",
      "newValue": "newValue2",
      "op": "replace"
    },..
  }
}
```

Amazon Cognito richiede il valore restituito della funzione nello stesso formato dell'input.

Quando scrivete funzioni per l'evento Sync Trigger, osserva quanto segue:

- Quando Amazon Cognito chiama la tua funzione Lambda durante UpdateRecords, la funzione deve rispondere entro 5 secondi. In caso contrario, il servizio Amazon Cognito Sync genera un'eccezione `LambdaSocketTimeoutException`. Non puoi aumentare il valore di timeout.
- Se ricevi un'eccezione `LambdaThrottledException`, ritenta l'operazione di sincronizzazione per aggiornare i record.
- Amazon Cognito fornisce tutti i record presenti nel set di dati come input per la funzione.

- I record che vengono aggiornati dall'utente dell'app hanno il campo op impostato come `replace`. I record eliminati hanno il campo op impostato come `remove`.
- Puoi modificare tutti i record, anche se non sono stati aggiornati dall'utente dell'applicazione.
- Tutti i campi, eccetto il datasetRecords, sono di sola lettura. Non modificarli. Se si modificano questi campi, non è possibile aggiornare i record.
- Per modificare il valore di un record, è sufficiente aggiornare il valore e impostare il campo op su `replace`.
- Per eliminare un record o impostare il campo op su `remove`, imposta un valore nullo.
- Per aggiungere un record, basta aggiungere un nuovo record alla gamma datasetRecords;
- Amazon Cognito ignora qualsiasi record omesso nella risposta quando aggiorna il record.

Funzione Lambda di esempio

Ecco un esempio di funzione Lambda che mostra come accedere, modificare e rimuovere i dati.

```
console.log('Loading function');

exports.handler = function(event, context) {
    console.log(JSON.stringify(event, null, 2));

    //Check for the event type
    if (event.eventType === 'SyncTrigger') {

        //Modify value for a key
        if('SampleKey1' in event.datasetRecords){
            event.datasetRecords.SampleKey1.newValue = 'ModifyValue1';
            event.datasetRecords.SampleKey1.op = 'replace';
        }

        //Remove a key
        if('SampleKey2' in event.datasetRecords){
            event.datasetRecords.SampleKey2.op = 'remove';
        }

        //Add a key
        if(!('SampleKey3' in event.datasetRecords)){
            event.datasetRecords.SampleKey3={'newValue':'ModifyValue3', 'op' :
 'replace'};
        }
```

```
    }
    context.done(null, event);
};
```

Utilizzo di Amazon Cognito Sync con pool di identità

Amazon Cognito Sync è una Servizio AWS libreria client che consente di sincronizzare i dati utente relativi alle applicazioni tra dispositivi. Amazon Cognito Sync può sincronizzare i dati del profilo utente su dispositivi mobili e sul Web senza utilizzare il tuo backend. Le librerie client archiviano localmente i dati nella cache, in modo che la tua app sia in grado di leggere e scrivere i dati indipendentemente dallo stato di connettività del dispositivo. Quando il dispositivo è online, puoi sincronizzare i dati. Se configuri la sincronizzazione push, puoi avvisare immediatamente altri dispositivi della disponibilità di un aggiornamento.

Gestione di set di dati

Se nell'applicazione hai implementato la funzionalità Amazon Cognito Sync, la console dei pool di identità di Amazon Cognito consente di creare ed eliminare manualmente set di dati e record per singole identità. Qualsiasi modifica apportata a set di dati o record di un'identità nella console dei pool di identità di Amazon Cognito non viene salvata finché non selezioniSynchronize (Sincronizza) nella console. La modifica non è visibile all'utente finale fino alla sincronizzazione delle chiamate dell'identità. I dati sincronizzati da altri dispositivi per le identità individuali sono visibili una volta aggiornata la pagina dei set di dati dell'elenco per un'identità in particolare.

Creazione di un set di dati per un'identità

Amazon Cognito Sync associa un set di dati a un'identità. Puoi compilare il set di dati con informazioni di identificazione sull'utente rappresentato dall'identità, quindi sincronizzare tali informazioni con tutti i dispositivi dell'utente.

Per aggiungere un set di dati e i record del set di dati a un'identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Scegli la scheda Browser di identità.
3. Seleziona l'identità da modificare.
4. In Set di dati, scegli Crea set di dati.

5. Inserisci un Nome set di dati e seleziona Crea set di dati.
6. Se desideri aggiungere record al set di dati, scegli il set di dati dai dettagli dell'identità. In Record, seleziona Crea record.
7. Inserisci una Chiave e un Valore per il record. Scegli Conferma. Ripeti l'operazione per aggiungere altri record.

Eliminazione di un set di dati associato a un'identità

Per eliminare un set di dati e i relativi record da un'identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Scegli la scheda Browser di identità.
3. Seleziona l'identità che contiene il set di dati da eliminare.
4. In Set di dati, scegli il pulsante di opzione accanto al set di dati da eliminare.
5. Seleziona Elimina. Esamina la scelta e seleziona nuovamente Elimina.

Pubblicazione in blocco di dati

La pubblicazione in blocco può essere utilizzata per esportare i dati già archiviati nell'archivio di Amazon Cognito Sync in un flusso Amazon Kinesis. Per istruzioni su come pubblicare in blocco tutti i flussi, consulta Implementazione dei flussi di Amazon Cognito Sync.

Attivazione della sincronizzazione push

Amazon Cognito monitora automaticamente l'associazione tra identità e dispositivi. L'utilizzo della funzione di sincronizzazione push, assicura che ogni istanza di una determinata identità venga comunicata quando si modificano i dati di identità. La sincronizzazione push è tale per cui ogni volta che il set di dati cambia per un'identità, tutti i dispositivi associati a tale identità ricevono una notifica push silenziosa per segnalare le modifiche.

Puoi attivare la sincronizzazione push nella console di Amazon Cognito.

Per attivare la sincronizzazione push

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Scegli la scheda Proprietà del pool di identità.

3. In Sincronizzazione push, seleziona Modifica
4. Seleziona Attiva sincronizzazione push con il pool di identità.
5. Scegli una delle Applicazioni di piattaforma Amazon Simple Notification Service (Amazon SNS) che hai creato nella Regione AWS corrente. Amazon Cognito pubblica notifiche push nell'applicazione di piattaforma. Seleziona Crea applicazione di piattaforma per passare alla console di Amazon SNS e crearne una nuova.
6. Per pubblicare nell'applicazione di piattaforma, Amazon Cognito assume un ruolo IAM in Account AWS. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza documento di policy per esaminare le autorizzazioni che Amazon Cognito assegna al tuo nuovo ruolo IAM.
7. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo Account AWS che desideri utilizzare. Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.
8. Seleziona Salva modifiche.

Configurazione di Amazon Cognito Streams

Amazon Cognito Streams offre agli sviluppatori il controllo e l'analisi dei loro dati archiviati in Amazon Cognito Sync. Gli sviluppatori possono ora configurare un flusso Kinesis per ricevere eventi come dati. Amazon Cognito può eseguire il push di ogni modifica del set di dati in un flusso Kinesis in tempo reale. Per istruzioni su come configurare Amazon Cognito Streams nella console di Amazon Cognito, consulta Implementazione dei flussi di Amazon Cognito Sync.

Configurazione di Amazon Cognito Events

Amazon Cognito Events ti consente di eseguire una AWS Lambda funzione in risposta a eventi importanti in Amazon Cognito Sync. Amazon Cognito Sync lancia l'evento trigger di sincronizzazione quando viene sincronizzato un set di dati. Puoi utilizzare l'evento del trigger di sincronizzazione per eseguire un'azione quando un utente aggiorna i dati. Per istruzioni sulla configurazione di Amazon Cognito Events dalla console, consulta Personalizzazione dei flussi di lavoro con Amazon Cognito Events.

Per ulteriori informazioni AWS Lambda, consulta. AWS Lambda

Personalizzazione del marchio dell'interfaccia utente ospitata (classica)

Puoi utilizzare l'API Console di gestione AWS, o the AWS CLI o, per specificare le impostazioni di personalizzazione classiche per l'interfaccia utente ospitata. È possibile caricare un'immagine del logo personalizzata da visualizzare nell'app. Puoi anche applicare alcune opzioni dei fogli di stile a cascata (CSS) all'aspetto dell'interfaccia utente.

Puoi personalizzare le impostazioni predefinite dell'interfaccia utente e sovrascrivere i singoli client dell'app con impostazioni specifiche. Amazon Cognito applica la configurazione predefinita a tutti i client di app che non dispongono di impostazioni a livello di client.

Nella console Amazon Cognito e nelle richieste API, la richiesta che imposta la personalizzazione dell'interfaccia utente non deve superare le dimensioni di 135 KB. In rari casi, le dimensioni della somma delle intestazioni di richiesta, del file CSS e del logo potrebbero eccedere i 135 KB. Amazon Cognito codifica il file di immagine in Base64. Ciò aumenta le dimensioni di un'immagine da 100 KB a 130 KB, mantenendo cinque KB per intestazioni della richiesta e CSS. Se la richiesta è troppo grande, la Console di gestione AWS o la tua richiesta `SetUICustomization` API restituisce un `request parameters too large` errore. Regolare l'immagine del logo in modo che le dimensioni non superino i 100 KB e le dimensioni del file CSS non siano maggiori di 3 KB. Non è possibile impostare CSS e personalizzazione del logo separatamente.

> **Note**
>
> Per personalizzare l'interfaccia utente, è necessario configurare un dominio per il pool di utenti.

Specificare un logo personalizzato nel marchio classico

Amazon Cognito centra il logo personalizzato sopra i campi di input in Endpoint Login.

Scegli un file PNG, JPG o JPEG scalabile fino a 350 x 178 pixel per il logo personalizzato dell'interfaccia utente ospitata. Le dimensioni del file del logo non possono essere maggiori di 100 KB o 130 KB dopo la codifica di Amazon Cognito in Base64. Per impostare un ingresso `ImageFile`

SetUICustomizationnell'API, converti il file in una stringa di Base64-encoded testo o, in AWS CLI, fornisci un percorso del file e lascia che Amazon Cognito lo codifica per te.

Specificare le personalizzazioni CSS nel branding classico

È possibile personalizzare il CSS per le pagine dell'app ospitata, con le seguenti restrizioni:

- È possibile utilizzare uno dei seguenti nomi di classe CSS:
  - `background-customizable`
  - `banner-customizable`
  - `errorMessage-customizable`
  - `idpButton-customizable`
  - `idpButton-customizable:hover`
  - `idpDescription-customizable`
  - `inputField-customizable`
  - `inputField-customizable:focus`
  - `label-customizable`
  - `legalText-customizable`
  - `logo-customizable`
  - `passwordCheck-valid-customizable`
  - `passwordCheck-notValid-customizable`
  - `redirect-customizable`
  - `socialButton-customizable`
  - `submitButton-customizable`
  - `submitButton-customizable:hover`
  - `textDescription-customizable`
- I valori delle proprietà possono contenere HTML, ad eccezione dei seguenti valori: `@import`, `@supports`, `@page`, oppure istruzioni `@media` o Javascript.

È possibile personalizzare le seguenti proprietà CSS.

Etichette

- font-weight è un multiplo di 100 da 100 e 900.

- color (colore) è il colore del testo. Deve essere un valore di colore CSS legale.

Campi di input

- width è la larghezza in percentuale del blocco contenitore.
- height (altezza) è l'altezza del campo di input in pixel (px).
- color (colore) è il colore del testo. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del campo di input. Può essere un qualsiasi valore di colore dello standard CSS.
- border (bordo) è un valore dello standard CSS del bordo che specifica la larghezza, la trasparenza e il colore del bordo della finestra della tua app. Larghezza può essere qualsiasi valore da 1px a 100px. Trasparenza può essere solida o nessuna. Colore può essere un qualsiasi valore di colori standard.

Descrizioni testuali

- padding-top è la quantità di padding sopra la descrizione testuale.
- padding-bottom è la quantità di padding sotto la descrizione testuale.
- display può essere `block` o `inline`.
- font-size è la dimensione del font per le descrizioni testuali.
- color (colore) è il colore del testo. Deve essere un valore di colore CSS legale.

Pulsante di invio

- font-size è la dimensione del font del testo del pulsante
- font-weight è il peso del font del testo del pulsante `bold`, `italic` o `normal`.
- margine è una stringa di quattro valori che indicano le dimensioni dei margini superiore, destro, inferiore e sinistro del pulsante.
- font-size è la dimensione del font per le descrizioni testuali.
- width (larghezza) è la larghezza del testo del pulsante in percentuale del blocco contenitore.
- height (altezza) è l'altezza del pulsante in pixel (px).
- color (colore) è il colore del testo del pulsante. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del pulsante. Può essere un valore di qualsiasi standard di colori.

Banner

- spaziatura interna è una stringa di quattro valori che indicano le dimensioni della spaziatura interna superiore, destro, inferiore e sinistro del banner.

- background-color è il colore di sfondo del banner. Può essere un qualsiasi valore di colore dello standard CSS.

Pulsante di invio con animazione al passaggio del mouse

- color (colore) è il colore di primo piano del pulsante al passaggio del mouse. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del pulsante al passaggio del mouse. Può essere un qualsiasi valore di colore dello standard CSS.

Pulsante provider di identità con animazione al passaggio del mouse

- color (colore) è il colore di primo piano del pulsante al passaggio del mouse. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del pulsante al passaggio del mouse. Può essere un qualsiasi valore di colore dello standard CSS.

Controllo password non valido

- color (colore) è il colore del testo del messaggio `"Password check not valid"`. Può essere un qualsiasi valore di colore dello standard CSS.

Contesto

- background-color è il colore di sfondo della finestra dell'app. Può essere un qualsiasi valore di colore dello standard CSS.

Messaggi di errore

- margine è una stringa di quattro valori che indicano le dimensioni dei margini superiore, destro, inferiore e sinistro.
- padding è la dimensione del padding.
- font-size è la dimensione del font.
- width (larghezza) è la larghezza del messaggio di errore in percentuale del blocco contenitore.
- background è il colore di sfondo del messaggio di errore. Può essere un qualsiasi valore di colore dello standard CSS.
- bordo è una stringa di tre valori che specificano la larghezza, trasparenza e colore del bordo.
- color (colore) è il colore del messaggio di errore. Può essere un qualsiasi valore di colore dello standard CSS.
- box-sizing viene utilizzata per indicare al browser quello che devono includere le proprietà di ridimensionamento (larghezza e altezza).

Pulsante provider di identità

- height (altezza) è l'altezza del pulsante in pixel (px).
- width (larghezza) è la larghezza del testo del pulsante in percentuale del blocco contenitore.
- text-align è l'impostazione per allineare il testo. Può essere `left`, `right` o `center`.
- margin-bottom è l'impostazione del margine inferiore.
- color (colore) è il colore del testo del pulsante. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del pulsante. Può essere un qualsiasi valore di colore dello standard CSS.
- border-color è il colore di sfondo del bordo. Può essere un qualsiasi valore di colore dello standard CSS.

Descrizioni provider di identità

- padding-top è la quantità di padding sopra la descrizione.
- padding-bottom è la quantità di padding sotto la descrizione.
- display può essere `block` o `inline`.
- font-size è la dimensione del font per le descrizioni.
- color è il colore del testo per le intestazioni delle sezioni IdP, ad esempio Accedi con il tuo ID aziendale. Deve essere un valore di colore CSS legale.

Testo legale

- color (colore) è il colore del testo. Può essere un qualsiasi valore di colore dello standard CSS.
- font-size è la dimensione del font.

> **Note**
>
> Personalizzando un Legal text (Testo legale), stai personalizzando il messaggio We won't post to any of your accounts without asking first (Non pubblicheremo su nessuno dei tuoi account senza il tuo consenso) che viene visualizzato sotto i provider di identità sociale nella pagina di accesso.

Logo

- max-width è la larghezza massima in percentuale del blocco contenitore.

- max-height è l'altezza massima in percentuale del blocco contenitore.
- background-color è il colore dello sfondo per i log con sezioni trasparenti. Deve essere un valore di colore CSS legale.

Focus del campo di input

- border-color è il colore del campo di input. Può essere un qualsiasi valore di colore dello standard CSS.
- outline è il bordo del campo di input in pixel.

Pulsante social

- height (altezza) è l'altezza del pulsante in pixel (px).
- text-align è l'impostazione per allineare il testo. Può essere `left`, `right` o `center`.
- width (larghezza) è la larghezza del testo del pulsante in percentuale del blocco contenitore.
- margin-bottom è l'impostazione del margine inferiore.

Controllo password valido

- color (colore) è il colore del testo del messaggio `"Password check valid"`. Può essere un qualsiasi valore di colore dello standard CSS.

Personalizzazione dell'interfaccia utente ospitata con il marchio classico in Console di gestione AWS

Puoi utilizzare il Console di gestione AWS per specificare le impostazioni di personalizzazione dell'interfaccia utente per la tua app.

> **Note**
>
> Puoi visualizzare l'interfaccia utente ospitata con le personalizzazioni costruendo il seguente URL, con le specifiche del bacino d'utenza, e inserendolo all'interno di un browser: `https://<your_domain>/login?response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callbac`
> Potrebbe essere necessario attendere fino a un minuto per aggiornare il browser prima che vengano visualizzate le modifiche effettuate nella console.
> Il dominio viene visualizzato nella scheda App integration (Integrazione app) alla voce Dominio. L'ID del client dell'app e l'URL di callback sono visualizzati nella scheda App client settings (Impostazioni client dell'app).

Specificazione delle impostazioni di personalizzazione dell'interfaccia utente dell'App

1. Accedi alla console Amazon Cognito.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Crea un dominio dalla scheda Dominio o aggiorna il dominio esistente. In Versione di branding, imposta il dominio per utilizzare l'interfaccia utente ospitata (classica).
4. Scegli il menu Accesso gestito.
5. Per personalizzare le impostazioni dell'interfaccia utente per tutti i client dell'app, individua Style nelle impostazioni dell'interfaccia utente ospitata e seleziona Modifica.
6. Per personalizzare le impostazioni dell'interfaccia utente per un client di app, vai al menu App clients e seleziona il client dell'app che desideri modificare, quindi individua lo stile Hosted UI (classico) e seleziona Override. Seleziona Edit (Modifica).
7. Per caricare il tuo file immagine logo, scegli Choose file (Scegli file) o Replace current file (Sostituisci file corrente).
8. Per personalizzare il CSS per l'interfaccia utente ospitata, scarica CSS template.css e modifica il modello con i valori che desideri personalizzare. Solo le chiavi incluse nel modello possono essere utilizzate con l'interfaccia utente ospitata. Le chiavi CSS aggiunte non si riflettono nell'interfaccia utente. Dopo aver personalizzato il file CSS, scegli Choose file (Scegli file) o Replace current file (Sostituisci file corrente) per caricare il tuo CSS personalizzato.

Personalizzazione dell'interfaccia utente ospitata con il marchio classico nell'API dei pool di utenti e con AWS CLI

Usa i seguenti comandi per specificare le impostazioni della personalizzazione dell'interfaccia utente dell'App del tuo bacino d'utenza.

Per ottenere le impostazioni di personalizzazione dell'interfaccia utente per l'interfaccia utente di un'app integrata del pool di utenti, utilizza le seguenti operazioni API.

- AWS CLI: `aws cognito-idp get-ui-customization`
- AWS API: GetUICustomization

Per configurare le impostazioni di personalizzazione dell'interfaccia utente per l'interfaccia utente di un'app integrata del pool di utenti, utilizza le seguenti operazioni API.

- AWS CLI dal file di immagine: `aws cognito-idp set-ui-customization --user-pool-id <your-user-pool-id> --client-id <your-app-client-id> --image-file fileb://"<path-to-logo-image-file>" --css ".label-customizable{ color: <color>;}"`
- AWS CLI con immagine codificata come testo binario Base64: `aws cognito-idp set-ui-customization --user-pool-id <your-user-pool-id> --client-id <your-app-client-id> --image-file <base64-encoded-image-file> --css ".label-customizable{ color: <color>;}"`
- AWS API: SetUICustomization

Cronologia dei documenti per Amazon Cognito

Nella seguente tabella sono descritte importanti aggiunte alla documentazione di Amazon Cognito. Inoltre, effettuiamo aggiornamenti secondari frequenti alla documentazione in risposta al feedback inviato. Per inviare un feedback, individuare il collegamento Feedbackin fondo a qualsiasi pagina della documentazione di Amazon Cognito.

Modifica	Descrizione	Data
API e riferimenti agli endpoint riorganizzati	Il riferimento agli endpoint del pool di utenti e all'acces so gestito è ora una sezione di primo livello della guida. La guida non contiene più collegamenti ai pool di utenti di Amazon Cognito, ai pool di identità di Amazon Cognito e ai riferimenti dell'API Amazon Cognito Sync. Questi riferimen ti API rimangono disponibi li nelle rispettive guide di riferimento API.	8 ottobre 2026
Codici di stato nelle risposte AWS WAF personalizzate ai blocchi	Amazon Cognito ora restituis ce un codice di `403` stato al posto dei codici di `5xx` stato delle risposte ai blocchi AWS WAF personalizzate nell'acce sso gestito e nella classica interfaccia utente ospitata.	1 ottobre 2026
Amazon Cognito Sync non è più aperto a nuovi clienti.	Amazon Cognito Sync non è più aperto a nuovi clienti. Per le alternative ad Amazon Cognito Sync, esplora e AWS	31 luglio 2026

	AppSync DynamoDB. Ulteriori informazioni.	
Machine-to-machine autorizzazione con GetClientToken	Ora puoi emettere token di accesso per l'autorizzazione da macchina a macchina (M2M) con l'operazione `GetClientToken` API Amazon Cognito, senza un dominio del pool di utenti.	29 luglio 2026
Importazione utente con hash della password	Ora puoi importare gli utenti con gli hash delle password esistenti, abilitando l'accesso immediato senza richieder e la reimpostazione della password.	13 luglio 2026
Gestione dei limiti predisposta	Ora puoi gestire i limiti assegnati per il tuo pool di utenti con le API di provision ing di Amazon Cognito.	2 luglio 2026
Amazon Cognito Sync non sarà più aperto a nuovi clienti a partire dal 30 luglio 2026.	Amazon Cognito Sync non sarà più aperto a nuovi clienti a partire dal 30 luglio 2026. I clienti esistenti di Amazon Cognito Sync possono continuare a utilizzare il servizio. Per le alternative ad Amazon Cognito Sync, esplora e AWS AppSync DynamoDB. Ulteriori informazioni.	30 giugno 2026

Selezione della versione di TLS per domini personalizzati	Ora puoi selezionare la versione TLS minima per il tuo dominio personalizzato quando crei o aggiorni un dominio con pool di utenti con l'API Amazon Cognito.	29 giugno 2026
Multi-Region replica per pool di utenti.	Ora puoi configurare una copia aggiuntiva dei tuoi pool di utenti in un altro Regione AWS.	28 maggio 2026
Chiavi KMS gestite dal cliente per la crittografia del pool di utenti a riposo.	Ora puoi configurare la crittografia del pool di utenti inattivo con la tua chiave KMS personalizzata.	28 maggio 2026
Documentazione aggiornat a sul comportamento dei tentativi di ripetizione dei trigger Lambda.	Documentazione aggiornat a per chiarire che Amazon Cognito può ripetere le chiamate della funzione Lambda allo scadere del timeout, senza specificare un numero fisso di tentativi.	23 marzo 2026
Rotazione segreta dei client per i client delle app.	Ora puoi associare fino a due segreti per client dell'app per la rotazione segreta. Puoi anche fornire il tuo valore segreto personalizzato quando aggiungi un segreto.	30 gennaio 2026

Inbound Federation Lambda Trigger è disponibile nei pool di utenti di Amazon Cognito.	Ora puoi utilizzare il trigger lambda della federazione in entrata per trasformare gli attributi utente federati durante il processo di autenticazione con provider di identità esterni.	28 gennaio 2026
AWS PrivateLink è disponibile nei pool di identità di Amazon Cognito.	Ora puoi inviare richieste alle operazioni dell'API del pool di identità di Amazon Cognito tramite endpoint VPC.	10 dicembre 2025
AWS PrivateLink è disponibile nei pool di utenti di Amazon Cognito.	Ora puoi inviare richieste alle operazioni API dei pool di utenti di Amazon Cognito tramite endpoint VPC.	4 novembre 2025
I pool di utenti di Amazon Cognito ora supportano l'associazione delle risorse.	I tuoi server di risorse possono ora eseguire la verifica dell'audience dei token di accesso con l'associazione delle risorse RFC 8707.	24 ottobre 2025
È stato aggiunto un capitolo sulla risoluzione dei problemi.	Nuovo capitolo con soluzioni ai problemi più comuni relativi ai pool di utenti.	24 ottobre 2025
Aggiunta una nuova quota per le richieste di documenti di chiavi web JSON del pool di utenti.	Nuova voce di quota per il recupero `.well-known/jwks.json` dai server di autorizzazione OAuth 2.0 del pool di utenti.	13 ottobre 2025
Documenti relativi ai termini e alle condizioni nel login gestito.	Ora puoi aggiungere link ai tuoi termini e condizioni e all'informativa sulla privacy nelle pagine di registrazione con accesso gestito.	2 ottobre 2025

Nuova applicazione di esempio introduttiva per pool di identità.	È ora possibile configurare un'applicazione di esempio che mostri le funzionalità dei pool di identità.	30 settembre 2025
Aggiornamenti ai requisiti per le politiche di fiducia dei ruoli IAM per i pool di identità.	Ora puoi apportare modifiche alle policy di trust dei ruoli per il service principal dei pool di identità di Amazon Cognito solo quando una chiave condizionale esamina il pubblico per la federazione OIDC (identità web) a uno o più pool di identità.	1 agosto 2025
Amazon Cognito è ora disponibile in Messico (Centrale). Regione AWS	Ora puoi creare risorse Amazon Cognito nella regione del Messico (Centrale).	24 luglio 2025
Amazon Cognito è ora disponibile nella regione Asia-Pacifico (Tailandia). Regione AWS	Ora puoi creare risorse Amazon Cognito nella regione Asia Pacifico (Tailandia).	24 luglio 2025
AWS WAF ACL web nell'accesso gestito.	Ora puoi applicare le regole ACL AWS WAF web ai client di app con pool di utenti che dispongono della versione Managed Login Branding.	24 giugno 2025
Esempi di trigger Lambda aggiornati.	È stata aggiornata la funzione di esempio per i trigger Lambda personalizzati del mittente di e-mail e SMS per renderla compatibile con 22.x. Node.js L'esempio è ora anche più accessibile per i test.	19 maggio 2025

Nuovo `prompt` parametro.	Ora hai un maggiore controllo sulla riautenticazione delle sessioni di accesso gestite esistenti con il `prompt` parametro. Puoi anche passare i valori per questo parametro a provider di terze parti.	15 maggio 2025
Metadati del client per le richieste M2M.	Ora puoi passare i metadati del client nelle credenziali del client o nelle richieste da macchina a macchina (M2M). Amazon Cognito passa i metadati del client M2M al trigger Lambda prima della generazione di token.	29 aprile 2025
Aggiorna la rotazione dei token.	Ora puoi ottenere nuovi token di aggiornamento e invalidar e gli originali nelle richieste di refresh token.	22 aprile 2025
Amazon Cognito è ora disponibile nella regione Asia-Pacifico (Malesia). Regione AWS	Ora puoi creare risorse Amazon Cognito nella regione Asia Pacifico (Malesia).	7 marzo 2025
Personalizzazione dei token di accesso per le identità delle macchine.	Il trigger Lambda prima della generazione di token ora include un evento della versione tre che modifica le attestazioni e gli ambiti dei token di accesso nelle concessioni di credenziali client per l'autorizzazione da macchina a macchina (M2M).	3 marzo 2025

AmazonCognitoPowerUser AWS Informazioni aggiornate sulla politica gestita.	È stata aggiunta un' AWS End User Messaging SMS operazione nella policy AWS gestita per i pool di utenti esperti di Amazon Cognito.	27 febbraio 2025
Panoramica aggiornata dell'integrazione con OpenID Connect (OIDC).	È stato aggiunto un diagramma che illustra come Amazon Cognito si autentica con i provider di identità OIDC.	25 febbraio 2025
Sono state aggiunte informazioni sulla logica MFA.	È stato aggiunto un diagramma che illustra come Amazon Cognito applica le impostazioni di autenticazione a più fattori (MFA) del pool di utenti agli utenti in fase di esecuzione.	25 febbraio 2025
Sono state aggiunte le best practice di sicurezza per i pool di utenti di Amazon Cognito.	È stata aggiunta una pagina sulla protezione dei segreti e su come seguire in altro modo le migliori pratiche di sicurezza nella configurazione del pool di utenti.	25 febbraio 2025
Aggiornamenti alle risorse introduttive per i pool di utenti.	L'esperienza introduttiva con i pool di utenti di Amazon Cognito presenta un nuovo design della console e nuove opzioni di applicazione.	21 novembre 2024

Nuovo modello di prezzo con piani di funzionalità.	È stato aggiornato il modello di fatturazione per i pool di utenti. Le funzionalità di sicurezza avanzate ora proteggono dalle minacce. I componenti della licenza per le funzionalità di sicurezza avanzate sono ora inclusi nei piani di funzionalità Essentials e Plus.	21 novembre 2024
Nuova funzionalità di accesso gestito.	Lancio dell'accesso gestito, un aggiornamento dell'interfaccia utente ospitata.	21 novembre 2024
Un nuovo metodo di autenticazione e nuovi flussi di autenticazione.	Ora puoi accedere ai pool di utenti di Amazon Cognito con password e password monouso.	21 novembre 2024
Informazioni aggiornate su. `AmazonCognitoUnAuthedIdentitiesSessionPolicy`	AWS Key Management Service Le operazioni nella policy AWS gestita per la riduzione delle identità non autenticate sono state spostate dalla policy in linea alla policy gestita. AWS	1 novembre 2024
Parametro aggiunto. `login_hint`	Ora puoi aggiungere un suggerimento per il nome utente alle richieste di autorizzazione per l'interfaccia utente ospitata, OIDC IdPs e Google. IdPs	3 ottobre 2024

Nuove funzionalità di sicurezza avanzate per l'MFA delle e-mail.	Ora puoi inviare codici di autenticazione a più fattori (MFA) tramite messaggio e-mail con funzionalità di sicurezza avanzate.	12 settembre 2024
Nuovi contenuti e modifiche alla pagina.	Titoli modificati, contenuti non necessari rimossi, aggiunte introduzioni basate su scenari, pool di utenti OIDC spostati e riferimento agli endpoint dell'interfaccia utente ospitati alla sezione pool di utenti.	9 settembre 2024
Informazioni aggiornate su. `AmazonCognitoUnAuthedIdentitiesSessionPolicy`	La politica AWS gestita per l'eliminazione delle identità non autenticate nei pool di identità ora consente Amazon Location Service.	9 agosto 2024
Nuova prevenzione delle minacce per l'autenticazione personalizzata con trigger Lambda e rilevamento avanzato delle minacce.	Ora puoi analizzare l'autenticazione personalizzata, l'accesso con protezione dalle minacce e applicare risposte di autenticazione adattive. La protezione dalle minacce ora analizza anche il traffico di accesso alla ricerca di distanze geografiche impossibili tra i tentativi.	8 agosto 2024

Nuove funzionalità di sicurezza avanzate per la prevenzione del riutilizzo delle password e l'esportazione dei registri delle attività degli utenti.	Ora puoi esportare i registri delle attività degli utenti e impostare una politica di cronologia delle password con funzionalità di sicurezza avanzate nei pool di utenti di Amazon Cognito.	6 agosto 2024
Amazon Cognito è ora disponibile nel Canada occidentale (Calgary) e nell'Asia Pacifico (Hong Kong). Regioni AWS	Ora puoi creare risorse Amazon Cognito nelle regioni Canada occidentale (Calgary) e Asia Pacifico (Hong Kong).	9 luglio 2024
Descrizione migliorata del comportamento delle applicazioni per una sicurezza avanzata	Informazioni aggiornate sui dati contestuali del dispositivo per l'autenticazione adattiva di sicurezza avanzata.	10 giugno 2024
Aggiunto il supporto per oggetti complessi nel trigger Lambda pre token	Ora puoi aggiungere array e oggetti JSON alle attestazioni ID e Access Token.	30 maggio 2024
Informazioni aggiornate sulle autorizzazioni verificate e Amazon Cognito.	Amazon Verified Permissio ns ora ha un'integrazione più diretta con Amazon Cognito.	15 maggio 2024
Multi-Region Identità verificate da Amazon SES.	In alcuni casi Regioni AWS senza Amazon SES, gli utenti di Amazon Cognito raggruppa no le e-mail di bilanciamento del carico tra due regioni remote.	10 maggio 2024

Sono state aggiunte informazioni sull'autorizzazione M2M e sulla gestione dei costi.	Scopri come utilizzare le concessioni di credenziali client per casi d'uso da macchina a macchina (M2M) con pool di utenti di Amazon Cognito.	9 maggio 2024
Amazon Cognito è ora disponibile in Europa (Spagna) e Asia Pacifico (Hyderabad). Regioni AWS	Ora puoi creare risorse Amazon Cognito nelle regioni Europa (Spagna) e Asia Pacifico (Hyderabad).	15 aprile 2024
Amazon Cognito è ora disponibile nella regione Asia-Pacifico (Melbourne). Regione AWS	Ora puoi creare risorse Amazon Cognito nella regione Asia Pacifico (Melbourne).	4 aprile 2024
È stata aggiunta un'app Android di esempio in Flutter per i pool di utenti di Amazon Cognito.	Puoi creare un'app mobile iniziale per Amazon Cognito a partire da un'applicazione Flutter di esempio. GitHub	4 aprile 2024
Nuovi contenuti introduttivi	Contenuti estesi per iniziare, scenari comuni, best practice multi-tenant e accesso alle risorse dopo l'accesso.	1 aprile 2024
Amazon Cognito è ora disponibile in Europa (Zurigo). Regione AWS	Ora puoi creare risorse Amazon Cognito nella regione Europa (Zurigo).	14 marzo 2024
Amazon Cognito è ora disponibile in Medio Oriente (Emirati Arabi Uniti). Regione AWS	Ora puoi creare risorse Amazon Cognito nella regione del Medio Oriente (Emirati Arabi Uniti).	8 marzo 2024

Nuove funzionalità SAML e contenuti migliorati.	Ora puoi firmare le richieste SAML, crittografare le risposte SAML e configurare SAML SSO. IdP-initiated	1 febbraio 2024
Sono disponibili aumenti delle quote.	Ora puoi acquistare capacità aggiuntiva per le quote dei tassi di richiesta di Amazon Cognito.	25 gennaio 2024
I pool di identità di Amazon Cognito supportano le tariffe di richiesta in Service Quotas.	Ora puoi monitorare le quote di richieste al secondo (RPS) per i pool di identità di Amazon Cognito e richiedere l'aumento nella console Service Quotas.	19 dicembre 2023
È stata aggiunta una nuova funzionalità per la personalizzazione del contenuto dei token di accesso.	È ora possibile aggiunger e, modificare e rimuovere attestazioni e ambiti nei token di accesso del pool di utenti.	12 dicembre 2023
Contenuti migliorati sui client delle app e sugli ambiti OAuth.	Modifiche e correzioni di Clarity a Application-specific impostazioni con i client dell'app e Ambiti, M2M e server di risorse. Istruzioni della console precedente rimosse.	14 novembre 2023
Contenuti migliorati sui dispositivi e sull'autenticazione dei dispositivi.	Nuovi contenuti sull'uso delle chiavi del dispositivo e sull'autenticazione SRP del dispositivo.	18 ottobre 2023

Console di gestione AWS Linee guida aggiornate.	È stato rimosso il riferimento alla console dei pool di utenti e gli argomenti sono stati ridistribuiti all'interno degli oggetti correlati e sono state aggiunte linee guida all'organizzazione basata su schede nella console Amazon Cognito.	30 agosto 2023
De-emphasized accesso diretto all'endpoint LOGIN.	È stata aggiunta una panoramica visiva del pool di utenti Endpoint Login ed enfatizzato l'avvio dell'autenticazione con Endpoint Authorize.	30 agosto 2023
Amazon Cognito è ora disponibile in Asia Pacifico (Osaka) e Israele (Tel Aviv). Regioni AWS	Ora puoi creare risorse Amazon Cognito nelle regioni di Asia Pacifico (Osaka) e Israele (Tel Aviv).	30 agosto 2023
Sono state introdotte informazioni sull'autorizzazione per Amazon Cognito con Amazon Verified Permissions.	Nella tua app, puoi richiamare l'API Autorizzazioni verificate per produrre decisioni di accesso da un'autorità centrale.	1° agosto 2023
È stata aggiunta una nuova funzionalità per la registrazione delle attività dettagliate degli utenti del pool di utenti su Amazon CloudWatch Logs.	Ora puoi registrare gli errori di recapito di e-mail e messaggi SMS nei gruppi di CloudWatch log.	1° agosto 2023

Informazioni aggiornate sulla politica AWS gestita per gli utenti guest del pool di identità.	L'ambito di applicazione delle autorizzazioni per gli utenti guest del pool di identità ora include sia una politica di sessione in linea che una politica di sessione gestita. AWS	16 maggio 2023
Miglioramento dei contenuti e nuove istruzioni della console per i pool di identità di Amazon Cognito.	Sono state aggiunte nuove procedure dettagliate sulla console per riflettere la nuova esperienza della console, dettagli sull'integrazione del codice migliorati per pool di identità.	16 maggio 2023
Aggiunte e miglioramenti alla home page del servizio e alla home page dei pool di utenti.	Pagine di panoramica aggiornate per Amazon Cognito e pool di utenti.	16 maggio 2023
Miglioramenti generali alla documentazione sui token del pool di utenti.	Sono stati aggiornati i token di esempio e sono state aggiunte nuove informazioni sulla verifica dei token.	16 febbraio 2023
Ora puoi registrare gli eventi relativi ai dati dei pool di identità di Amazon Cognito. AWS CloudTrail	CloudTrail supporta la selezione di pool di identità Amazon Cognito, operazioni API ad alto volume in percorsi che registrano gli eventi relativi ai dati.	15 febbraio 2023

Esempi e descrizioni aggiornati dei trigger Lambda.	Gli esempi di trigger Lambda vengono aggiornati alla JavaScript versione 3. Ora puoi correlare direttamente i trigger Lambda alle operazioni API.	31 gennaio 2023
I pool di identità di Amazon Cognito applicano una policy AWS gestita alle sessioni non autenticate.	Gli utenti del pool di identità che si autenticano utilizzando il flusso avanzato ora hanno una politica AWS gestita aggiuntiva applicata alla loro sessione.	31 gennaio 2023
Sono stati aggiunti esempi di codice.	Questa guida ora include codice di esempio per la tua app Amazon Cognito in una varietà di linguaggi di programmazione.	23 gennaio 2023
Sono state aggiunte informazioni sui modelli API e sull'autenticazione con i pool di utenti di Amazon Cognito.	I pool di utenti di Amazon Cognito dispongono di più interfacce API e formati per l'autorizzazione della richiesta.	15 dicembre 2022
Amazon Cognito è ora disponibile in Europa (Milano). Regione AWS	È ora possibile creare pool di utenti di Amazon Cognito nella regione Europa (Milano).	6 dicembre 2022
Sono state aggiunte informazioni sulla protezione dall'eliminazione del pool di utenti.	Quando crei un nuovo pool di utenti con Console di gestione AWS, ora è protetto dall'eliminazione per impostazione predefinita.	20 ottobre 2022

È stata aggiunta una guida per l'utente per l'interfaccia utente ospitata e informazioni sull'MFA TOTP nell'interfaccia utente ospitata.	Gli utenti ora possono registrare un dispositivo con autenticazione MFA con token di software TOTP nell'interfaccia utente ospitata di Amazon Cognito. È ora possibile visualizzare in anteprima l'interfaccia utente ospitata predefinita.	8 settembre 2022
Sono state aggiunte informazioni su AWS WAF Amazon Cognito.	Ora puoi associare un ACL AWS WAF Web a un pool di utenti Amazon Cognito.	3 agosto 2022
Sono stati aggiunti altri eventi di esempio AWS CloudTrail .	Amazon Cognito ora registra le richieste della federazione e dell'interfaccia utente ospitata nel tuo trail.	15 giugno 2022
Sono state aggiunte informazioni sulla verifica degli attributi in due passaggi.	Ora puoi scegliere se l'utente deve verificare un nuovo indirizzo e-mail o numero di telefono prima di poter eseguire l'accesso con esso.	09 giugno 2022
Documentazione federativa aggiornata. Nuova funzionalità di propagazione degli indirizzi IP.	Procedure dettagliate aggiornate per la configurazione dei social pool di utenti. IdPs Aggiunta di informazioni sui profili di utenti federati e sulla mappatura degli attributi . Sono state aggiunte nuove informazioni sulle impronte digitali dei dispositivi per una sicurezza avanzata.	31 maggio 2022

Accedi agli utenti federati senza interazione con l'interfaccia utente ospitata	È stata aggiunta una nuova pagina su come aggiungere segnalibri alle applicazioni in modo che Amazon Cognito indirizzi silenziosamente gli utenti all'accesso federato.	29 maggio 2022
In-Region Messaggi SMS ed e-mail per pool di utenti Amazon Cognito	Ora puoi utilizzare Amazon Simple Notification Service per i messaggi SMS e Amazon Simple Email Service per i messaggi e-mail nello Regione AWS stesso pool di utenti.	14 marzo 2022
Aggiornamenti alla pagina delle quote	Sono state aggiunte e chiarite le quote relative alle risorse e ai tassi di richiesta.	10 gennaio 2022
Nuova esperienza di console con pool di utenti Amazon Cognito	Sono state aggiornate le istruzioni per creare e gestire i bacini d'utenza nella console Amazon Cognito aggiornata.	18 novembre 2021
RevokeToken API ed endpoint di revoca	È possibile utilizzare l' RevokeToken operazion e per revocare un token di aggiornamento per un utente.	10 giugno 2021
Multi-tenant migliori pratiche	Sono state aggiunte le migliori pratiche per le applicazioni multi-tenant.	04 marzo 2021

Attributi per il controllo degli accessi	I pool di identità di Amazon Cognito forniscono attributi per il controllo degli accessi (AFAC) per consentire ai clienti di concedere agli utenti l'accesso alle risorse. AWS L'autorizzazione può essere eseguita in base agli attributi degli utenti dal provider di identità utilizzato per la federazione con Amazon Cognito.	15 gennaio 2021
Trigger Lambda del mittente SMS personalizzato e Trigger Lambda del mittente e-mail personalizzato	Il trigger Lambda del mittente SMS personalizzato e il trigger Lambda del mittente e-mail personalizzato consentono a un provider di terza parte di inviare notifiche e-mail e SMS agli utenti dall'interno del codice della funzione Lambda.	30 novembre 2020
Aggiornamenti del token Amazon Cognito	Sono state aggiunte informazioni aggiornate sulla scadenza dei token di accesso, ID e aggiornamento.	29 ottobre 2020

Quotas del servizio Amazon Cognito	Le Service Quotas sono disponibili per le quote di categoria di Amazon Cognito. Puoi utilizzare la console Service Quotas per visualizzare l'utilizzo delle quote, richiedere un aumento delle quote e creare CloudWatch allarmi per monitorare l'utilizzo delle quote. Come parte di questa modifica, la sezione CloudWatch Metriche disponibili per i pool di utenti di Amazon Cognito è stata aggiornata per riflettere le nuove informazioni. Il nuovo nome della sezione è: Tracciamento delle quote e dell'utilizzo in CloudWatch e Service Quotas	29 ottobre 2020
Categorizzazione delle quote di Amazon Cognito	Per monitorare l'uso delle quote e richiederne un aumento sono disponibili le categorie di quote. Le quote sono raggruppate in categorie in base a casi d'uso comuni.	17 agosto 2020
Amazon Cognito è ora supportato negli Stati Uniti AWS GovCloud	Amazon Cognito è ora supportato nella regione AWS GovCloud (Stati Uniti).	13 maggio 2020

Aggiornamenti dei documenti Amazon Cognito Pinpoint	È stato aggiunto un nuovo ruolo collegato ai servizi. Le istruzioni in "Utilizzo dell'analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito" sono state aggiornate.	13 maggio 2020
Nuovo capitolo dedicato alla sicurezza di Amazon Cognito	Il capitolo Sicurezza può aiutare la tua organizza zione a ottenere informazi oni approfondite sulla sicurezza dei servizi integrata e configurabile. AWS I nostri nuovi capitoli forniscono informazioni sulla sicurezza del cloud e nel cloud.	30 aprile 2020
Amazon Cognito Identity Pools ora supporta l'accesso con Apple	L'accesso con Apple è disponibile in tutte le regioni in cui opera Amazon Cognito, ad eccezione della regione cn-north-1.	7 aprile 2020
Nuova versione dell'API di Facebook	Aggiunta la selezione della versione all'API di Facebook.	3 aprile 2020
Aggiornamento della differenz a tra maiuscole e minuscole	Aggiunta la raccomandazione sull'abilitazione della disattiva zione della distinzione tra maiuscole e minuscole del nomeutente prima di creare un bacino d'utenza.	11 febbraio 2020

Nuove informazioni su AWS Amplify	Sono state aggiunte informazioni sull'integrazione di Amazon Cognito con la tua app web o mobile AWS Amplify utilizzando SDK e librerie. Rimosse le informazioni sull'utilizzo degli SDK Amazon Cognito precedenti a AWS Amplify.	22 novembre 2019
Nuovo attributo per i trigger del pool di utenti	Amazon Cognito ora include un `clientMetadata` parametro nelle informazioni sugli eventi che trasmette alle AWS Lambda funzioni per la maggior parte dei trigger del pool di utenti. Puoi utilizzare questo parametro per migliorare il flusso di lavoro di autenticazione personalizzato con dati aggiuntivi.	4 ottobre 2019
Limite aggiornato	Il limite di limitazione per l'azione ListUsers API viene aggiornato.	25 giugno 2019
Nuovo limite	I limiti flessibili per i bacini d'utenza ora includono un limite per il numero di utenti.	17 giugno 2019

Impostazioni e-mail di Amazon SES per i pool di utenti Amazon Cognito	È possibile configurare un bacino d'utenza in modo che Amazon Cognito contatti via e-mail gli utenti tramite la configurazione di Amazon SES in uso. Questa impostazione consente ad Amazon Cognito di inviare e-mail con un volume di distribuzione più elevato di quanto altrimenti possibile.	8 Aprile 2019
Supporto per l'etichettatura	Aggiunte informazioni sull'assegnazione di tag alle risorse Amazon Cognito.	26 marzo 2019
Modifica il certificato per un dominio personalizzato	Se utilizzi un dominio personalizzato per l'interfaccia utente ospitata di Amazon Cognito, puoi modificare il certificato SSL per tale dominio in base alle esigenze.	19 dicembre 2018
Nuovo limite	È stato aggiunto un nuovo limite per il numero massimo di gruppi ai quali può appartenere ogni utente.	14 dicembre 2018
Limiti aggiornati	Sono stati aggiornati i limiti flessibili per i bacini d'utenza.	11 dicembre 2018
Aggiornamento della documentazione per la verifica degli indirizzi e-mail e dei numeri di telefono	Sono state aggiunte informazioni sulla configurazione del pool di utente per richiedere la verifica di e-mail o telefono quando un utente effettua la registrazione all'app.	20 novembre 2018

Aggiornamento della documentazione per testare le e-mail	Sono state aggiunte istruzioni per l'avvio delle e-mail da Amazon Cognito durante il test dell'app.	13 novembre 2018
Sicurezza avanzata Amazon Cognito	Sono state aggiunte nuove funzionalità di sicurezza per consentire agli sviluppatori di proteggere app e utenti da bot dannosi, garantire gli account utente se le credenziali sono state compromesse e regolare in modo automatico le sfide richieste per effettuare l'accesso in base al rischio calcolato del tentativo di accesso.	14 giugno 2018
Domini personalizzati per l'interfaccia utente ospitata da Amazon Cognito	Consente agli sviluppatori di utilizzare il proprio dominio completamente personalizzato per l'interfaccia utente ospitata in bacini d'utenza di Amazon Cognito.	4 giugno 2018
Pool di utenti di Amazon Cognito (OIDC Identity Provider)	Aggiunta della procedura di accesso al bacino d'utenza tramite un provider di identità OpenID Connect (OIDC), ad esempio Salesforce o Ping Identity.	17 maggio 2018
Trigger di migrazione Amazon Cognito Lambda	Aggiunte le pagine che descrivono la funzione trigger di migrazione Lambda	8 aprile 2018

Aggiornamento della Guida per gli sviluppatori di Amazon Cognito	Aggiunto il livello principale "Che cos'è Amazon Cognito" e "Nozioni di base su Amazon Cognito". Aggiunti inoltre scenari comuni e riorganizzato il sommario dei bacini d'utenza. Aggiunta una nuova sezione "Nozioni di base sui bacini d'utenza di Amazon Cognito".	6 aprile 2018
Beta di sicurezza avanzata di Amazon Cognito	Sono state aggiunte nuove funzionalità di sicurezza per consentire agli sviluppatori di proteggere app e utenti da bot dannosi, garantire gli account utente se le credenziali sono state compromesse altrove su Internet, oltre a regolare in modo automatico le sfide richieste per effettuare l'accesso in base al rischio calcolato del tentativo di accesso.	28 novembre 2017
Integrazione con Amazon Pinpoint	È stata aggiunta la possibilità di usare Amazon Pinpoint per fornire analisi dei dati per le app dei bacini d'utenza di Amazon Cognito e per migliorare i dati utente per le campagne di Amazon Pinpoint.	26 settembre 2017

Federazione e funzionalità integrate dell'interfaccia utente delle app dei pool di utenti di Amazon Cognito	È stata aggiunta la possibilità di consentire agli utenti di accedere al bacino d'utenza tramite Facebook, Google, Login with Amazon oppure attraverso un provider di identità SAML. È stata aggiunta una interfaccia utente dell'app integrata personalizzabile e il supporto di OAuth 2.0 con attestazioni personalizzate.	10 agosto 2017
Modifiche delle funzionalità relative alla conformità a HIPAA e PCI	È stata aggiunta la possibilità di consentire agli utenti di utilizzare un numero di telefono o un indirizzo e-mail come nome utente.	6 luglio 2017
Gruppi di utenti e funzionalità di controllo degli accessi basate sui ruoli	È stata aggiunta la possibilità amministrativa di creare e gestire i gruppi di utenti. Gli amministratori possono assegnare ruoli IAM agli utenti in base all'appartenenza al gruppo e alle regole create dall'amministratore.	15 dicembre 2016
Aggiornamento della documentazione	Esempi aggiornati che mostrano come utilizzare i AWS Lambda trigger con i pool di utenti.	27 novembre 2016
Aggiornamento della documentazione	Esempi di codice iOS aggiornati.	18 novembre 2016

Aggiornamento della documentazione	Sono state aggiunte informazioni sul flusso di conferma per gli account utente.	9 Novembre 2016
Funzionalità di creazione di account utente	È stata aggiunta la funzione di gestione per creare gli account utente attraverso la console di Amazon Cognito e l'API.	6 ottobre 2016
Funzionalità di importazione degli utenti	È stata aggiunta la funzionalità di importazione in blocco per i bacini d'utenza di Cognito. Utilizza questa funzionalità per migrare gli utenti dal tuo provider di identità esistente a un bacino d'utenza di Amazon Cognito.	1 settembre 2016
Disponibilità generale dei pool di utenti di Cognito	È stata aggiunta la funzionalità dei bacini d'utenza di Cognito. Utilizza questa funzionalità per creare e gestire una directory utente e per aggiungere la registrazione e l'accesso alla tua app per dispositivi mobili o applicazione Web tramite i bacini d'utenza.	28 luglio 2016
Supporto SAML	È stato aggiunto il supporto per l'autenticazione con provider di identità tramite Security Assertion Markup Language 2.0 (SAML 2.0).	23 giugno 2016
CloudTrail integration	Aggiunta integrazione con AWS CloudTrail.	18 febbraio 2016

Integrazione di eventi con Lambda	Consente di eseguire una AWS Lambda funzione in risposta a eventi importanti in Amazon Cognito.	9 aprile 2015
Flusso di dati verso Amazon Kinesis	Fornisce controllo e informazioni nei flussi di dati.	4 marzo 2015
Supporto OpenID Connect	Abilita il supporto per i provider di OpenID Connect.	23 novembre 2014
Sincronizzazione push	Abilita il supporto per la sincronizzazione push in modalità silenziosa.	6 Novembre 2014
Developer-authenticated supporto per le identità aggiunto	Consente agli sviluppatori che possiedono propri sistemi di gestione di autenticazione e di identità ad essere trattati come provider di identità in Amazon Cognito.	29 settembre 2014
Disponibilità generale di Amazon Cognito		10 luglio 2014

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.